SANDIA REPORT

SAND2018-6879
Unlimited Release
Printed June 26, 2018

Sierra Structural Dynamics Verification Test Manual, 4.48 release

Sierra Structural Dynamics Development Team

Latest Software Release:
4.47-7-Development 2017-02-06

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

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

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

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

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

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

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

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

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

SAND2018-6879
Unlimited Release
Printed June 26, 2018

Sierra Structural Dynamics Verification Test Manual, 4.48 release

Sierra Structural Dynamics Development Team

COMPSIM
STRUCTURAL DYNAMICS

Abstract

This document presents tests from the Sierra Structural Mechanics verification test suite. Each of these tests is run nightly with the Sierra/SD code suite and the results of the test checked versus the correct analytic result. For each of the tests presented in this document the test setup, derivation of the analytic solution, and comparison of the Sierra/SD code results to the analytic solution is provided. This document can be used to confirm that a given code capability is verified or referenced as a compilation of example problems.

Contents

Appendix

Chapter 1

Executive Summary

Verification and validation (V&V) of scientific computing programs are important at Sandia National Laboratories (SNL) 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 structural dynamics analysis code (Sierra/SD) plays a central role in the qualification of weapon systems and components for normal and hostile environments throughout the Stockpile-to-Target Sequence. Sierra/SD is used:

- To redesign weapon components.
- To certify weapon components and systems for target environments such as hypersonic vehicles.
- To certify that components will survive the thermomechanical shock loads associated with hostile environments.
- To evaluate current stockpile issues, including issues associated with uncertainty quantification.
- To address many other problems that are encountered in stockpile management.

Furthermore Sierra/SD is an engineering code that is used at Los Alamos National Laboratories (LANL), and elsewhere for achieving the engineering deliverables of those organizations.

This document describes the verification plan for the Sierra/SD code. Additionally detailed description is provided for several key verification tests. The verification tests assure that the mathematics and numerical algorithms associated with functionality describing engineering phenomena in Sierra/SD are implemented correctly. The suite of verification tests will evolve as the functionality of Sierra/SD evolves.

Sierra/SD is developed in accordance with a set of Software Quality Engineering (SQE) practices.[1,2] These procedures conform to those outlined in,[3] but are tailored to Sierra/SD development. It is important to understand the role that these SQE practices play in the overall verification and validation effort.

Chapter 2

Sierra/SD Verification Procedures

2.1 Overview

This document contains a verification overview for the software package Sierra/SD. In contrast to the Sierra/SD User's Manual,[4] which demonstrates how to use the code, and the Theory Manual,[5] which details the underlying mathematics of the code, the verification manual is a list of well documented verified examples demonstrating how the code performs on a subset of verification problems. In additional to the verification tests detailed in this document high confidence in the correctness of Sierra/SD is maintained by an extensive test suite, several code quality tools, and rigorous team processes. The intent is to fully verify each capability in Sierra/SD. This manual should be used to gain a level of confidence in the rigor for which Sierra/SD is verified for high consequence analysis. However, quality verification is a journey of continuous improvement. There may be gaps in the verification coverage. If there is a clear gap in the verification coverage that is essential to analysis, the Sierra/SD team should be contacted at sierra-help@sandia.gov.

2.2 Code Development Practices

The first step to a well verified code is code development practices that ensure all new code features are properly tested. The Sierra/SD team follows the laws of test driven development (TDD) coding practice as outlined in Clean Code.[6] The three laws of TDD are

1. You may not write production code until a failing unit test is written.
2. You may not write more of a unit test than is sufficient to fail.
3. You may not write more production code than is sufficient to fix the currently failing test.

Following these laws ensures that all new capability is covered by tests, and that all capability modified through user stories or corrected by user support is also covered by tests. However, these practices fail to ensure that *all* legacy capability is adequately covered, or that all permutations of capability are well verified. The Sierra/SD process for covering permutations of capability is outlined in 2.8. In addition to the enumerated TDD practices the Sierra/SD development team also uses code reviews, pair programming, and external beta testing as additional safeguards to prevent coding errors.

2.3 Overview of testing Pyramid

In order to efficiently maintain code quality a properly organized suite of tests must be used, a large number of small tests of individual capabilities building up to smaller numbers of large and complex tests. There are many types of tests for Sierra/SD: Unit, Fast (Continuous), Performance, Verification, Regression, and Acceptance. For tests to have value they most be run regularly and in an automated fashion. With the exception of a few large acceptance tests the entire Sierra/SD test suite is run nightly.

- *Unit Tests:* a test of an individual source code function. Unit tests are generally run through the Google GTEST framework. A unit test can be used to verify a given function has the correct behavior for every possible input. Unit tests are very fast. Sierra/SD currently uses many thousand unit tests.

- *Fast Tests:* a test that must run in under ten seconds. Fast tests are run every hour on the master branch of the Sierra code base. This high run frequency allows quickly pinpointing any issues introduced into the code base. The fast test suite is designed to give a broad coverage of all core Sierra/SD features. Sierra/SD uses about a thousand fast tests.

- *Verification Tests:* a test that compares test outputs to an analytic result or confirms the test has some expected property (such as a convergence rate.) Verification tests are one of the most valuable test types and the verification test suite will continue to be expanded over time. Sierra/SD maintains about a thousand verification tests.

- *Regression Tests:* a test that confirms the code produces and expected output, but without rigorous mathematical demonstration that the output is indeed correct. Generally a test case is produced and then engineering judgment used to confirm the test case is behaving as expected. The test then confirms this approved behavior is maintained. An example would be the modal decomposition of a complex shape part. Currently Sierra/SD uses several thousand regression test. Regression tests are a necessity, but the the Sierra/SD development team is moving over time to a larger balance of tests in the more valuable unit and verification categories.

- *Performance Tests:* a test used to confirm Sierra/SD maintains acceptable runtime and memory use bounds. These tests are expensive and Sierra/SD maintains only about a hundred.

- *Acceptance Tests:* a test of a full analysis use case provided by an analyst. Acceptance tests are the largest and most complex tests in the system. An acceptance test ensures the work flow for an entire complex analysis chain maintains functionality. As acceptance tests are very expensive Sierra/SD maintains only about a dozen to cover the most important and commonly used work flows.

2.4 User Support Process

The key to credible capability is a user support process that identifies, patches, and tests against any bugs found by Sierra/SD analysts. When a bug report is submitted a minimal representative example of the bug is produced by the developers and added as a test to the nightly test suite. After necessary development is done to resolve the issue the new nightly test ensures that the bug will not reappear in future releases.

2.5 Verification Policy for New Features

When new capability is added to Sierra/SD, the code development processes outlined in Clean Code[6] and Test Driven Development are followed. The new development *always* begins with a unit testing of new functionality. After completing the unit test, a self-documenting verification test is added that demonstrates the capability reproduces an analytical result. Additionally, regression tests may added that exercise the range of inputs of the capability. Once these tests are in place, an acceptance model, received from key analyst stakeholders, is run to ensure the capability behaves as expected and gives an acceptable result.

The Sierra/SD team migrated to a structure of individual test documentation maintained in the test repositories in 2013. The legacy formats are also included in this document, and eventually will be migrated to the new format. Thus though all verification tests are verified to a high level of rigor, not all verification tests are included in this verification test manual.

2.6 Nightly Testing Process

Every night the entire code base is compiled on multiple platforms with multiple compilers. Some subset of the nightly tests are run on each platform. Every fast and nightly test is run on the development platform, compiled with both debug/release and gcc/intel compilers. Additionally, all nightly tests are run on the Trinity surrogate (both Haswell and Knights Landing chips). The entire test suite (including performance tests) are run on intel-release on the primary HPC production platform dedicated to Nuclear Deterrence. Some subsection of the tests are run on experimental platforms, such as Darwin (MAC-OS), Broadwell, and Ride (GPU). These tests are useful because they may identify software quality issues that don't cause problems in the production platforms, but could in the future as new platforms move into production.

2.7 Other SQA Tools

In addition to the nightly testing process, other software quality tools are run nightly to check for possible code errors or gaps in testing coverage. These tools include the memory checker Valgrind, the Feature Coverage Tool (FCT), and the Line Coverage Tool (LCOV).

2.7.1 Valgrind

Valgrind is a tool used to check for memory leaks and memory errors. A memory leak is when memory is allocated, but never freed while the program is still running. The existence of memory leaks within loops can lead to a simulation taking an increasing amount of memory as simulation time increases, eventually leading to code failure. A memory error represents the executable accessing memory that has not been allocated, or is otherwise out of bounds. A memory error generally results in unpredictable behavior, and can lead to fatal segmentation faults. Valgrind is run nightly on both the "nightly" and "fast" tests. All memory leaks and errors are eliminated for every sprint snapshot and release version of Sierra/SD.

2.7.2 LCOV

The coverage tool For Sierra/SD, LCOV, measures the code source line coverage of unit, fast, and nightly testing. The LCOV tool reports how many times each line of code is called for the respective test suite. For each file, folder, and executable in Sierra LCOV reports the percentage of lines in the code that are covered by at least one test. For example, as of the 4.48 release, unit tests cover 48.3%, fast tests cover 79.4%, and nightly tests cover 86.0% of the code base. It is up to the development team to ensure that all new features are well covered. The Sierra/SD development team strives to improve test code coverage over time. However, 100.0% coverage is not always practical. Some uncovered code is either non-released research capability or depreciated legacy capability. Additionally many error messages do not have a test that hits the error message, therefore the line of code with the error message may be uncovered.

2.8 FCT

For Sierra/SD the Feature Coverage Tool (FCT) creates three documents from an input file; the annotated input file, the two way coverage graph and the list of best matching tests.

The FCT can be used by analysts to assess the Sierra/SD verification rigor for a specific analysis. Additionally the Sierra/SD development team can use output of the FCT prioritize needs for verification test suite improvement.

The annotated input file shows the features (corresponding to input deck lines) that are used in verification tests (in green), regression tested (yellow) or untested (red). Developers and analysts can use this tool to see if for an analysis in question untested features are used and take action to mitigate or explain them. One mitigation strategy is to create a new verification test for the feature. An explanation is needed if the FCT has indicated a false positive (the FCT tool is helpful, but still in development).

The second document produced by FCT is the two way coverage chart. The two way coverage chart indicates for any two features if a verification or regression test exists that uses both of those features simultaneously. It can be impractical to add a verification test every possible feature combination. However, the two way coverage report can be used to see if certain key feature combinations are tested together, such as damping in a transient analysis or strain output on shell elements. Lack of a two way coverage test may indicate additional verification testing is needed, though engineering judgment must be applied to identify the most critical feature combinations.

The third FCT output is a of list the top 5 verification tests nearest to (in the sense of using the same capabilities) as used in the input file. If an analysis has a very closely matching rigorous verification test is gives high confidence that the entire use case of the analysis and all feature combinations used are well verified in conjunction.

This page intentionally left blank.

Chapter 3

Sierra Structural Dynamics Verification Tests

12

3.1 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.

3.1.1 One Hex Models

The model is shown in Figure 3.1. There are two hex elements in the structure. The element on the right of the figure will be reduced to a superelement. The element on the left is the "residual structure," which uses the previously generated superelement. It is clamped on the left surface. 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 as the sensitivity parameter. The model is selected so there are no repeated frequencies.

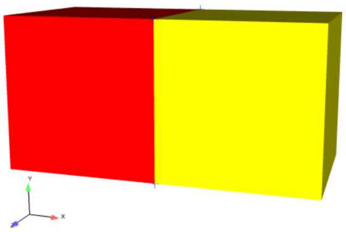


Figure 3.1: One Hex superelement model

3.1.2 Analysis

3.1.2.1 Analytic Analysis

For a constant vector sensitivity analysis, the reduced order matrices are given by,

$$\begin{aligned} \tilde{k}_o &= T_o^T K(p_o) T_o && (3.1)\\ \tilde{k}_1 &= T_o^T K(p_o + \Delta p) T_o && (3.2)\\ \frac{d\tilde{k}}{dp} &\approx \frac{\tilde{k}_1 - \tilde{k}_o}{\Delta p} && (3.3) \end{aligned}$$

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.

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.

For a **finite difference** sensitivity analysis, the relations are somewhat different.

$$\begin{aligned} \tilde{k}_o &= T_o^T K(p_o) T_o && (3.4)\\ \hat{k}_1 &= T_1^T K(p_o + \Delta p) T_1 && (3.5)\\ \frac{d\tilde{k}}{dp} &\approx \frac{\hat{k}_1 - \tilde{k}_o}{\Delta p} && (3.6) \end{aligned}$$

Here,

T_1 is the transformation matrix evaluated at $p_o + \Delta p$,

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.

3.1.2.2 Numerical Results

Figure 3.2 shows a comparison of the system level solutions as a function of density. 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 3.3

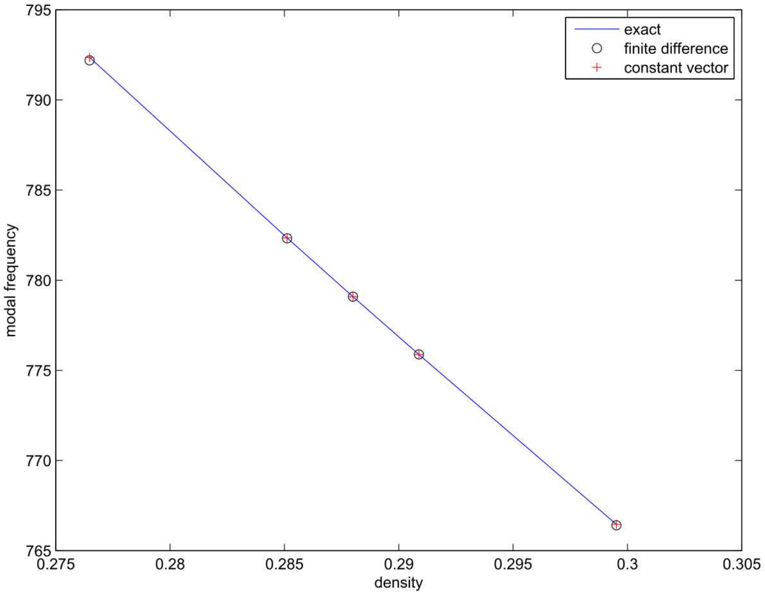


Figure 3.2: Modal Frequency Variation with Density

3.1.3 Summary

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 A.4.

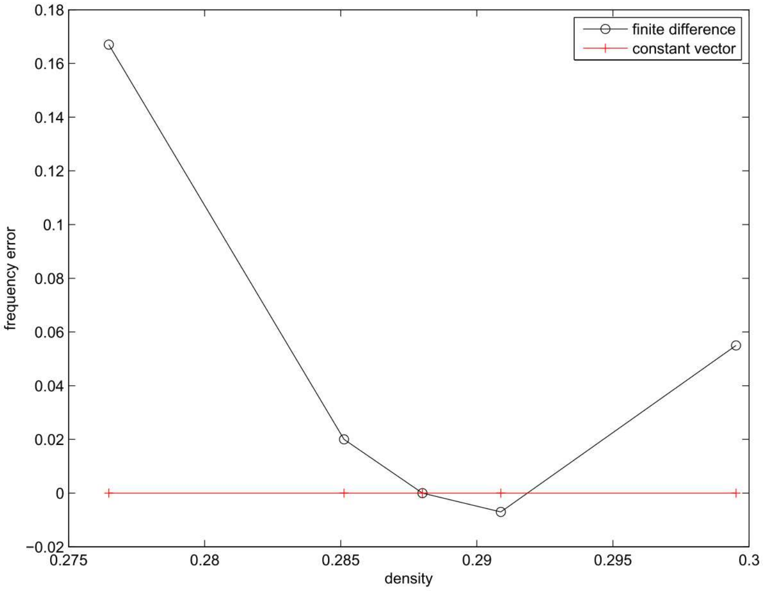


Figure 3.3: Modal Frequency Error with Density

3.2 Superelement 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 superelement run are entered in the block section of the input deck during the CBR solution. Figure 3.5 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 A.5.

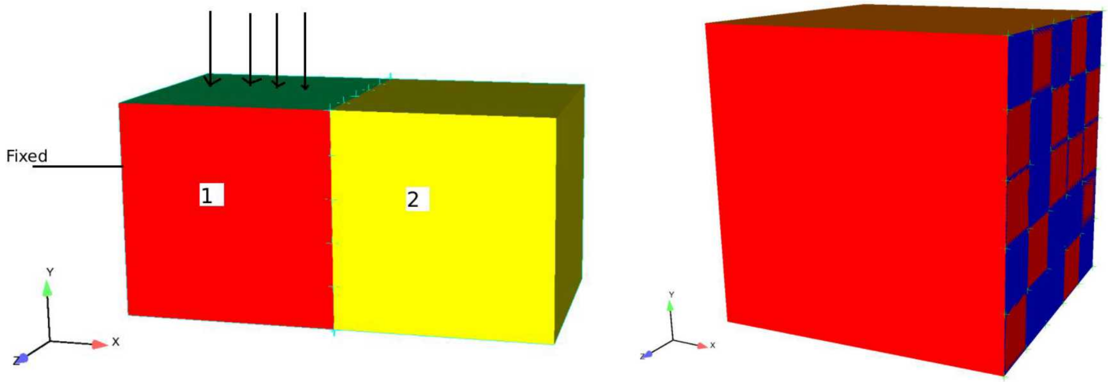


Figure 3.4: Initial model and model with superelement

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

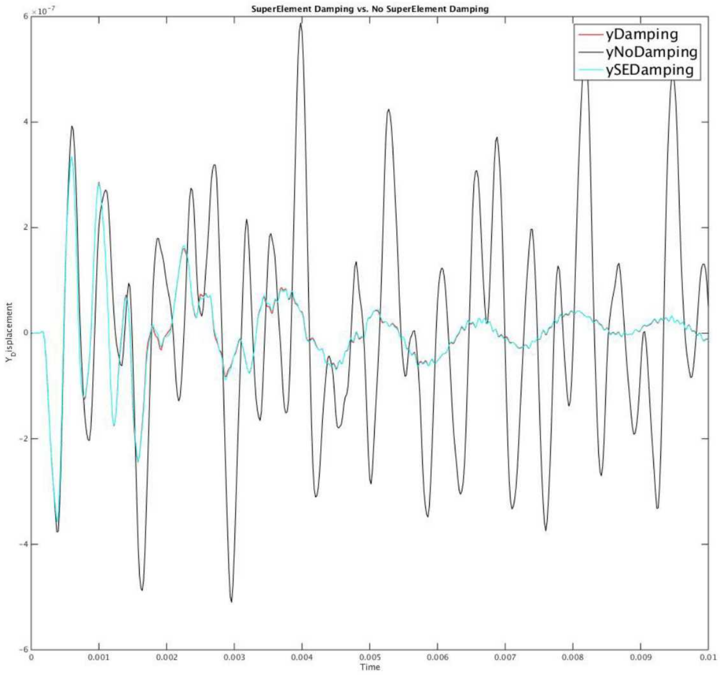


Figure 3.5: Superelement Damping Results. Damped and undamped response of full system models compared with damped model of the reduced order model.

3.3 SierraSM to SierraSD Coupling

3.3.1 Beam Preload

This is a verification test comparing Abaqus to Sierra-SD using selective deviatoric hex8 elements. The geometry of this model can be seen in Figure 3.6. 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 Sierra-SM, and then a modal analysis is performed in Sierra-SD. For verification, the first 4 modes are compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 3.1.

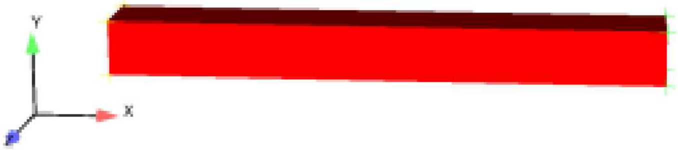


Figure 3.6: Geometry of Bar

Table 3.1: 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

3.3.2 Plate Preload Verification

This example is a similar to the previous model, except that it has the geometry of a plate, as shown in Figure 3.7. The plate consists of selective deviatoric hex8 elements and is fixed on oneside and constrained in the y and z directions on the other. A prescribed displacement is applied in the x-direction in Sierra-SM, and then a modal analysis is performed in Sierra-SD. For verification, the first 5 modes are compared to the Abaqus finite element code.The Eigenvalue results are shown in Table 3.2. The path to these verification tests is Salinas_rtest/verification/adagio_coupling/barModelPreload.

For input deck see Appendix A.19.

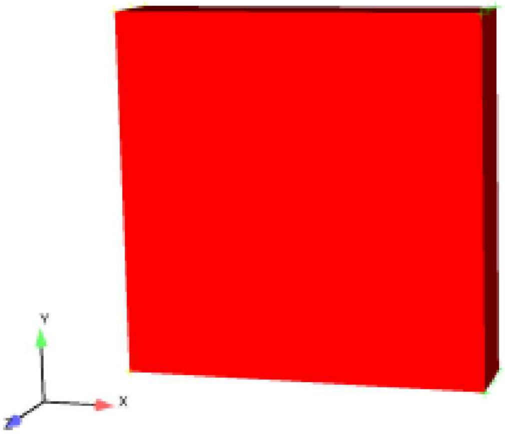


Figure 3.7: Geometry of plate

Table 3.2: 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

3.4 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 3.9. For restart analysis two solution cases and input decks are needed. First, a restart=write solution where the desired amount of steps are analyzed and a output file is created with results. Second, a restart=read solution where the output file that was created is now read in and analyzed to the new desired amount of steps. For this test the write file had 10 steps and the read in file had 20 steps. This test includes superelements, infinite elements, and tied joints. A truth model was constructed with no restart and used for verification. Figure 3.8 shows the comparison of the truth model with no restart and the model with restart.

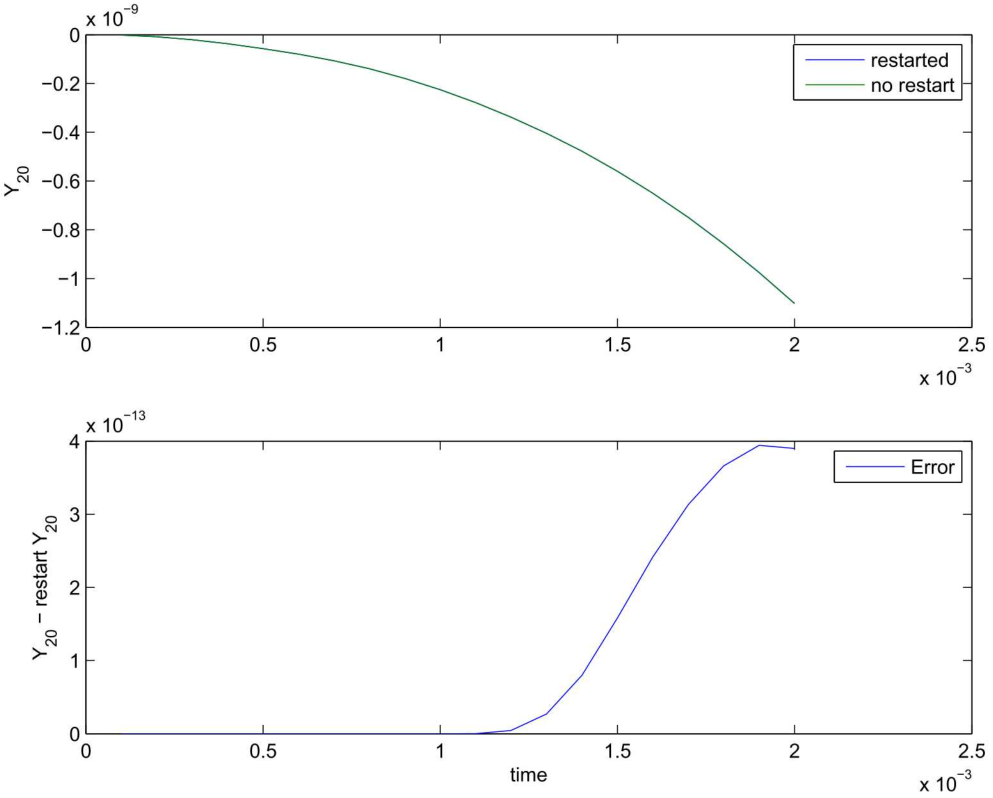


Figure 3.8: Comparison between truth model and restart

In Figure 3.8 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.

3.4.1 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.

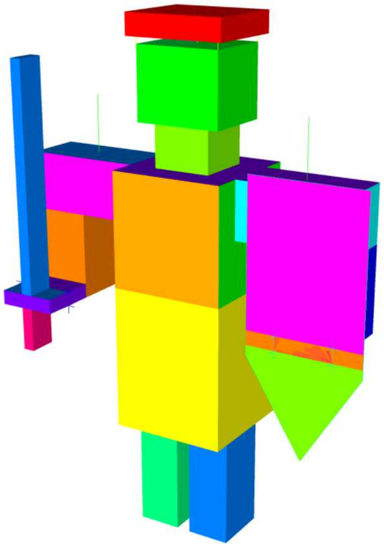


Figure 3.9: Restart Model Geometry. "Ninjabot"

For input deck see Appendix A.24.

3.5 Filter Rigid Modes from Loads

3.5.1 Introduction and Purpose

For some analyses, it is advantageous to remove the rigid body component of a solution. 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. The static preload is a singular system if the force is not properly self-equilibrated. 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. The numerical analyst may represent that physics by a random pressure load. 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 the solve, leaving a displacement that has no rigid body component.

3.5.2 Description of the Test

In this test, a small beam of Hex8 elements has a load applied transverse to one end. See Figure 3.10. Because there are no boundary conditions, the resulting system is singular for a statics solution. Figure 3.11 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.

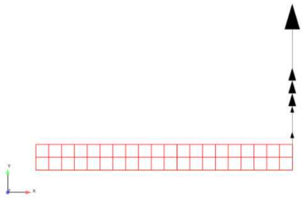


Figure 3.10: Beam Loading

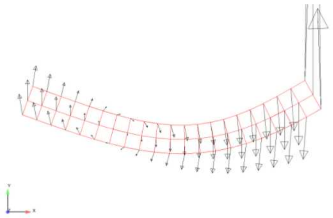


Figure 3.11: Filtered Beam Forces and Displacements

3.5.3 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 A.25.

3.6 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 3.12 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 3.12.

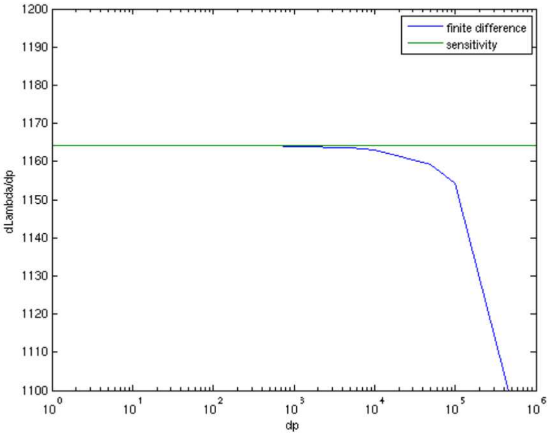


Figure 3.12: dLambda/dp vs. dp

Figure 3.13 shows the frequency vs. *dp*. For input deck see Appendix A.28.

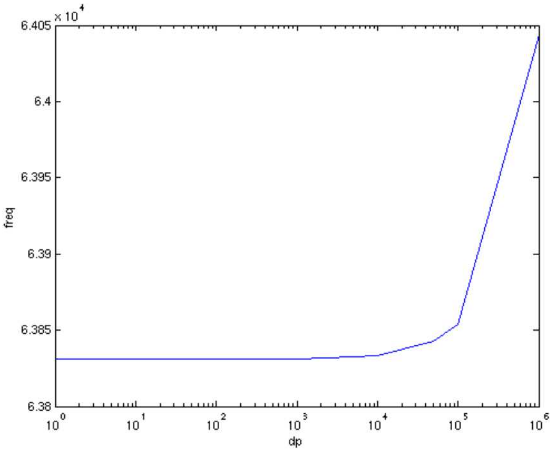


Figure 3.13: Frequency vs.dp

3.7 Sensitivity Analysis with a Superelement

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.

3.7.1 Blade Model

The model is shown in Figure 3.14. 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.

3.7.2 Analysis

3.7.2.1 Analytic Analysis

For a constant vector sensitivity analysis, the reduced order matrices are given by,

$$\tilde{k}_o = T_o^T K(p_o) T_o \tag{3.7}$$

$$\tilde{k}_1 = T_o^T K(p_o + \Delta p) T_o \tag{3.8}$$

$$\frac{d\tilde{k}}{dp} \approx \frac{\tilde{k}_1 - \tilde{k}_o}{\Delta p} \tag{3.9}$$

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,

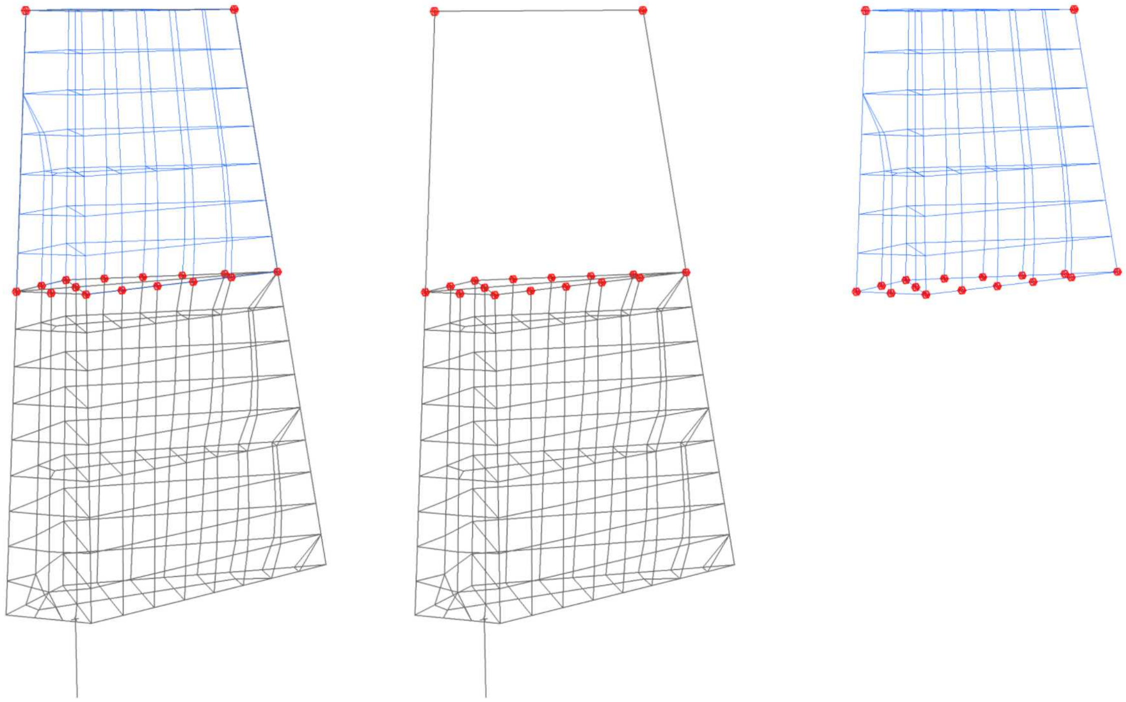


Figure 3.14: Blade superelement model

$\tilde{k}$ 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.

$$\tilde{k}_o = T_o^T K(p_o) T_o \tag{3.10}$$
$$\hat{k}_1 = T_1^T K(p_o + \Delta p) T_1 \tag{3.11}$$
$$\frac{d\tilde{k}}{dp} \approx \frac{\hat{k}_1 - \tilde{k}_o}{\Delta p} \tag{3.12}$$

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.

3.7.2.2 Numerical Results

Figure 3.15 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 3.16

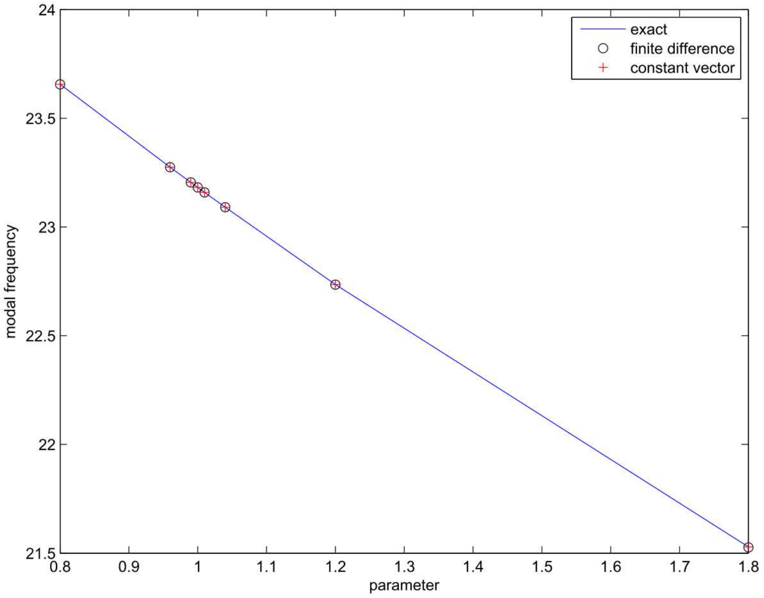


Figure 3.15: Modal Frequency Variation with Density

3.7.3 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 eigensolver to fail. For input deck see Appendix A.29.

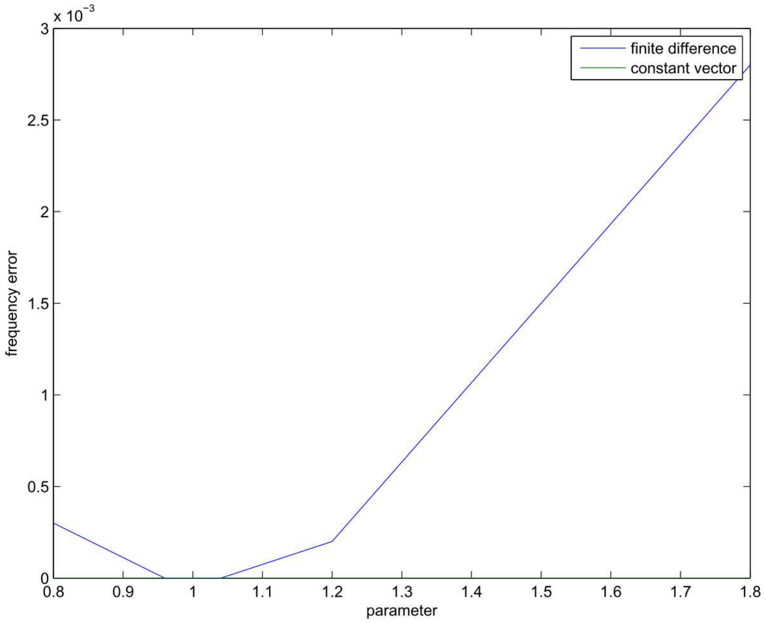


Figure 3.16: Modal Frequency Error with Density

3.8 Shock Tube

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

3.8.1 Problem Description

This is the verification test of nonlinear acoustics.

3.8.2 Verification of Solution

The SierraSD nonlinear acoustics equation is the Kuznetsov equation. In the SierraSD Verification manual, see section 10.2 and specifically the subsectiom 10.3. Fubini's exact solution to a wave guide is used. A Matlab script, the section Verification of solution of Kuznetsov equation. A Matlab script, `shocktube_exact_solution.m` generates the exact solution.

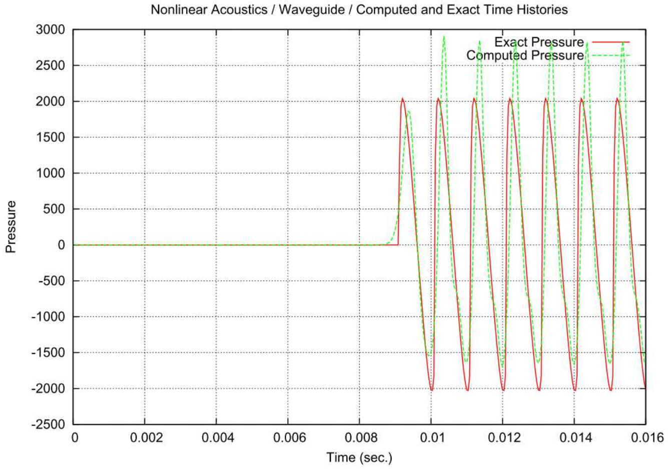


Figure 3.17: Shock Tube

For input deck see Appendix A.30.

3.9 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 sawtooth 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 3.18 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 3.19 shows the difference of the solutions.

For comparison, Figure 3.20 compares results with a CBR model that includes no generalized dofs. As anticipated, the results are not nearly as good.

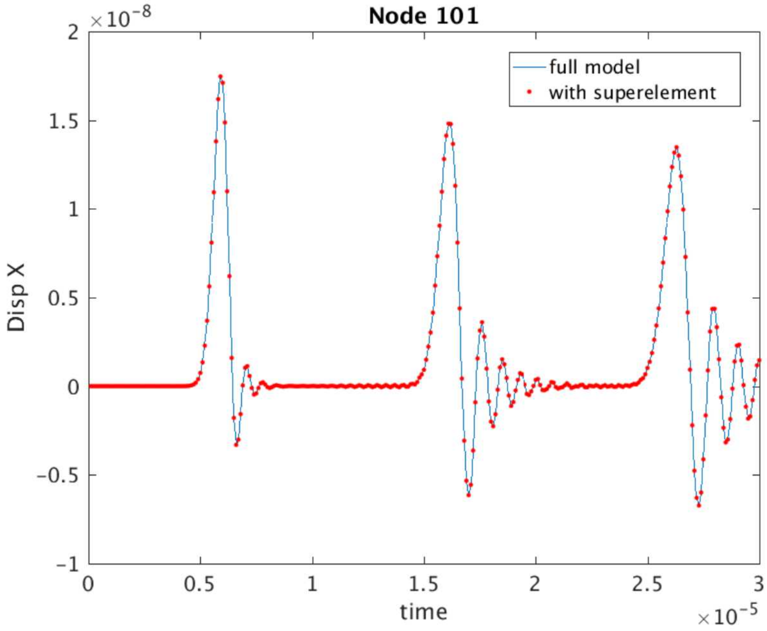


Figure 3.18: Comparison of Full Model with CBR Reduction

For input deck see Appendix A.39.

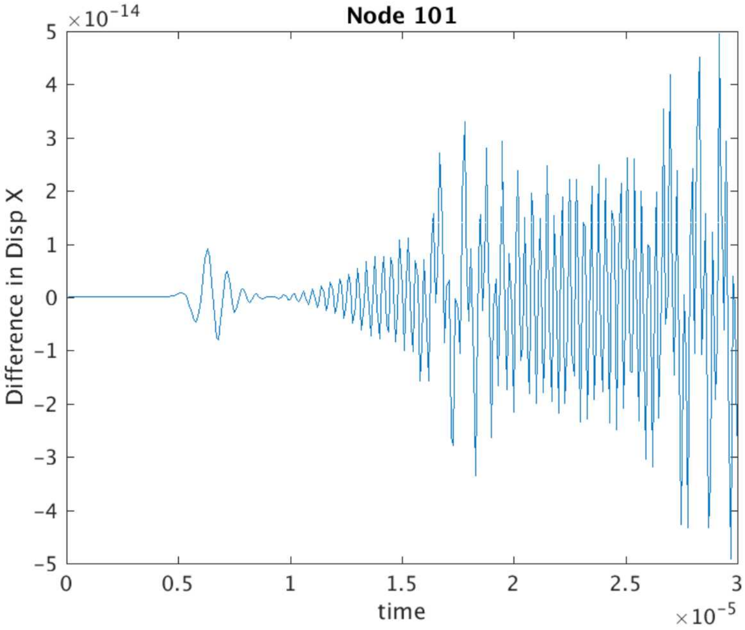


Figure 3.19: Error in CBR reduction

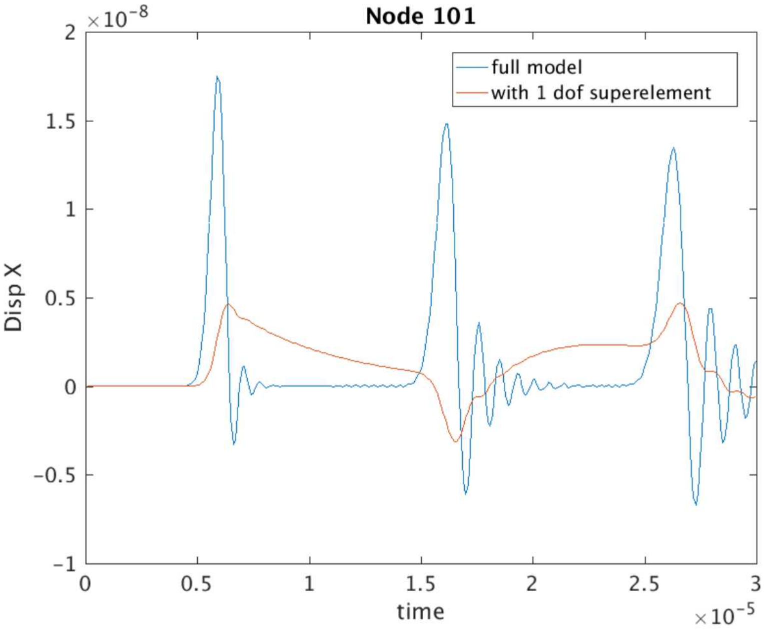


Figure 3.20: Comparison of Full Model with Guyan Reduction. Without the generalized dofs, the comparison is poor.

3.10 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 3.21. The model used is shown in Figure 3.22. 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).

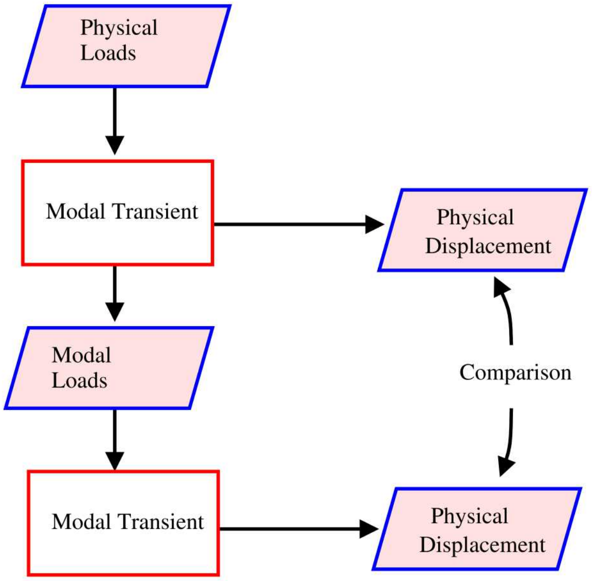


Figure 3.21: Verification Process for Modal Force

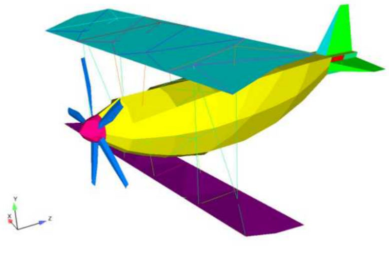


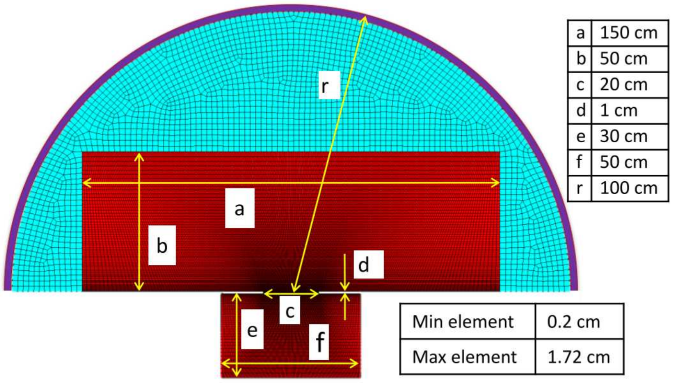
Figure 3.22: Biplane Model

For input see Appendix A.42

3.11 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 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 is used to create an equivalent elemental force vector. The divergence of the Lighthill tensor is provided from a Fuego incompressible fluids simulation.

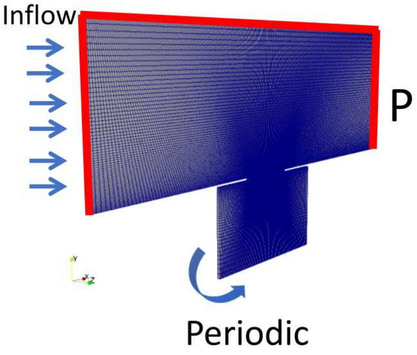
Verification of the Lighthill loading is performed for the Helmholtz resonator shown in Figure 3.23 which has an analytic resonant frequency of 120Hz. The discretized mesh, material properties, initial and boundary conditions used in the Fuego simulation are shown in Figure 3.24. Fuego then calculates the divergence of the Lightill tensor and writes this out to exodus as nodal data at variable timesteps.



a	150 cm
b	50 cm
c	20 cm
d	1 cm
e	30 cm
f	50 cm
r	100 cm

Min element	0.2 cm
Max element	1.72 cm

Figure 3.23: Dimensions of Helmholtz resonator



Single element depth of 1 cm
Number of elements = 27352
Number of nodes = 55640
Inflow = 2700 cm/s
P = 1.01325e06 dynes/cm^2
Time = 0.3 sec
Time step ~3e-5 - 6e-5 sec
CFL = 0.9
Equations:
Continuity
X/Y/X Momentum
Turbulent Kinetic Energy
Turbulent Frequency

Figure 3.24: 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 3.23 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 3.24. boundary with absorbing boundary conditions eliminate the rigid body modes from the solution which can cause a which linear linear growth in the pressure field. The nodal DivT data on the Fuego domain is converted to nodeset data using the ejoin flag -convert_nodal_to_nodesets. The distribution factors for the new nodeset data are changed from 0 to 1. The Sierra/SD simulation reads in the time varying nodeset data from Fuego and interpolates it to the nearest timestep either linearly or using the closest timestep. 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 3.25 for acoustic pressure versus time. An FFT of the pressure data is shown in Figure 3.26 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.

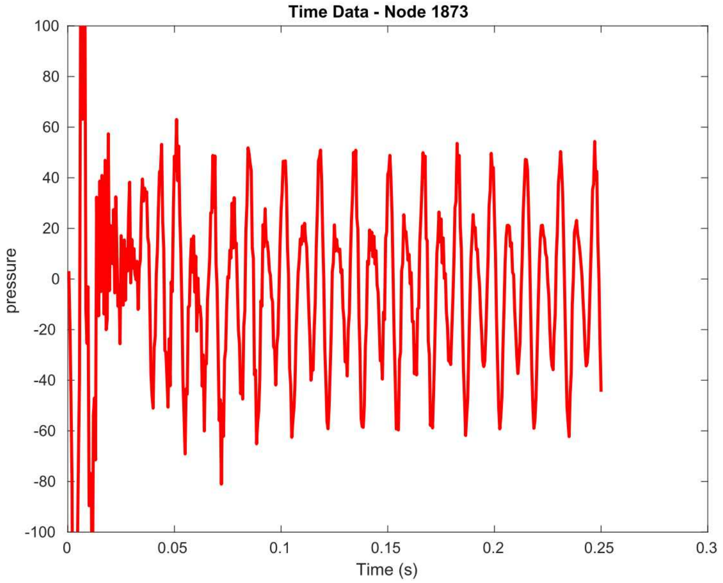


Figure 3.25: Sierra/SD time history of pressure for Lighthill loading.

For input see Appendix A.43

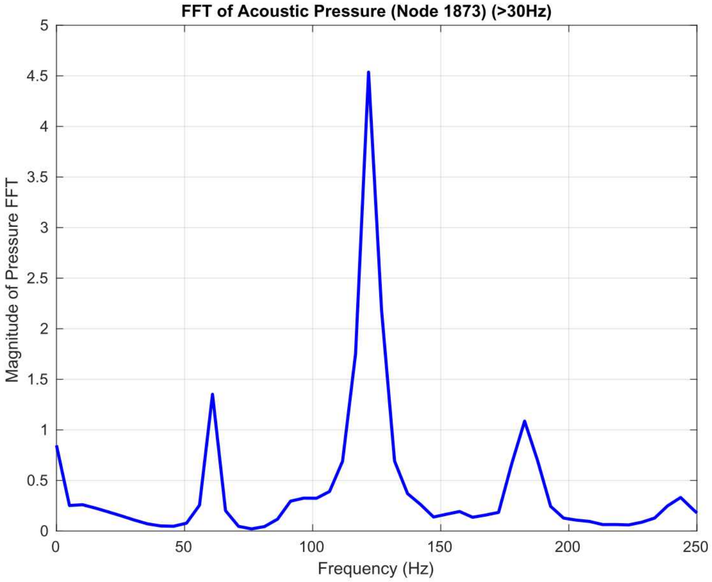


Figure 3.26: FFT of Sierra/SD pressure data shown in Figure 3.25

3.12 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 3.27a). 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

$$(\nabla \cdot \mathbf{T})_x = \left(1 + \cos\left(\frac{\pi x}{20}\right)\right) \sin^2\left(\frac{\pi t}{40}\right) \text{ for } t \leq 40\text{s} \tag{3.13}$$

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, $x = \pm 20$.

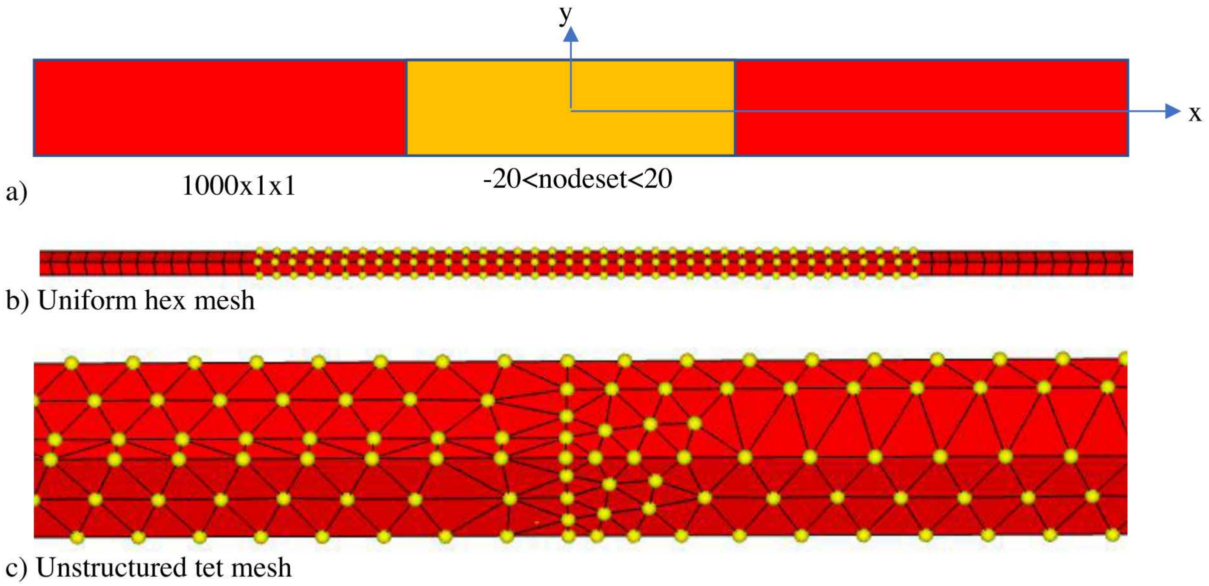
The same pressure response as that given in equation 3.13 is produced with a scalar nodal load equal to $\nabla \cdot (\nabla \cdot \mathbf{T})$ properly scaled by the number of nodes and area it is acting over. For the $\nabla \cdot \mathbf{T}$ used in this example,

$$\nabla \cdot (\nabla \cdot \mathbf{T}) = -\frac{\pi}{20}\left(\sin\left(\frac{\pi x}{20}\right)\right) \sin^2\left(\frac{\pi t}{40}\right) \text{ for } t \leq 40\text{s} \tag{3.14}$$

and the scalar nodal force applied using Point Volume Acceleration is $\frac{1}{4}\nabla \cdot (\nabla \cdot \mathbf{T})$ for the uniform linear hexahedral mesh shown in Figure 3.27b where each element is 1x1x1.

Figure 3.28 shows the pressure output at t=75s over the length of the waveguide for `lighthill` and `pointVolAcc` loading given by equations 3.13 and 3.14, respectively, applied to the uniform mesh shown in Figure 3.27b. 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 3.29 at t=75s. The L1 error of the pressure over the domain is shown at each simulation timestep in Figure 3.30. 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 3.27a) was also discretized with an unstructured linear tetrahedral mesh shown in Figure 3.27c) and `lighthill` loading was applied to the domain. Results for these simulations are also shown in Figures 3.28-3.30 and show the same error as the uniform hexahedral mesh with Lighthill loading.



a)

b) Uniform hex mesh

c) Unstructured tet mesh

Figure 3.27: 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.

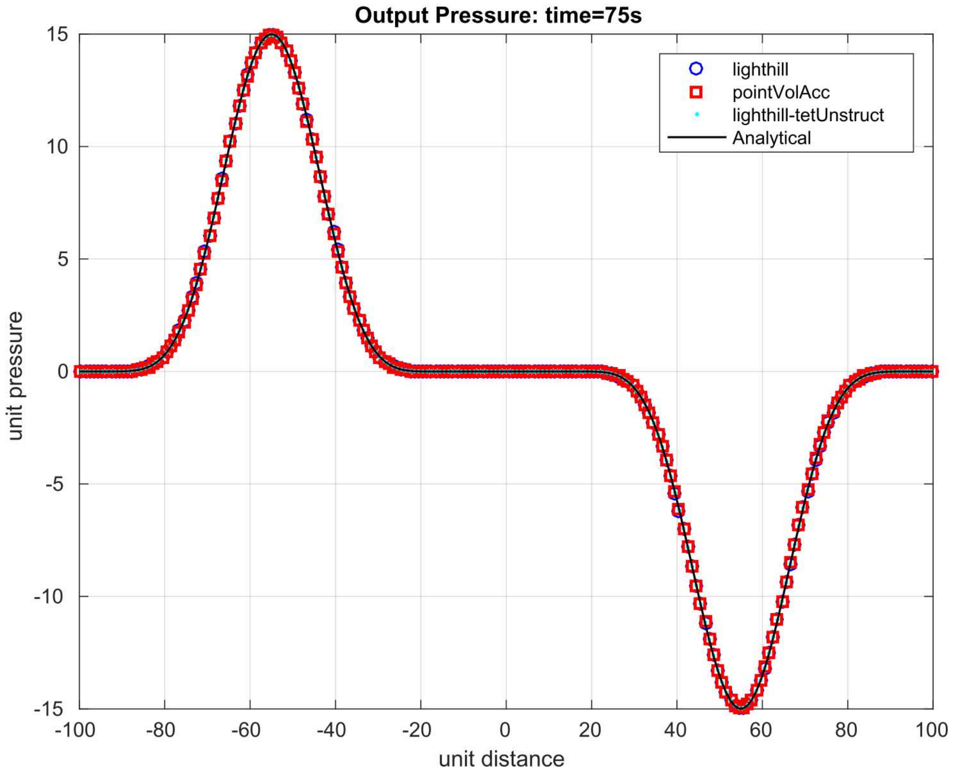


Figure 3.28: Pressure output for 3 load cases compared to analytical result at t=75s.

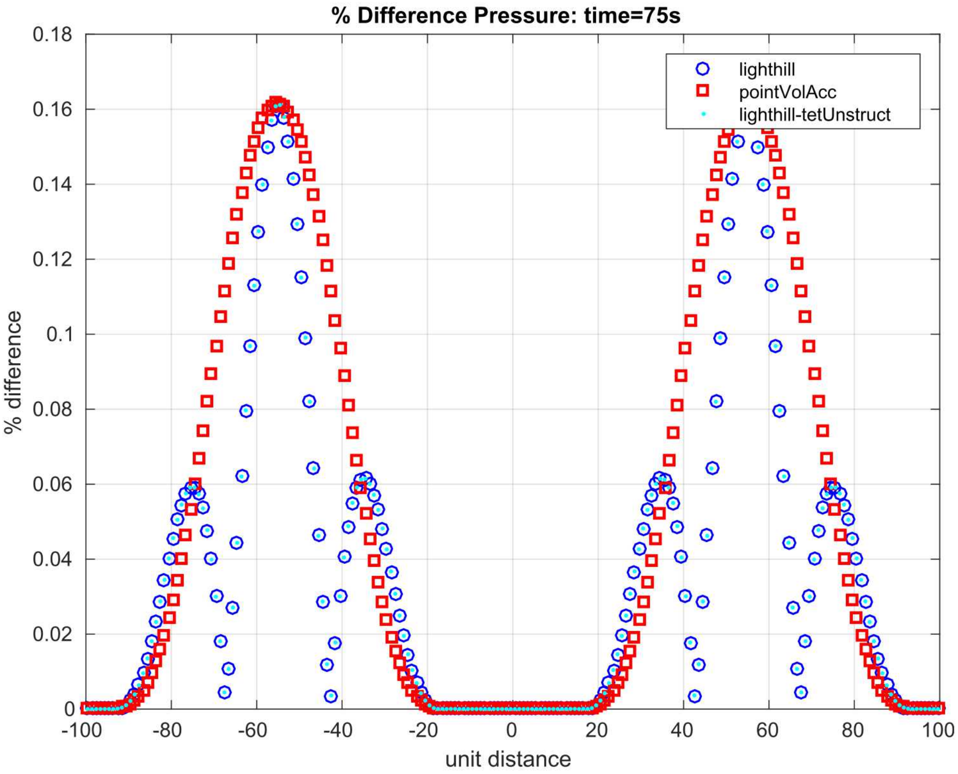


Figure 3.29: Percent difference in pressure between the three load cases and the analytical pressure for t=75s.

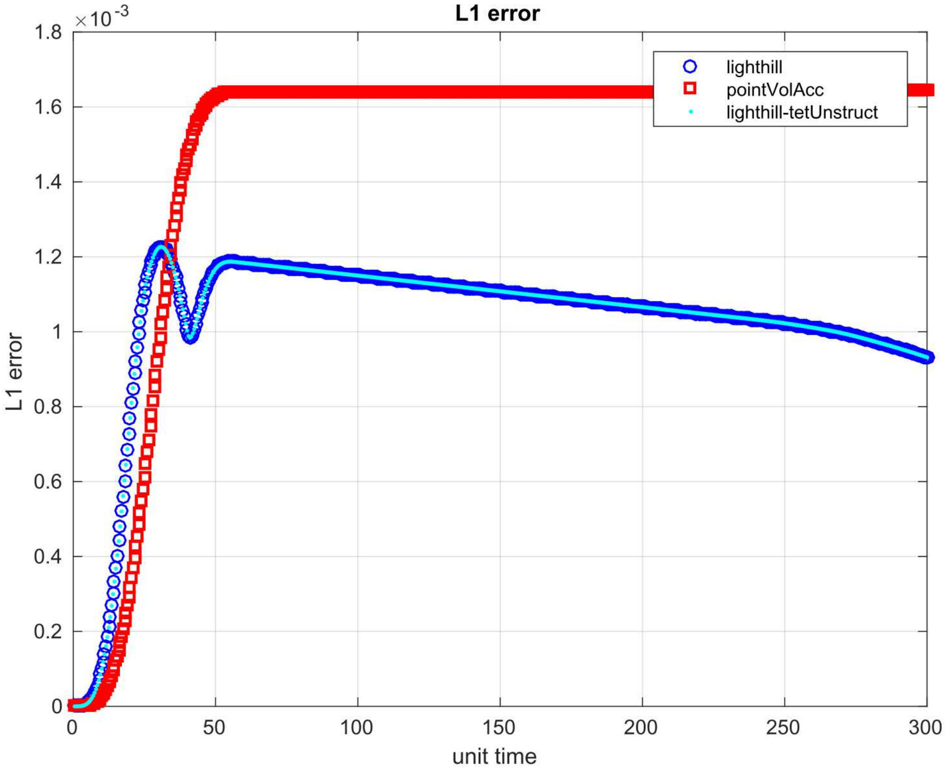


Figure 3.30: L1 error in pressure for each load type versus time.

For input see Appendix A.44

3.13 Superelement Superposition

A four truss, 1-D problem provides a simple verification of Craig-Bampton Reduction (CBR) and superposition based reconstruction. As illustrated in Figure 3.31, 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 3.32 provides a comparison of the solutions using the full model, and the individual components.

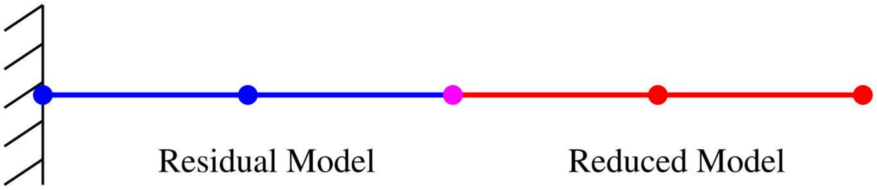


Figure 3.31: Four Truss Geometry

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

[2] The CBR reduction must use lumped masses for consistency with the statics solutions.

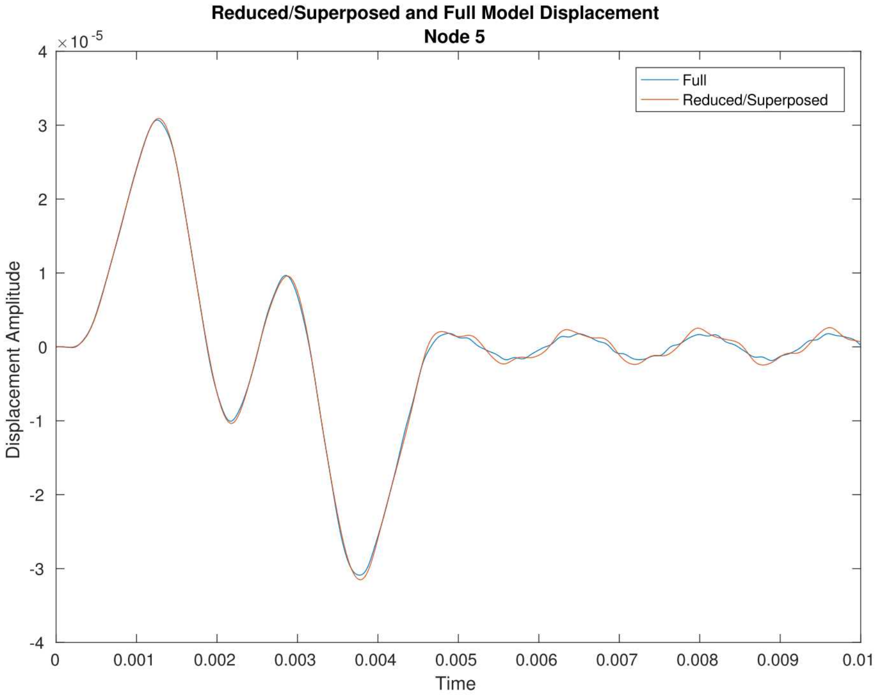


Figure 3.32: Results of Superposition Problem

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.

The transient response on the interior degrees of freedom is,

$$u_k(t_n) = \sum_{i}^{nmodes} q_i(t_n)\phi_{ik} + \sum_{j}^{nconstraint} w_j(t_n)\psi_{jk} \tag{3.15}$$

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 A.45

3.14 Superelement Inertia Tensor and Mass Inertia Matrix

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.

There are two matrices associated with CB reduction and rigid body applications. The inertial tensor, $I_v = T^T R$, is used to establish initial velocity. Here T is the CB reduction matrix and R is a six column rigid body vector in the physical space. The mass inertia matrix, $I_m = T^T M R$, can be used to apply gravity or other body loads. M is the mass matrix in the physical system.

3.14.1 Inertia Tensor, $I_v = T^T R$

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 lsdyna results. Also, Sierra/SD outputs the fixed interface modes first, while LS-Dyna puts them last. The model is shown in Figure 3.33.

The overall comparison of the values is very good with a relative L2 norm about 6%. Figure 3.34 compares the values of the matrix. There are 3 rigid body modes (corresponding to each of the three translations). There are 10 fixed interface modes and 12 constraint modes, for a total of 22 columns in the inertia tensor. There is significant difference for mode 10, but that is expected because it is the last mode, and the next mode is very near in frequency.

3.14.2 Mass Inertia Matrix, $I_m = T^T M R$

The Mass Inertia matrix, I_m, is determined by a comparison with an independent MATLAB^{TM} 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.

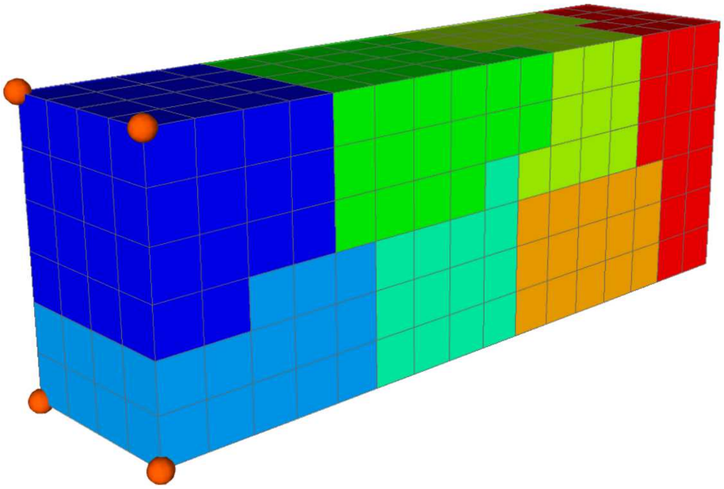


Figure 3.33: LS-Dyna and Sierra/SD Inertia Tensor Model. The model is colored by the parallel decomposition.

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 3.35.

10. Results are compared in serial and in parallel.

These steps found in the Matlab script, `massInertiaTensorCompare.m`.

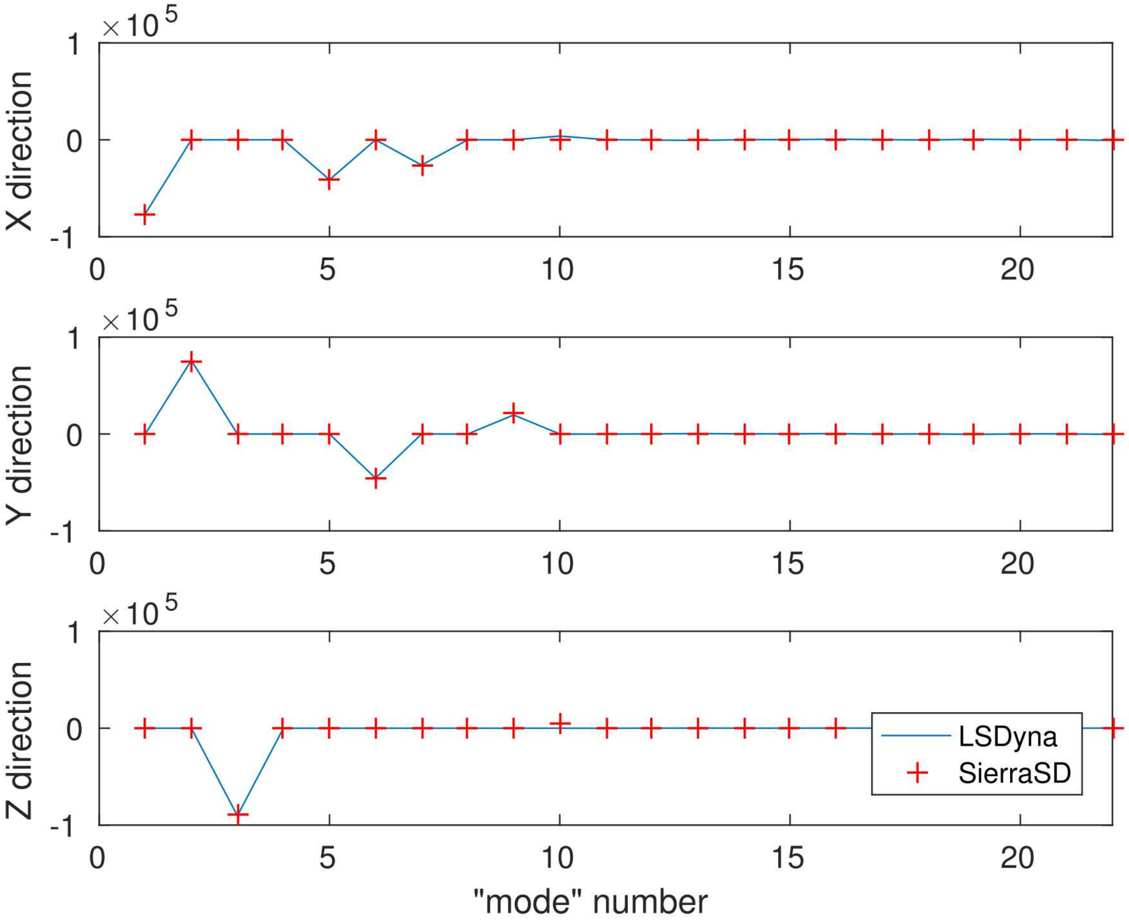


Figure 3.34: LS-Dyna and Sierra/SD Inertia Tensor Terms

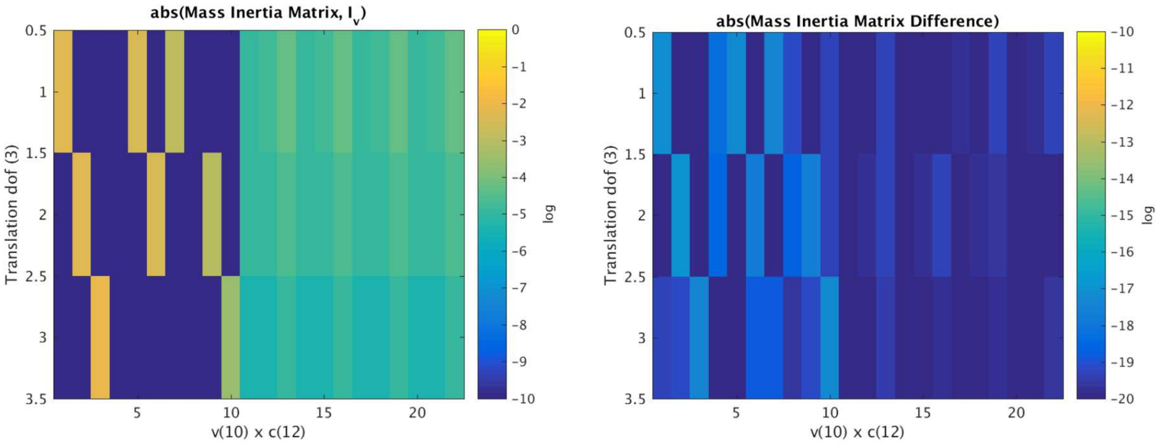


Figure 3.35: Mass Inertia Matrix. Values (left) and Differences (right).

For input see Appendix A.46.

3.15 Nastran/SierraSD Interoperability with Superelements

3.15.1 Needs and Requirements

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 3.15.7 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.

3.15.1.1 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. With the exception of the superelement capabilities, these lie outside the scope of this set of testing.

3.15.2 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 3.36. 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 eigenmodes 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.

Table 3.3 provides a comparison of the frequencies for vibration of the structure.

Figure 3.37 compares the modal FRF solutions for the Sierra and Nastran solutions.

The model is considered suitable for evaluation.

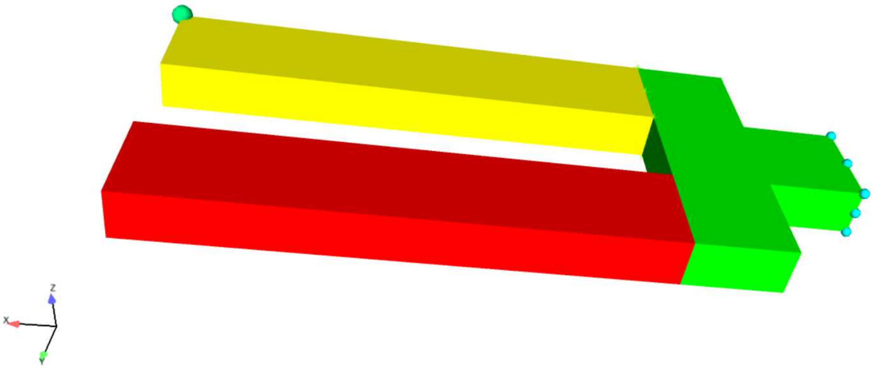


Figure 3.36: Tuning Fork Model

#	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 3.3: Vibrational Frequency Comparison

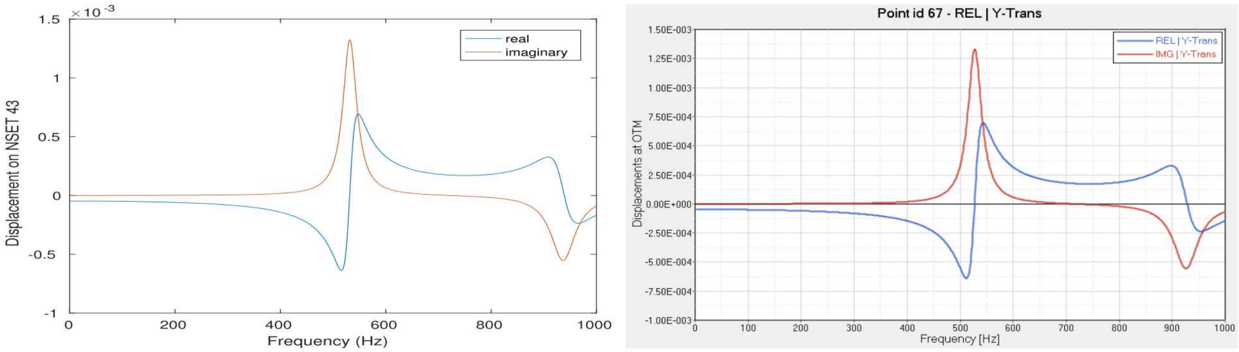


Figure 3.37: FRF Solutions with 3% damping. Sierra/SD and Nastran.

3.15.3 Superelement Reduction and Insertion

3.15.3.1 Sierra/SD Model 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 3.38. Results of the eigen analysis, compared with the full model, are shown in Table 3.4. 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 3.4: Eigen Value Comparison - SierraSD full model and with ROM

3.15.3.2 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`.

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 3.38: Running Sierra/SD solution with Superelement

2. The CB reduction input is generated in `cbr.bdf`. This requires using the `EXTSEOUT` card in the case control section. Also required is the definition of a `BSET` card that contains the interface nodes (a-set dofs) to be constrained during the dynamic reduction step. 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 is run to execute an eigen solution step (`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 7 (MSC Nastran 2017 Superelements User's Guide). Results of the eigenanalysis for the full model and the residual model with superelement are shown in Table 3.5. The results are practically identical.

Figure 3.39 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.

3.15.4 Using Sierra/SD Superelements in Nastran

It is also informative to compare the eigenanalysis results 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 eigenanalysis step. In the second case, DMIG matrices are generated entirely within 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 3.5: : Eigenanalysis Comparison – MSC Nastran Full Model with ROM

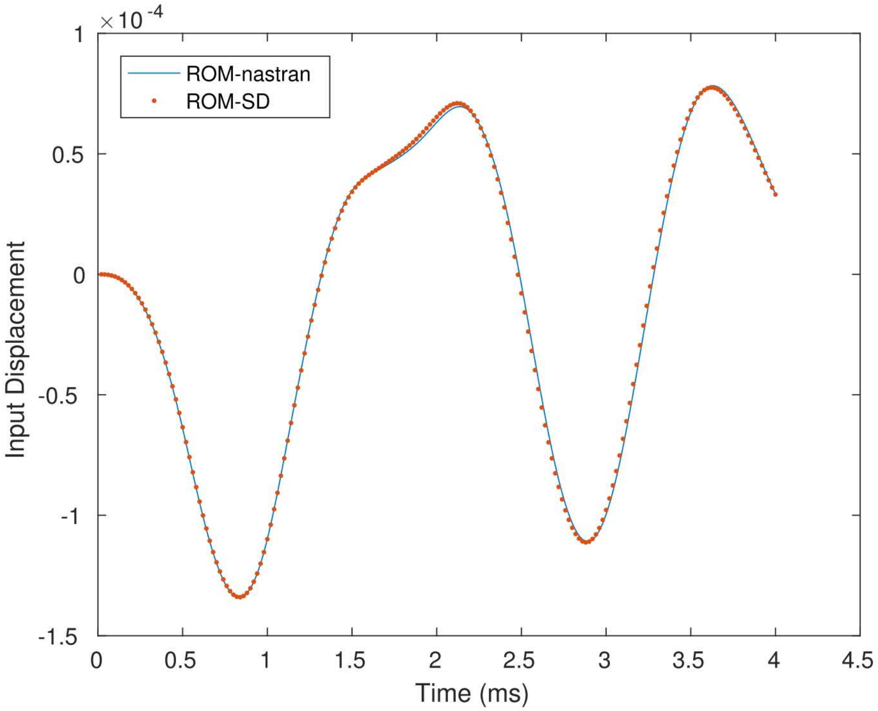


Figure 3.39: Modal Transient Comparison. The input displacement of the Sierra/SD and MSC/Nastran ROM on a Sierra/SD residual

These results, shown in Table 3.6, 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 3.40. The ouput 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 3.6: Eigenanalysis Comparison – Sierra/SD -generated DMIG and Nastran-generated DMIG. Residual and Superelement are employed in each analysis.

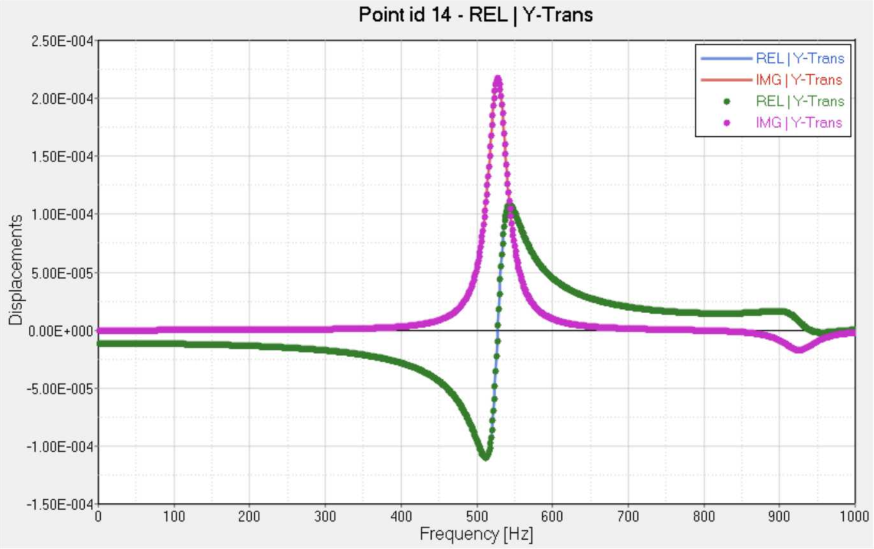


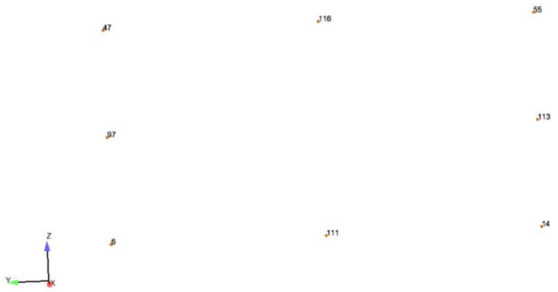
Figure 3.40: Modal FRF Comparison: Full Model (solid line) and Residual Model with Superelement (solid markers)

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

[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.

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 3.41 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.



```
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 3.41: 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.

The original BDF file must be modified as follows.

1. Copy original, and remove the five elements in the ROM region.
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 3.7 provides a comparison of the full Nastran model with the eigen solution using a reduced order model from Sierra/SD. The solution with four fixed interface modes provides good accuracy.[5]

3.15.5 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,

[4] Direct Matrix Input at Grid points

[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.

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 3.7: Eigen Value Comparison - Nastran full model and with Sierra/SD ROMs.

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 3.8. 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 3.8: Comparison of Nastran and Sierra/SD Eigenvalues using Nastran Superelement

3.15.6 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.

Figure 3.42 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 3.43 illustrates the deformation of the full model, compared with the residual and superimposed superelement.

3.15.7 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.

3.15.7.1 Eigen Problem

The nastran eigen problem of the entire model may be evaluated by running:

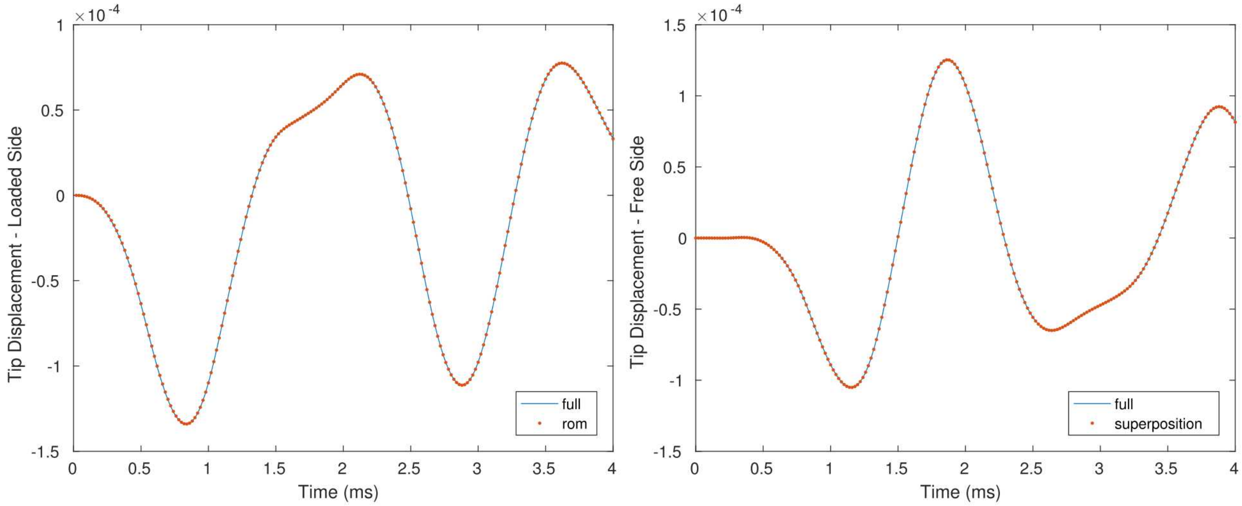


Figure 3.42: Comparison of Output Displacements. The plot on the left compares displacements of the full and reduced order models at the input location. The plot on the right compares displacements on the unloaded tine after the **selem_superp** tool is used to extract the displacement from the reduced model.

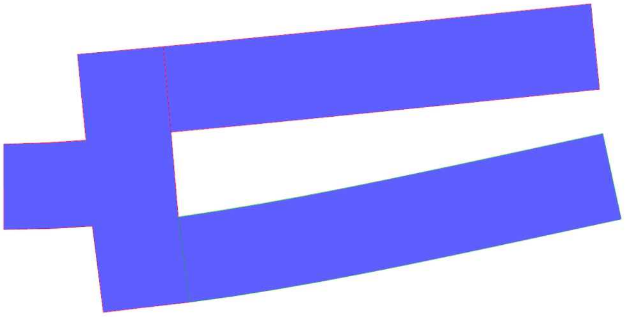


Figure 3.43: Superposition Solution and Full Deformation, $t = 2\,ms$. The full model (in blue background) is compared with the residual and the post-processed superelement.

```
workstation> nastran tuningfork.bdf
```

The resulting output in `tuningfork.f06`, may be evaluated for the appropriate normal mode frequencies.

3.15.7.2 Modal FRF

The eigen problem 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.

3.15.7.3 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 3.15.7.1), but with the ROM of the right tine. Comparison of the modal frequencies provides validation of the analysis.

3.15.7.4 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 3.15.5.

For input see Appendix A.47.

Chapter 4

Sierra/SD Contact, Constraints and MPCs

4.1 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 4.1. 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.

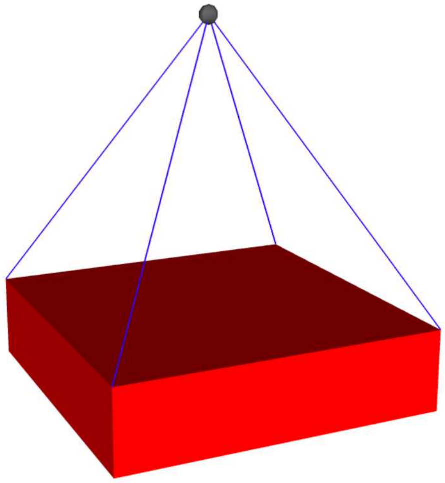


Figure 4.1: Model for Parallel Distribution of Load through Rbars

For input deck see Appendix A.1.

4.2 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 4.2. 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 4.2. 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 4.3. 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 4.3. There are 6 constraint equations for each of these connections. 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 4.4 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 A.2.

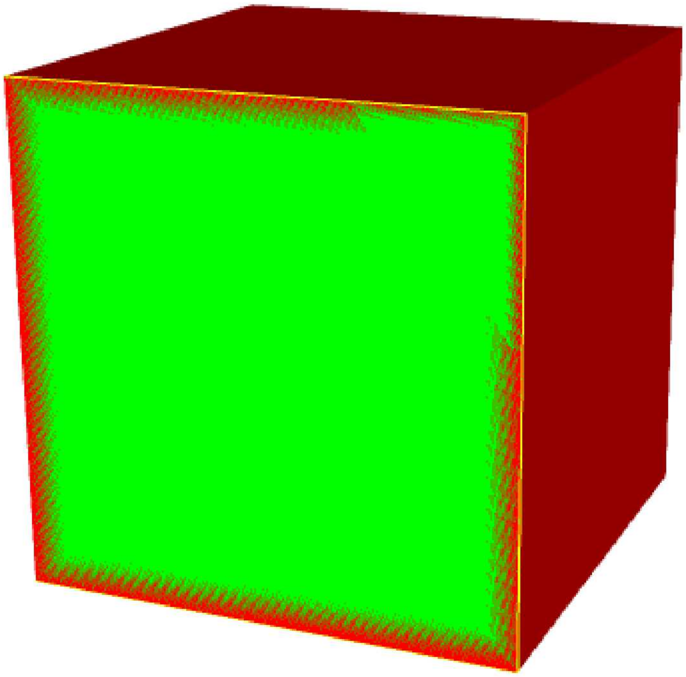


Figure 4.2: A model of a single hex.

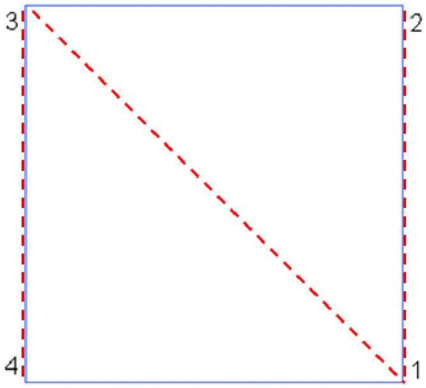


Figure 4.3: A wireframe view of the sideset used for the rigidset in Figure 4.2.

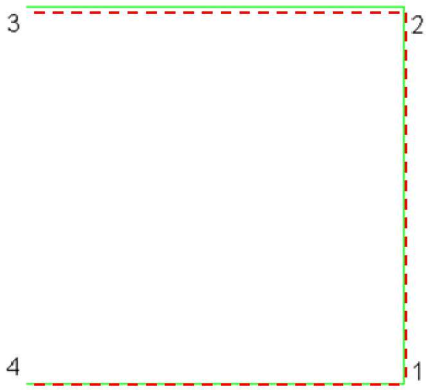


Figure 4.4: A wireframe view of the block of beams used for the rbar collection in Figure 4.2.

4.3 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 4.5 through 4.12. 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 4.13 shows mode 15 of the solution, with a large degree of deformation.

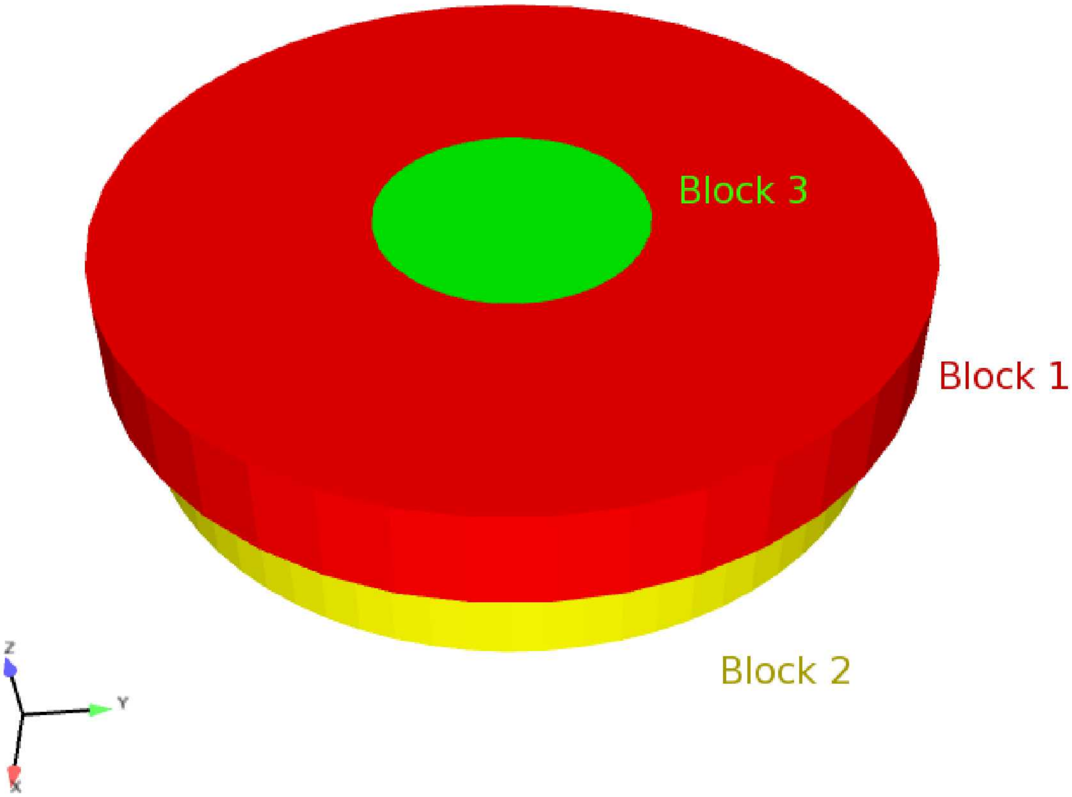


Figure 4.5: All three blocks from an above angle.

For input deck see Appendix A.3.

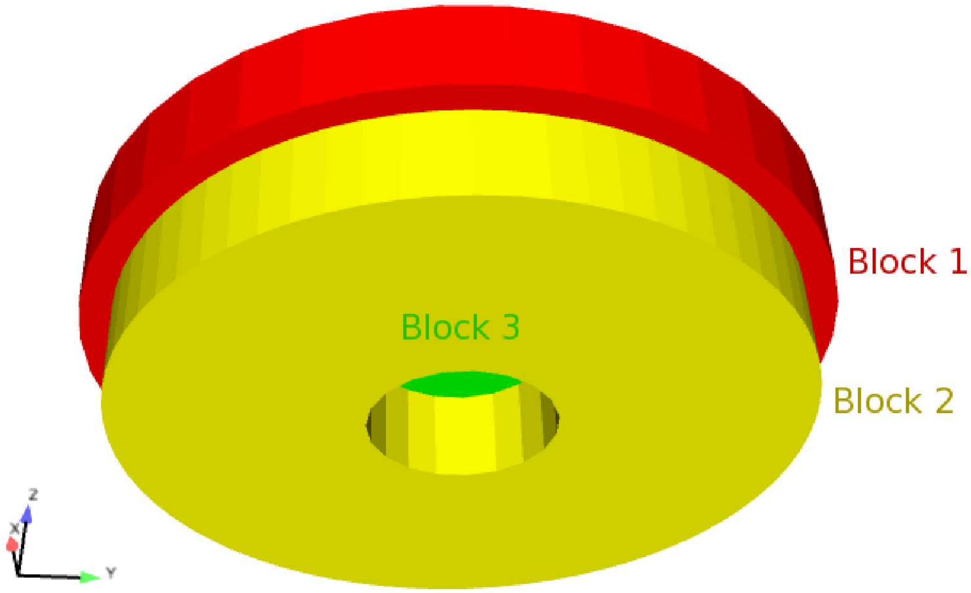


Figure 4.6: All three blocks from a below angle.

4.4 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 master-slave pairings. Evidence based usage guidelines for tied data are provided based on the results.

4.4.1 Description of the Test

Three load cases are considered: A gravity load on a cantilever beam (Figure 4.14(a), a bar fixed at one and with a traction load on the other (Figure 4.14(b), and free-free eigen. 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.

The mesh is generated with Hex8 elements. The geometries used are pictured in figure 4.15. The top mesh is a contiguous mesh to be used as a comparison baseline, and refined significantly for a "truth" solution. The middle mesh uses straight interfaces between the

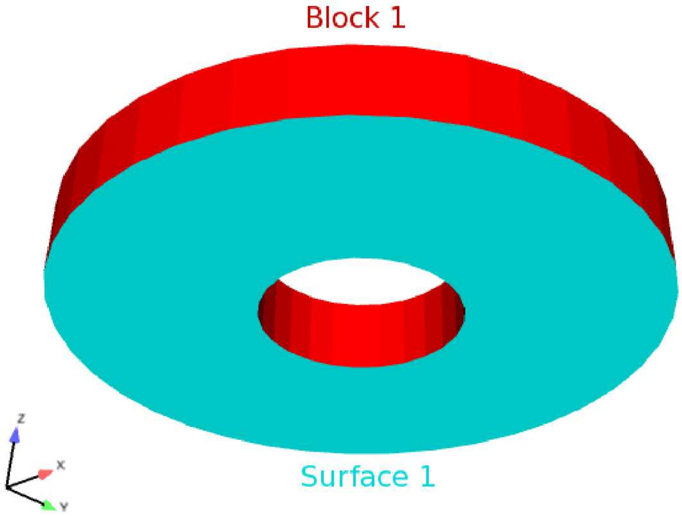


Figure 4.7: Block 1 and Surface 1.

block partitions. The bottom mesh uses curved interfaces between the block partitions.

4.4.2 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.

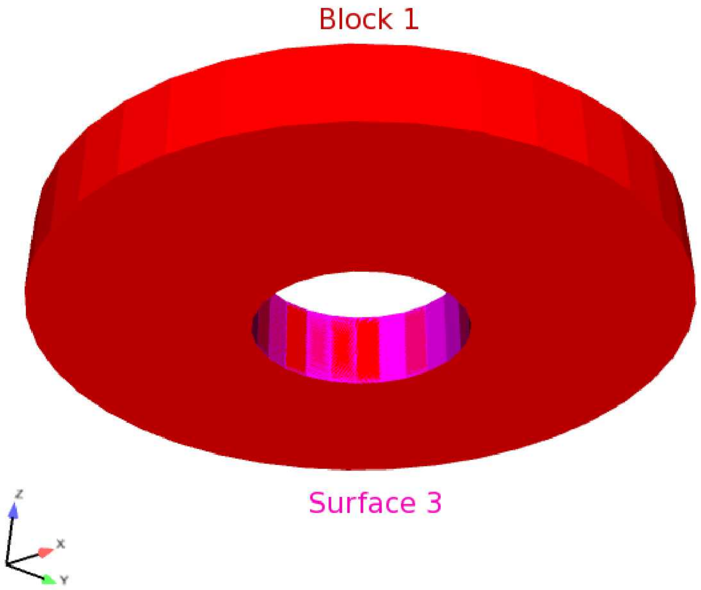


Figure 4.8: Block 1 and Surface 3.

4.4.3 Evaluation of Free-Free Eigen Load Case

The bar is constrained to deform inplane 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 4.16(a) (535.5 Hz), (b) (1272.6 Hz), and (c) (1453.9 Hz).

4.4.3.1 Convergence Rate for Eigen Values

The mesh convergence for the first three flexible modes are shown in Figures 4.17(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.

4.4.3.2 Invariance to Rigid Body Rotation

Figure 4.18 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

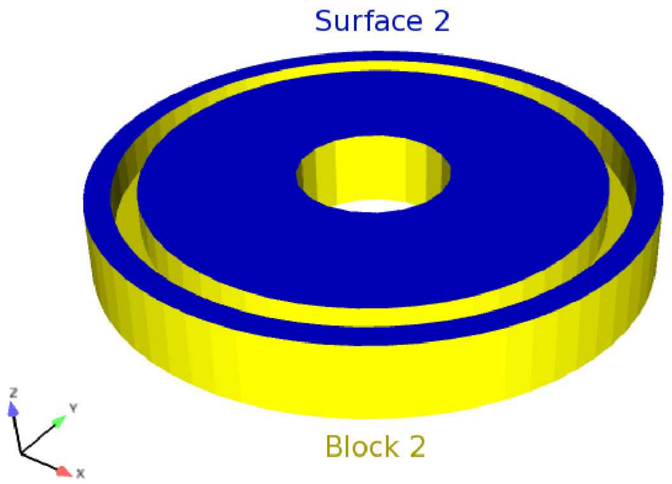


Figure 4.9: Block 2 and Surface 2.

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 gapped constraints is quite substantial.

4.4.3.3 Effect of Master/Slave Interaction Pairing

For optimal accuracy, it is imperative to chose the correct surface for master/slave interactions in tied data. The previous results were made with the recommended setting of using the finer meshed surface as the slave surface (nodes), and the coarser surface as the master (faces). The master and slave surfaces are selected by the order of surfaces in the tied data section of te input deck. As an example, the below syntax selects the nodes of surface 101 as the slave nodes and the faces of surface 100 as the master nodes.

```
TIED DATA
  SURFACE 100, 101
END
```

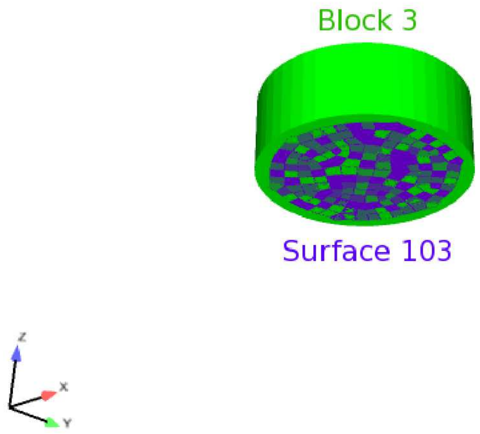


Figure 4.10: Block 3 and Surface 103.

The opposite master slave pairing is given by:

```
TIED DATA
  SURFACE 101, 100
END
```

If master and slave surfaces are selected properly, MPC_Status, which is specified by the slave_constraint_info output option, will appear as shown in figure 4.19(a). If the wrong surface is chosen as master, then the results appear as shown in figure 4.19(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 master 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 slave surface lie on the faces of the master surface, but does not impose the same requirement on the nodes of the master surface. If both surfaces are at approximately the same refinement, it does not matter which side is the master surface, but when the master surface is at a significantly higher refinement than the slave surface, there will be some faces of the master surface which are not constrained to any slave nodes, and are allowed to move without any stiffness contribution from the slave 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.

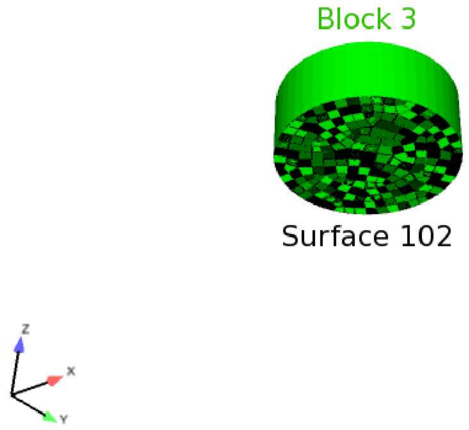


Figure 4.11: Block 3 and Surface 102.

The eigen mode convergence with reversed master/slave interactions is shown in Figure 4.20(a)-(c). With the non-recommended master/slave 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 master/slave pairings.

4.4.4 Evaluation of Cantilever Beam Static Results

The result for contiguous cantilever beam is shown in Figure 4.21.

4.4.4.1 Convergence Rate

The mesh convergence of tip displacement for the cantilever beam is shown in Figure 4.22. 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.

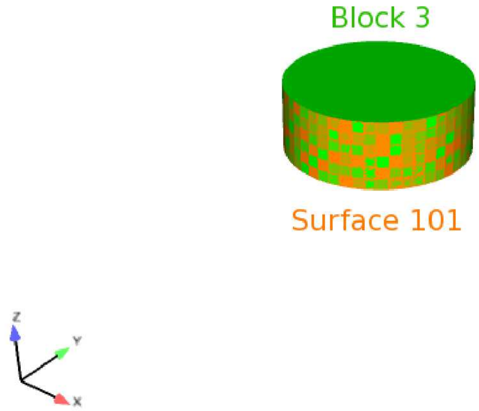


Figure 4.12: Block 3 and Surface 101.

4.4.4.2 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 4.23. With symmetric constraints there is no convergence to the correct solution. As seen in Figure 4.24 the symmetric contact interfaces cannot bend, leading to a completely spurious displacement and stress result.

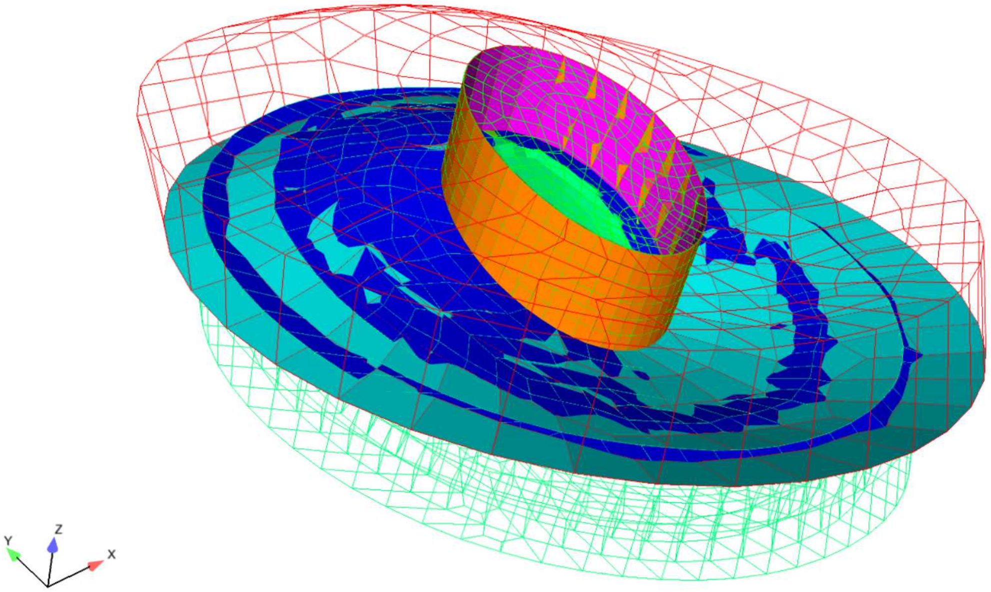


Figure 4.13: Mode 15. Showing sideset Tying

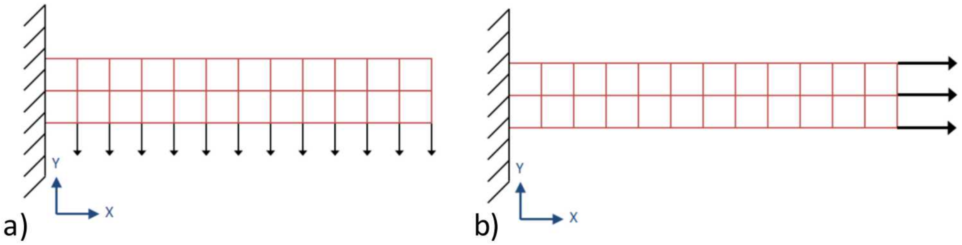


Figure 4.14: Beam under (a) gravity loading and (b) traction loading.

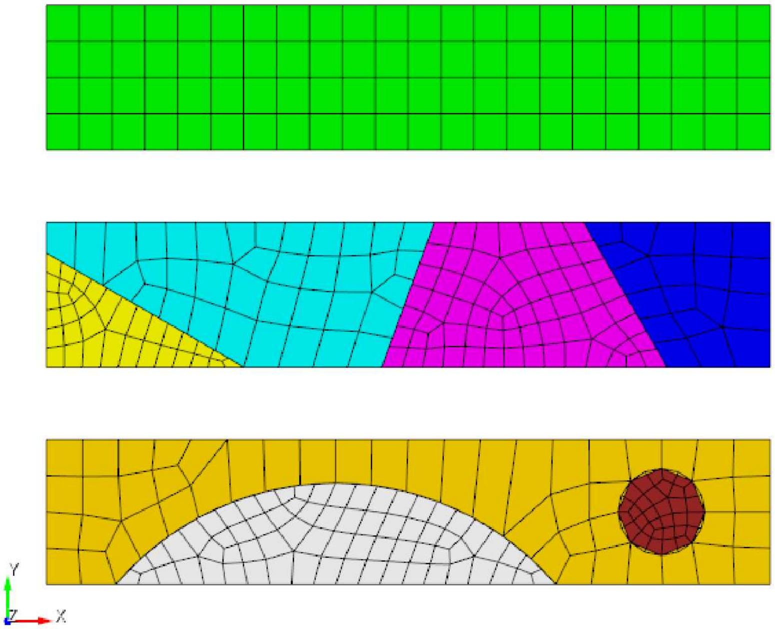


Figure 4.15: Mesh Geometry

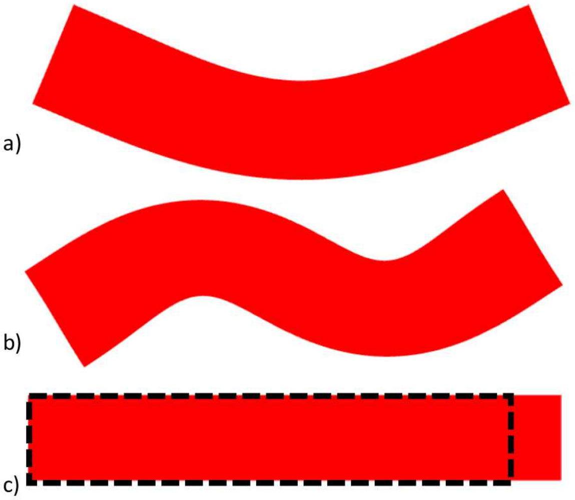


Figure 4.16: Flexible mode shapes (a) mode 1 (b) mode 2 and (c) mode 3 (non-uniform axial elongation)

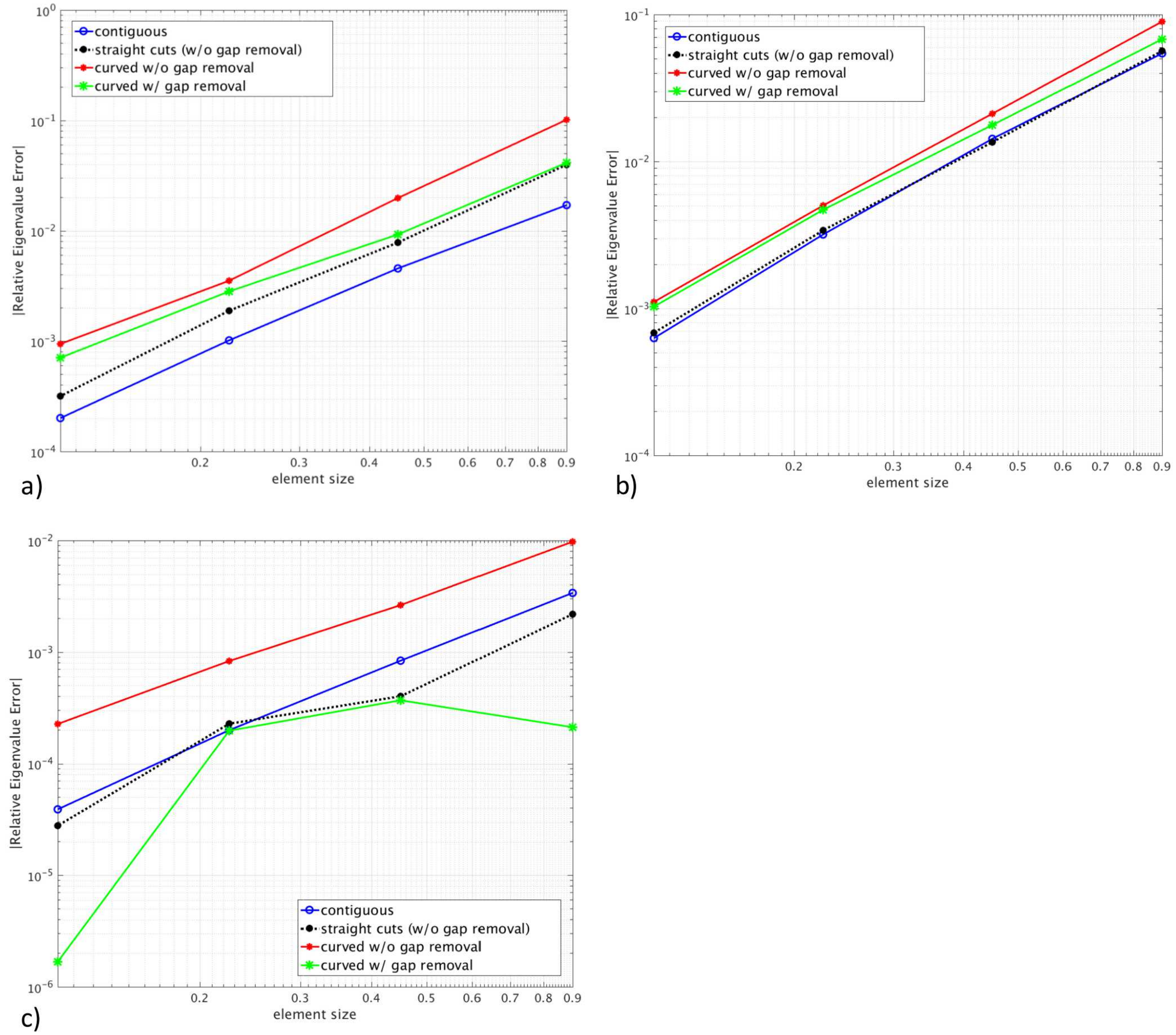


Figure 4.17: Convergence rates for flexible modes. (a) First elastic mode converges to 534.5 Hz (b) Second elastic mode converges to 1272.6 Hz (c) Third elastic mode converges to 1453.9 Hz.

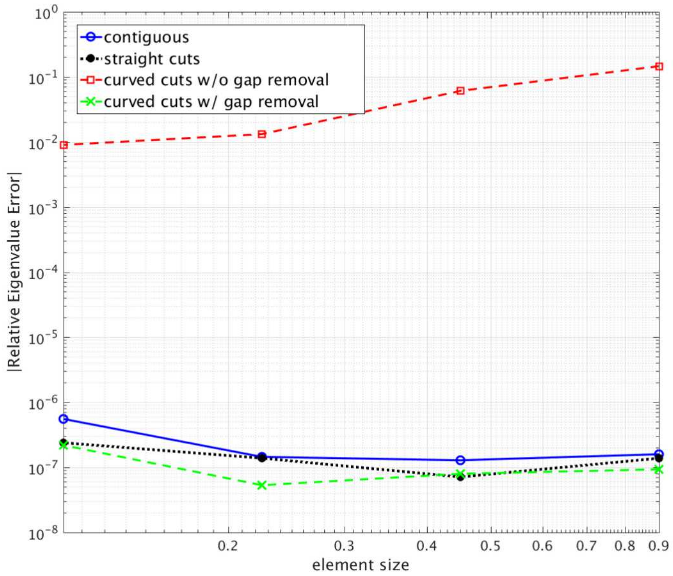


Figure 4.18: Error in rigid body rotation mode relative to first flexible mode

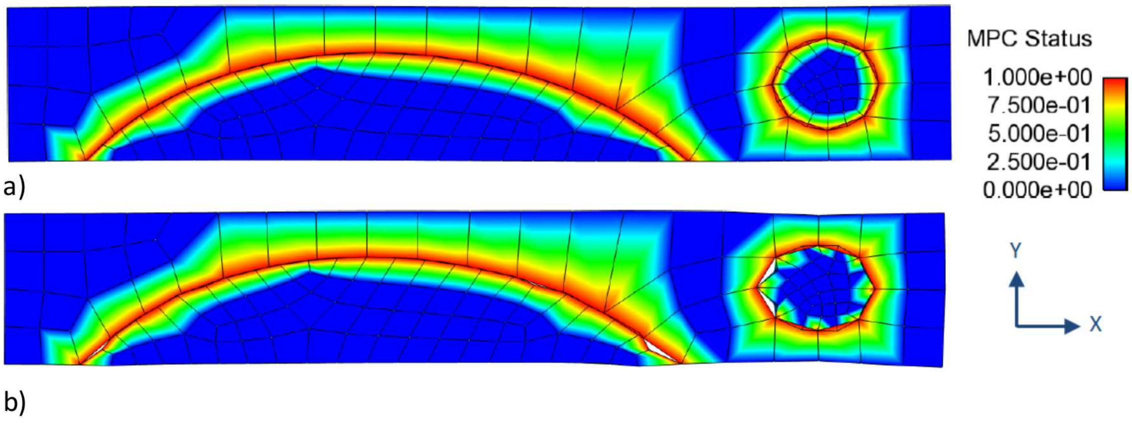


Figure 4.19: MPC Status (a) correct and (b) incorrect.

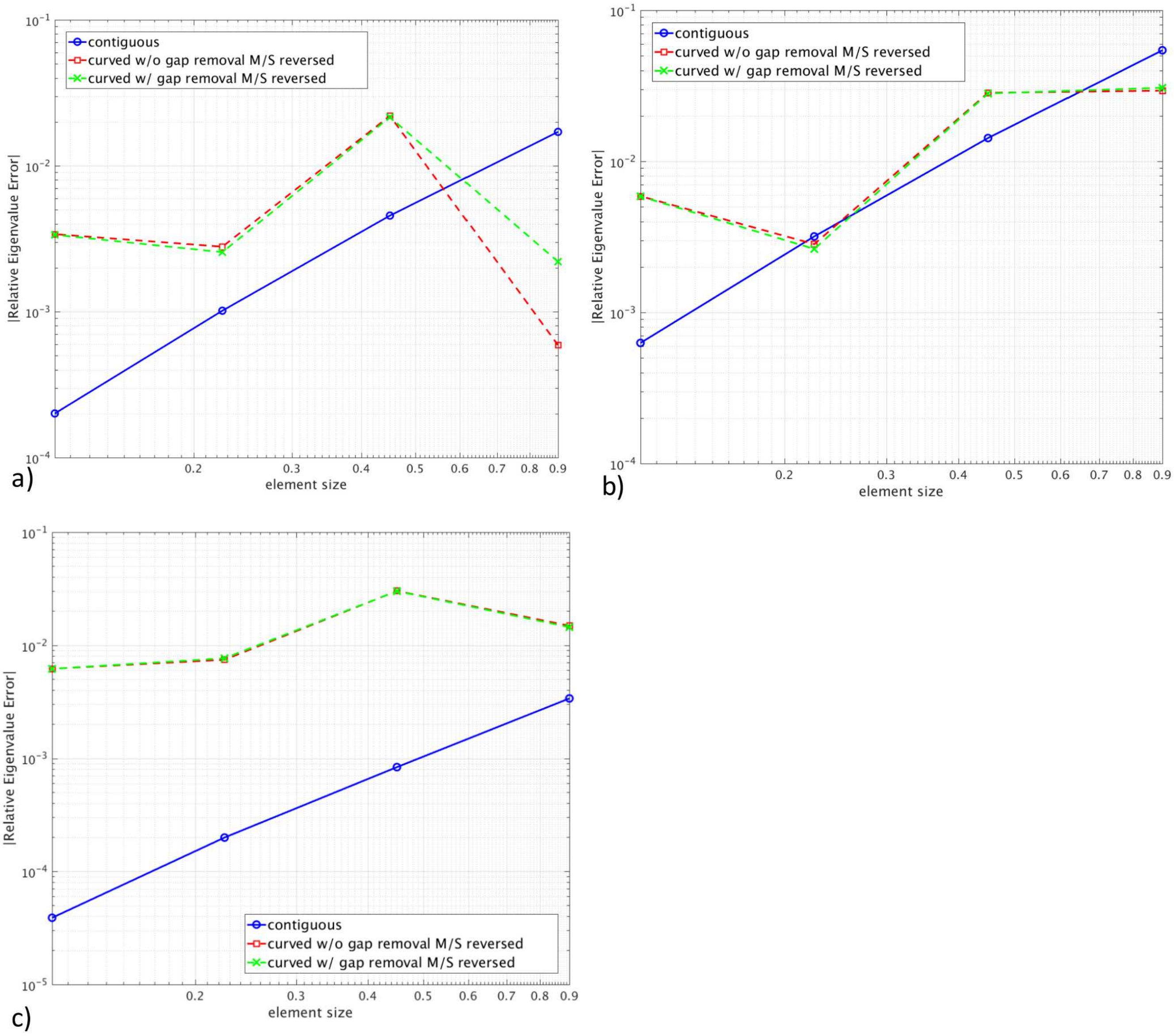


Figure 4.20: Flexible mode convergence rates with reversed master/slave. (a) First flexible mode converged to 534.5 Hz. (b) Second flexible mode converged to 1272.6 Hz. (c) Third flexible mode converged to 1453.9 Hz.

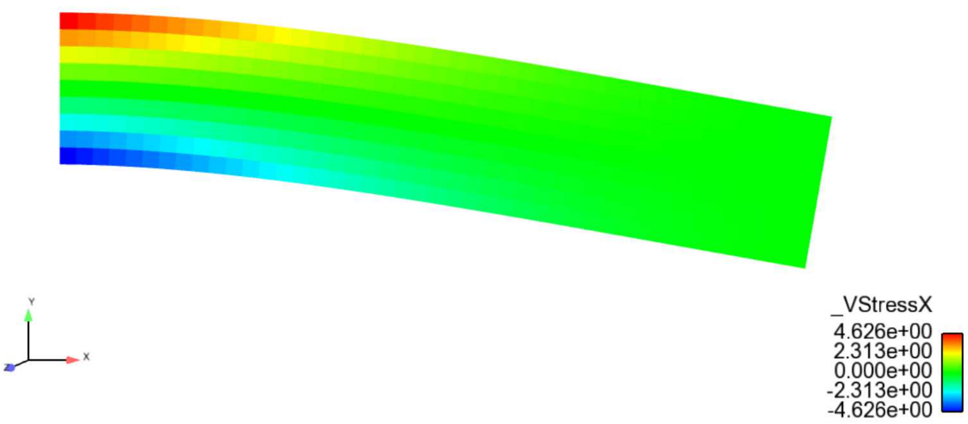


Figure 4.21: Cantilever Beam Deformed result (greatly magnified)

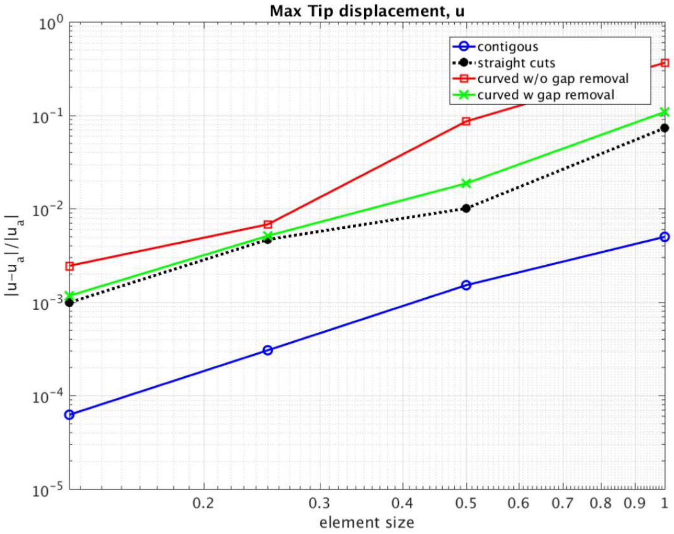


Figure 4.22: Cantilever Beam Convergence For Tip Displacement

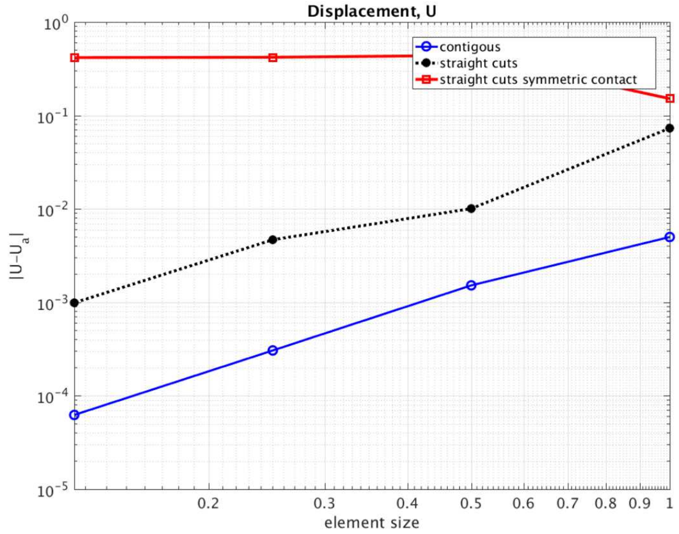


Figure 4.23: Cantilever Beam Convergence with Symmetric Constraints

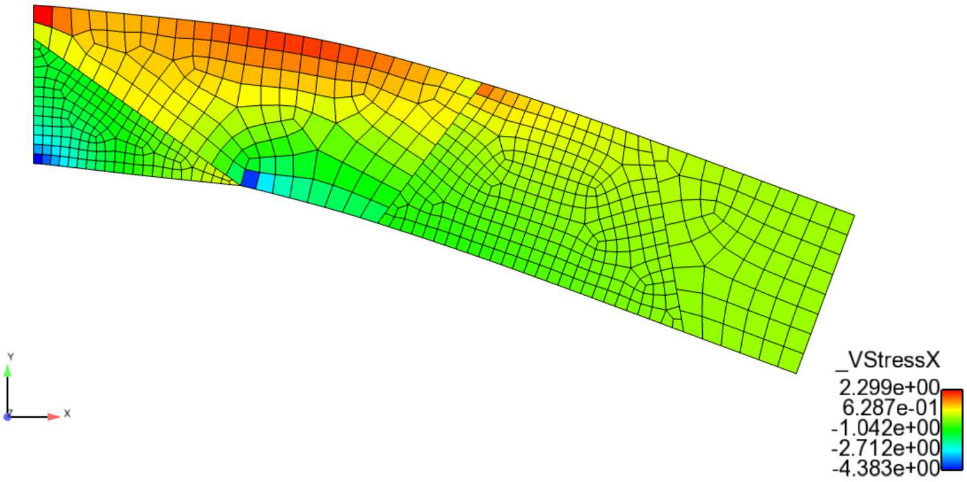


Figure 4.24: Incorrect Cantilever Beam Result with Symmetric Contact

4.4.5 Evaluation of Axial Pull Results

The axial pull results should produce an exact uniform XX direction stress of 1000, however the nature of tied contact constraints produces artificial stress concentrations at the contact interface. The convergence of stress is shown in Figure 4.25 and the distribution of stress on two mesh resolutions shown in Figure 4.26. The magnitude of tied data stress concentrations are not remedied by mesh refinement. The stress concentrations do become somewhat more localized with mesh refinement.

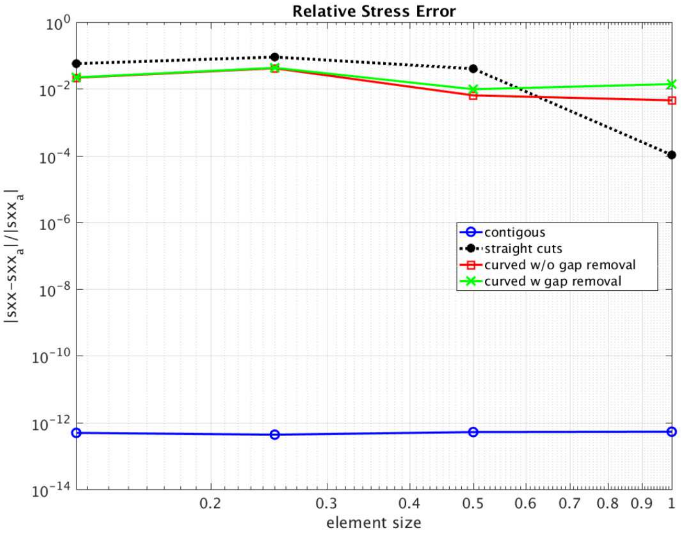


Figure 4.25: Axial Pull Convergence for Maximum Stress

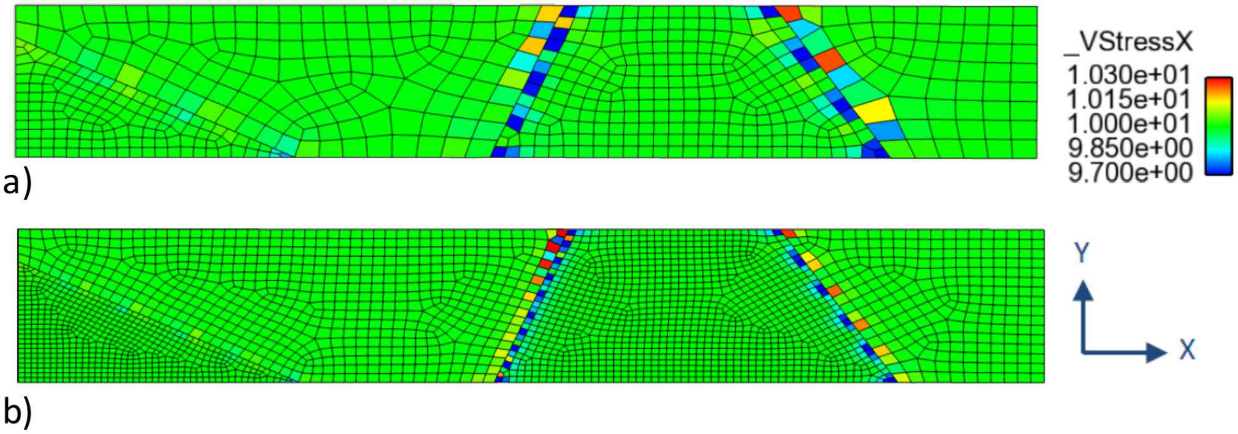


Figure 4.26: Spurious Local Stress Concentrations with (a) coarse and (b) fine meshes.

4.4.6 Usage Guidelines

Used carefully, tied data can greatly simplify the model creation process by eliminating the need for contiguous meshes. However, there are a number of 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 'slave nodes' of tied data interactions and the coarser surface the 'master 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 A.48

This page intentionally left blank.

Chapter 5

Sierra/SD Solutions

5.1 Waterline of a ship

A code to code comparison was performed between Sierra-SD and the Navys 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 5.1. Three key parameters were analyzed between the two codes the 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 5.1. For input deck see Appendix A.20.

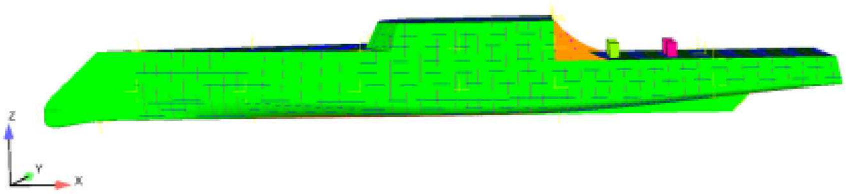


Figure 5.1: uhwmGeometry

Table 5.1: 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

5.2 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.

Figure 5.2 shows the final displacement of the deformed beam.

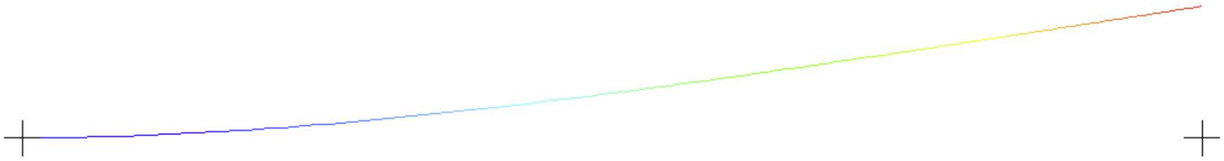


Figure 5.2: Verification Problem - Beam

Figure 5.3 shows the time history result for the problem, solved at three different timesteps.

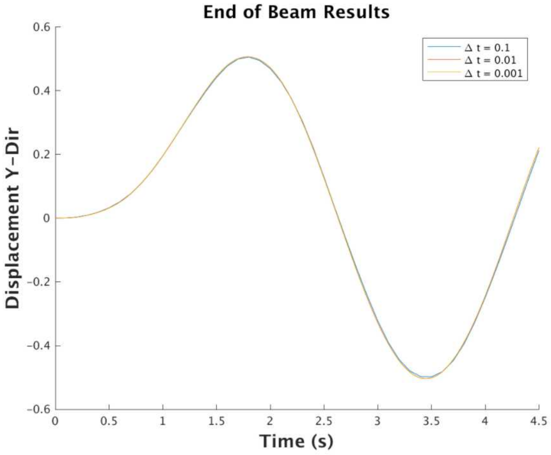


Figure 5.3: Time History of Transient Verification Problem

Figure 5.4 shows the Richardson Convergence of the problem. Convergence values $n = 2$ implies second order convergence.

For input deck see Appendix A.21.

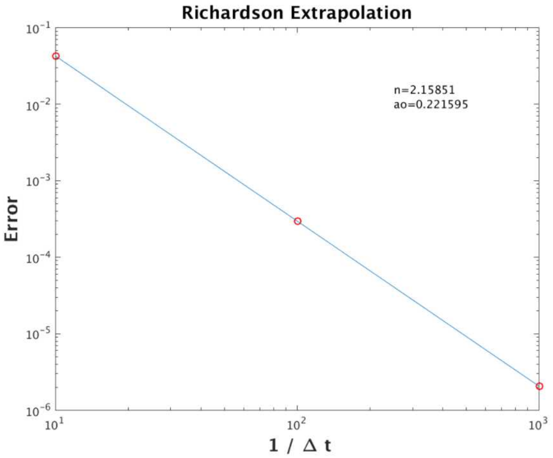


Figure 5.4: Richardson Extrapolation of Transient Verification Problem

5.3 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 5.5 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.

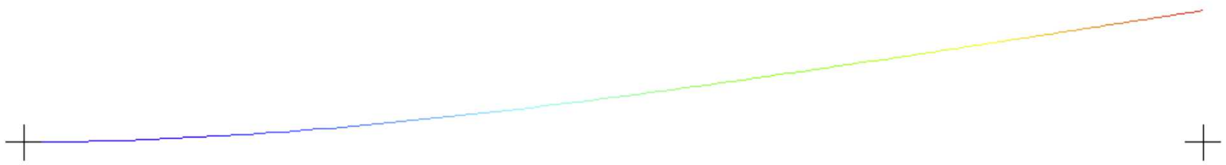


Figure 5.5: Verification Problem - Beam

Figure 5.6 shows the time history of the beam end point for the problem for three time steps.

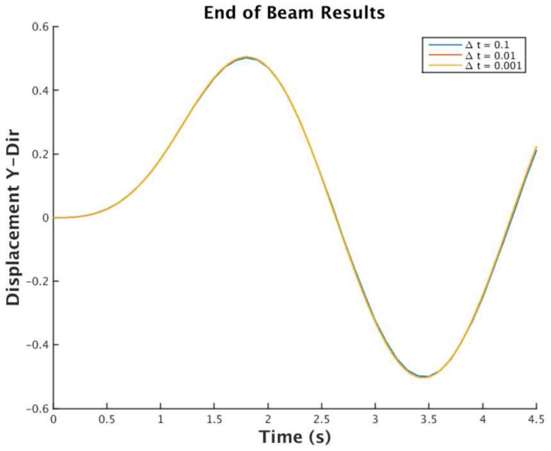


Figure 5.6: Time History of Modal Transient Verification Problem

The Richardson convergence was used as a means of determining the order of convergence for the modal transient method. Figure 5.7 shows the Richardson Convergence of the problem. Convergence values $n = 2$ implies second order convergence. This results is very 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 5.8 shows the difference in displacement at the end of the beam between the direct transient method and the modal transient method for $\Delta t = 0.001$.

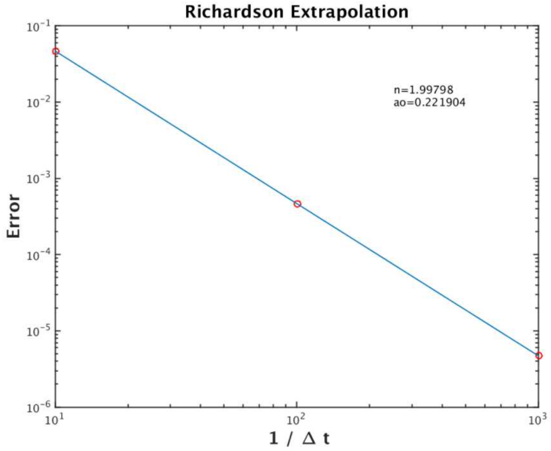


Figure 5.7: Richardson Extrapolation of Modal Transient Verification Problem

It should be noted that 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 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.

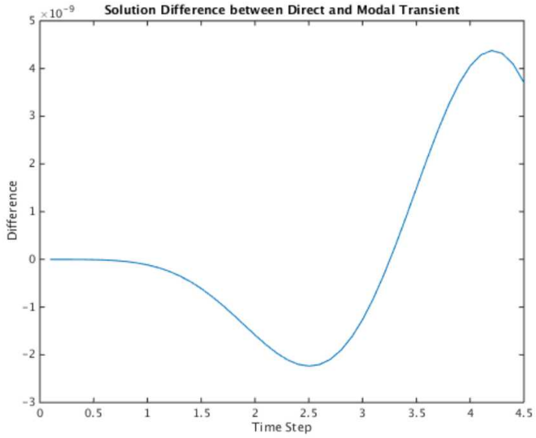


Figure 5.8: Displacement Difference for Modal and Direct Transient Solutions

For input deck see Appendix A.22.

5.4 Transient Restart

Analysts should be able to restart from any of the transient analysis capabilities into any of the others.[1] 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 quite loose on this check. Each integrator provides a somewhat different solution (as expected).
8. Visual comparison of the results.

Table 5.2 indicates the tests that have been performed. Nonlinear transient as the first integrator is not currently tested. Figure 5.9 provides the data for the second row of Table 5.2, which includes all cases where the direct transient was the first integrator. Likewise, Figures 5.10 through 5.11 show data for modaltransient and explicit as the initial integrators.

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

Table 5.2: Tested restart capabilities for transient integrators in Sierra/SD.

For example inputs, see Appendix A.23. The model is shown in Figure 3.9.

[1] The qmodal solutions cannot be restarted.

Figure 5.9: 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 5.10: 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.

Figure 5.11: Restart from Explicit Transient Analysis. In each case, $30\,ms$ of analysis is completed using a Explicit transient run, and is followed by a restart.

5.5 Q Modal Transient

The qmodal solution method is a standard Newmark-beta integrator that is applied in parallel on the modal space. It is limited to loadings 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.

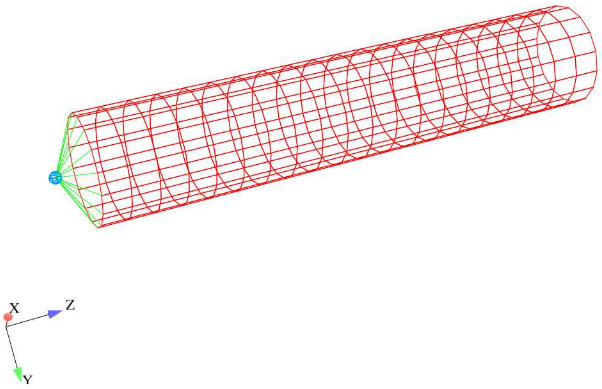
2. Repeat the above, but eliminate the rigid body motion. A comparison with the standard modal solution provides the verification.

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. All are run in parallel. The model is shown in Figure 5.12. It consists of a thin cylinder with beams on one end attaching to a large mass. The loading is applied to the mass.

Figure 5.12: Q Modal Verification Model



5.5.1 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,

$$\begin{aligned}
a(t) &= \frac{F_o}{m} && (5.1)\\
v(t) &= \int_o^t a(t')dt' && (5.2)\\
&= \frac{F_o}{m}t && (5.3)\\
d(t) &= \int_o^t v(t')dt' && (5.4)\\
&= \frac{F_o}{2m}t^2 && (5.5)
\end{aligned}$$

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 5.13 compares the analytic and numerical solutions for displacement. Figure 5.14 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.

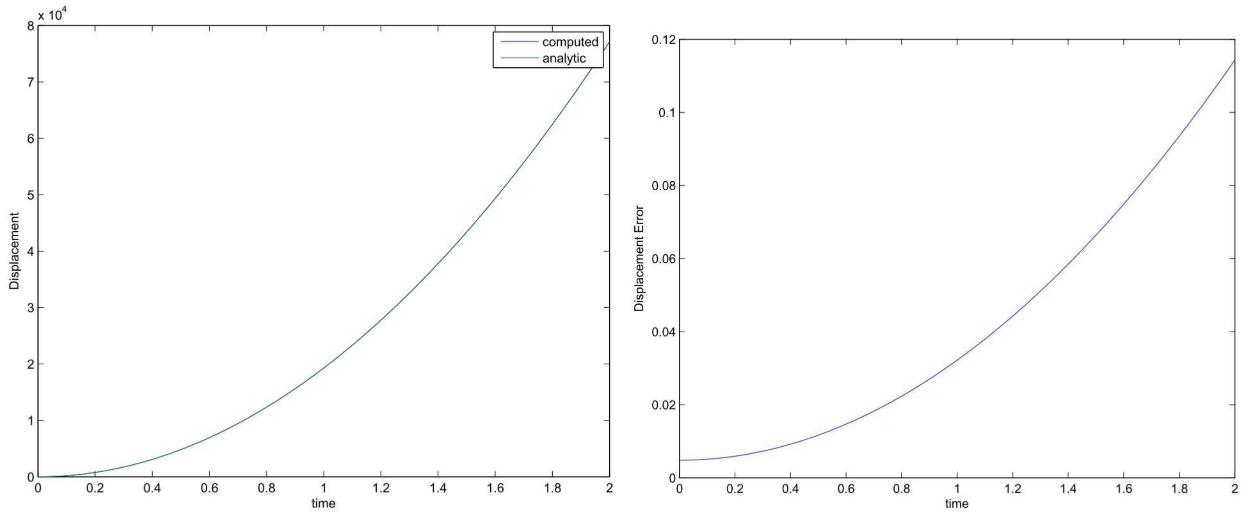


Figure 5.13: Response of Rigid Body Mode

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.

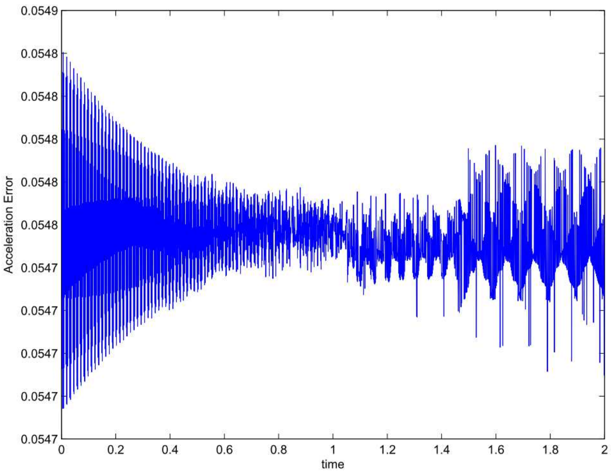


Figure 5.14: Acceleration Response of Rigid Body Mode. The analytic acceleration is a constant of $F/m \approx 38561.8$. The error in the figure is much smaller than this, and represents the elastic response of higher frequency elastic modes, that are just slightly active in the analysis.

5.5.2 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.

$$
\begin{aligned}
a(t) &= F_o \alpha \cos(\omega_i t) && (5.6)\\
v(t) &= \int_o^t a(t')dt' && (5.7)\\
&= F_o \omega_i \alpha \sin(\omega_i t) && (5.8)\\
d(t) &= \int_o^t v(t')dt' && (5.9)\\
&= \frac{F_o}{\omega_i^2}\alpha(1-\cos(\omega_i t)) && (5.10)
\end{aligned}
$$

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 5.15.

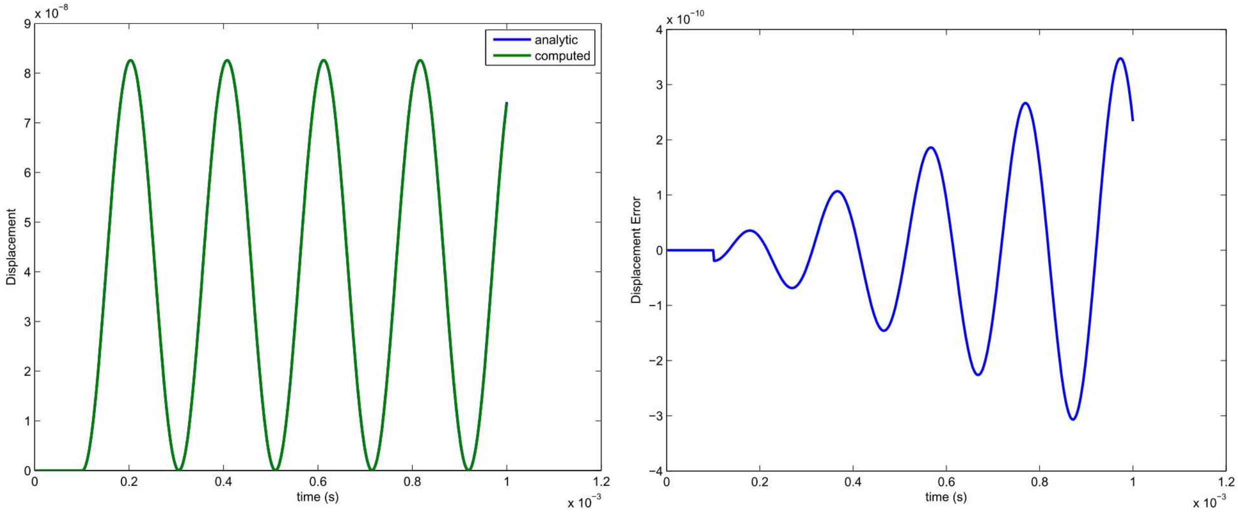


Figure 5.15: Step Function response of Undamped Oscillator

5.5.3 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,

$$x(t) = A\left(1 - e^{-\zeta\omega_i t}\frac{\sin\left(\sqrt{1-\zeta^2}\right)\omega_i t + \phi}{\sin(\phi)}\right) \tag{5.11}$$

Results for the analytic and numeric solutions are shown in Figure 5.16.

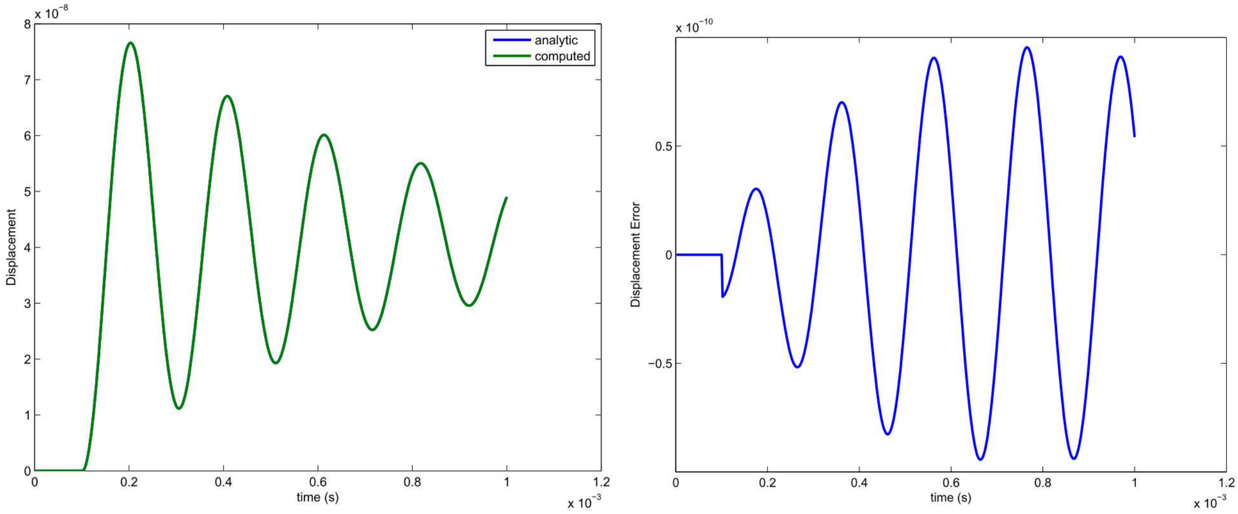


Figure 5.16: Step Function response of Damped Oscillator

5.5.4 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 modal-transient method. This boundary condition is essentially an impulse which causes a linear increase in displacement. There is no difference between the modaltrans and qmodaltrans solutions.

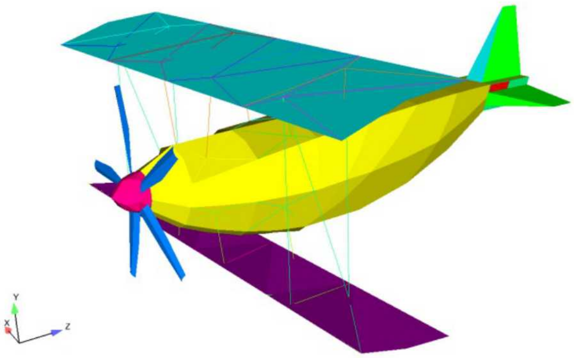
For input deck see Appendix A.26.

5.6 Q Modal Frequency Response

The qmodal solution method is a standard Newmark-beta integrator that is applied in parallel on the modal space. It is limited to loadings that are space/time separable, and all outputs must fit on single processor. Qmodalfrf is used to perform a fast modal frf analysis on a subset of the mesh.

The verification runs modalfrf and qmodalfrf on the biplane model seen in Figure 5.17. The load is constant and applied to the underside of one of the tail fins.

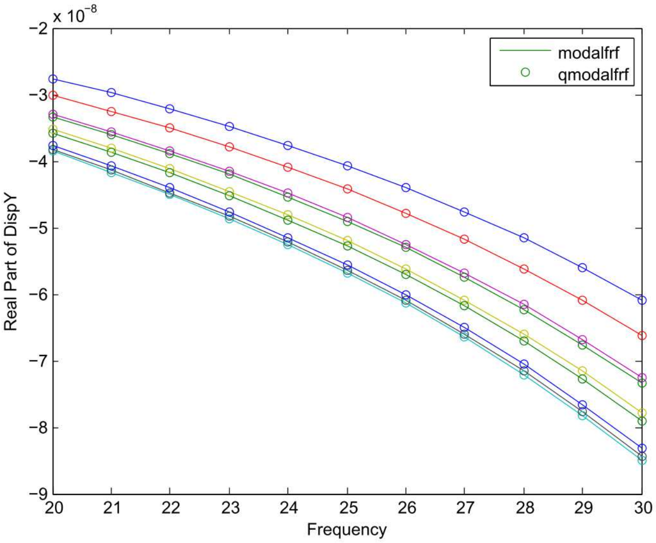
Figure 5.17: Q Modal Frf Verification Model



The verification for qmodalfrf assumes that the modalfrf results are correct. This test compares the frequency response function output of the modalfrf and qmodalfrf tests to ensure that they are the same. The displacement of modalfrf and qmodalfrf must match. The real displacement in the y direction versus frequency for the modalfrf and qmodalfrf tests are shown in Figure 5.18. Note how the lines (modalfrf) and circles (qmodalfrf) match.

For input deck see Appendix A.27.

Figure 5.18: Real Displacement in the Y Direction Versus Frequency for the Modal Frf and Q Modal Frf Analyses



5.7 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 5.19 and a steel sphere with no added mass. The fluid is an acoustic medium.

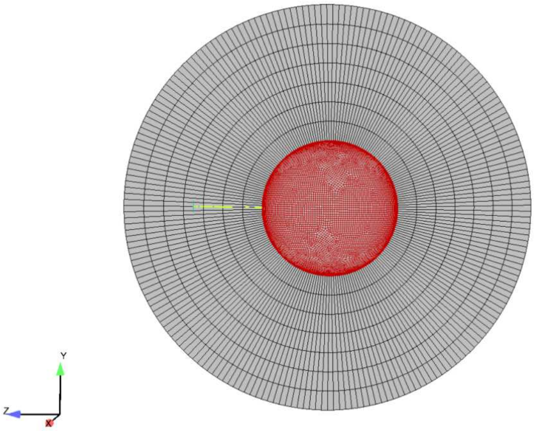


Figure 5.19: Model of the hollow sphere and spring submerged in water

5.7.1 Analytical solution

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 and massless with no damping $\omega =$:

$$\omega = \sqrt{\frac{K}{m}} \tag{5.12}$$

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

$$\omega = \sqrt{\frac{K}{m + m_a}} \tag{5.13}$$

The formulas for various shapes are documented.[8] In the case of a spherical structure, the added mass is given by:

$$m_a = \frac{2}{3}\pi\rho a^3 \tag{5.14}$$

The first mode computed in SierraSD should match the analytical solution.

There are several parameters for this test. The steel sphere is fixed in the x and y direction so that displacements were only allowed in the z direction (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. The node attached to the end of the spring is fixed for no 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.

5.7.2 Computational Approach

The eigenfrequencies of the coupled structural acoustic system require computation of a quadratic eigenvalue problem (QEVP).

$$(K + C\lambda + \lambda^2 m)u = 0 \tag{5.15}$$

where K is the stiffness matrix, m the mass matrix, and C is the gyroscopic coupling matrix. The solutions to the equation include only purely imaginary eigenvalues $\lambda = i\omega$. Two methods for computing QEVP are applied. SA_eigen uses a modal projection to reduce the dimension of the problem and solve dense QEVP using LAPACK routines. The QEVP/Anasazi method is a custom solution solving the full problem without the approximation of a modal projection.

Shell elements were investigated in particular detail. When analyzing shell elements the thickness of the inner sphere was as thin as 0.0001. The test was run using SA-eigen and Anasazi. The number of modes, refinements, and test parameters varied to maximize accuracy of the results. A collection of results using SA-eigen and Anasazi with various thicknesses is shown in Table 5.3.

Table 5.3: 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

A visual representation of the frequencies in SierraSD using SA-eigen and Anasazi compared to the analytical solution is shown in Figure 5.20. 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 QEVP/Anasazi method tracks the analytical solution very well. The QEVP/SA_eigen solution is not as accurate, but the solution is still much better then the coupled solution.

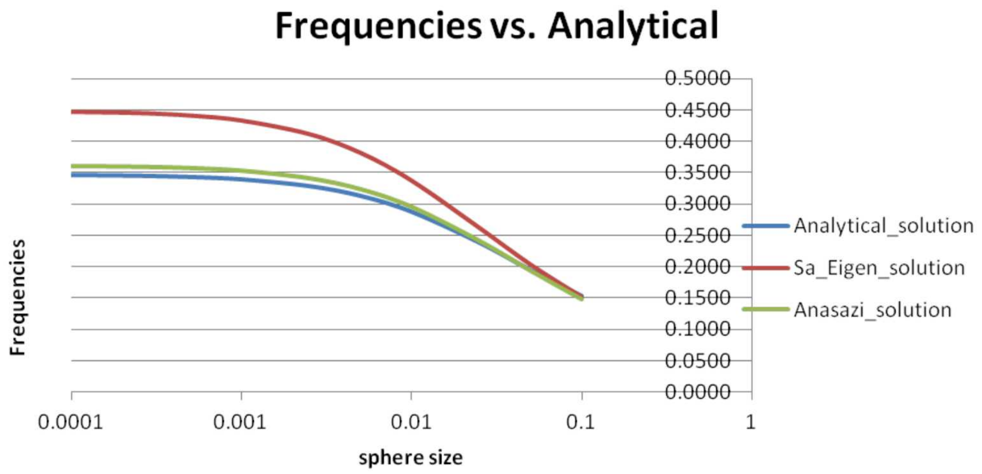


Figure 5.20: Frequencies in SierraSD compared to the analytical solution.

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 then 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 5.21 shows the comparison of having an added mass to your system and shows the results between SierraSD and the analytical solution.

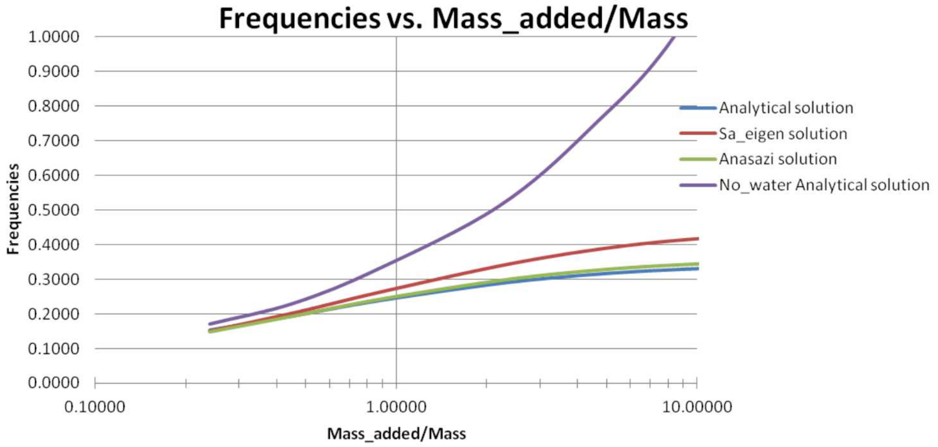


Figure 5.21: Frequencies in SierraSD vs the mass ratio of the system.

This model was also investigated using using hexahedral and tetrahedral 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 A.31.

5.8 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 5.22 is an illustration of the model.

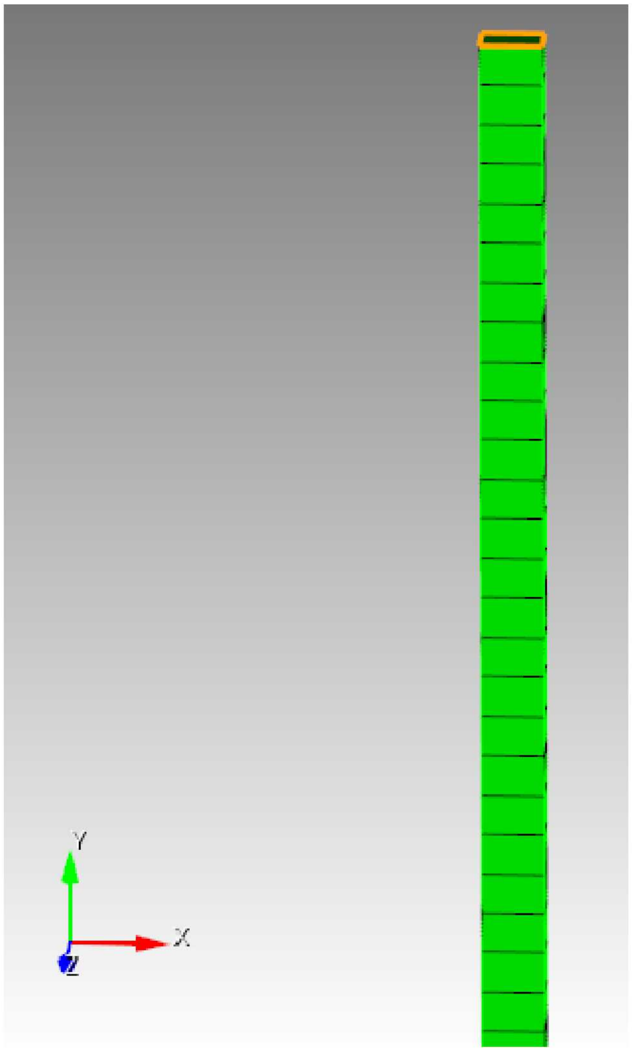


Figure 5.22: 1D FSI Plate Shell Model in SD

This model consists of a single structural rectangular plate 1.5 in by 1 in. The plate consists of a QuadT shell element with a thickness of one. 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. The physical properties used for the analysis were in imperial units. 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 an intergration time step of 1.313e−5 sec.

The Sierra-SD results were compared to and verified against published results.[10] Figure 5.23 of the y component of velocity versus time reproduces the published results.

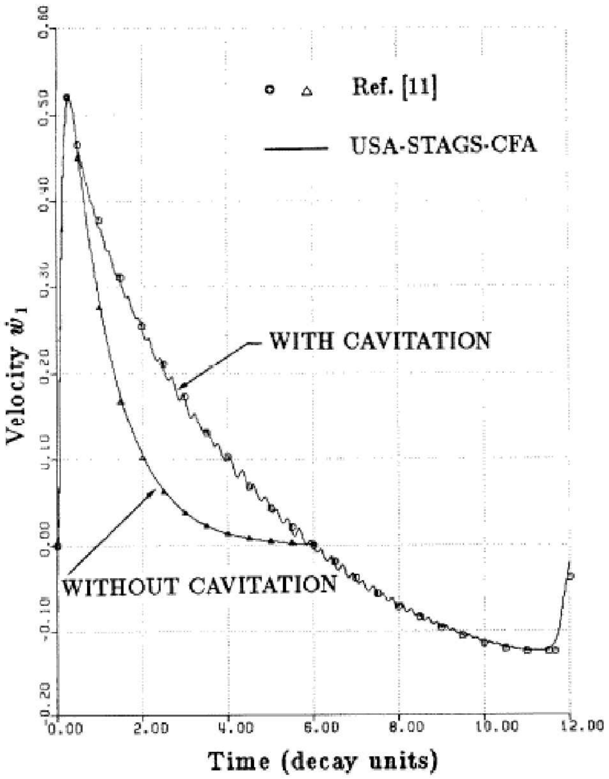


Figure 5.23: Velocity vs Time, Results from Felippa and DeRuntz

The model without cavitation was reproduced in Sierra-SD and compared to.[10] This is shown in Figure 5.24. 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 A.32.

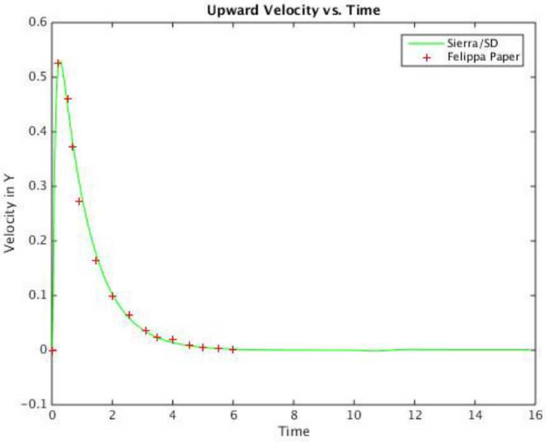


Figure 5.24: Velocity vs Time, Results from Sierra-SD

5.9 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 5.25.

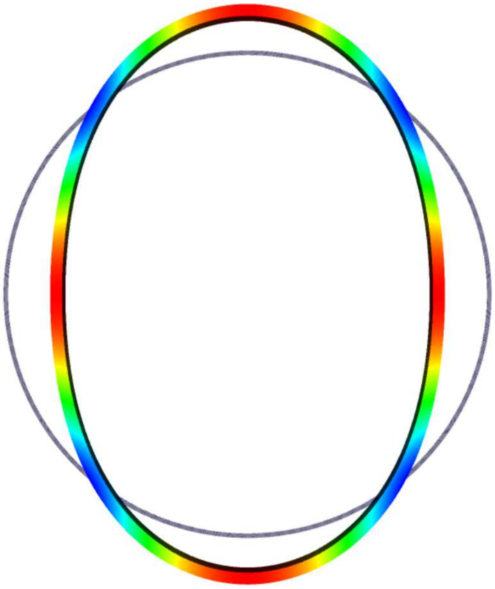


Figure 5.25: Buckling Ring Example. Model parameters

Diameter: 40

Material: aluminum

Cross Section I: 1/12

Cross Section Area: 2.0

Cross Section Thickness: 1.0

5.9.0.1 Buckling of a Circular Ring

In this example, we consider buckling of a circular ring subjected to a uniform, external pressure. The critical buckling pressure is given[11] as

$$P_{cr} = \frac{3EI}{R^3} \tag{5.16}$$

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

$$P_{cr} = \frac{3 \times 10^7 \times \frac{1}{12}}{20^3} = 312.5 \tag{5.17}$$

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 A.49

This page intentionally left blank.

Chapter 6

Sierra/SD Element Verification

6.1 Euler Beam 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,

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

Figure 6.1 shows the comparison with the analytic solution for a beam of length $L = 1$, $E = 10e6$, and bending moment $I1 = 0.2$ for a 100 element beam. Figure 6.2 shows the convergence as a function of the number of elements in the beam. The solutions here are performed with a direct solver, sparsepak, and with the GDSW solver with 2 processors. The lack of convergence to the analytic solution is expected, and indicates the increased numerical error as the matrices become more ill conditioned. As the number of elements increases, the matrix condition worsens. Even the serial solver accuracy suffers, but parallel iterative solvers are particularly vulnerable to reduced accuracy for poorly conditioned systems.[1]

In some sense, the lack of convergence is pathological in this example. The exact solution is a cubic, which can be met exactly by a single element of the 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 A.6.

[1] Note that for this example we have used standard solver parameters for GDSW. With care, the solution can be forced to be more accurate.

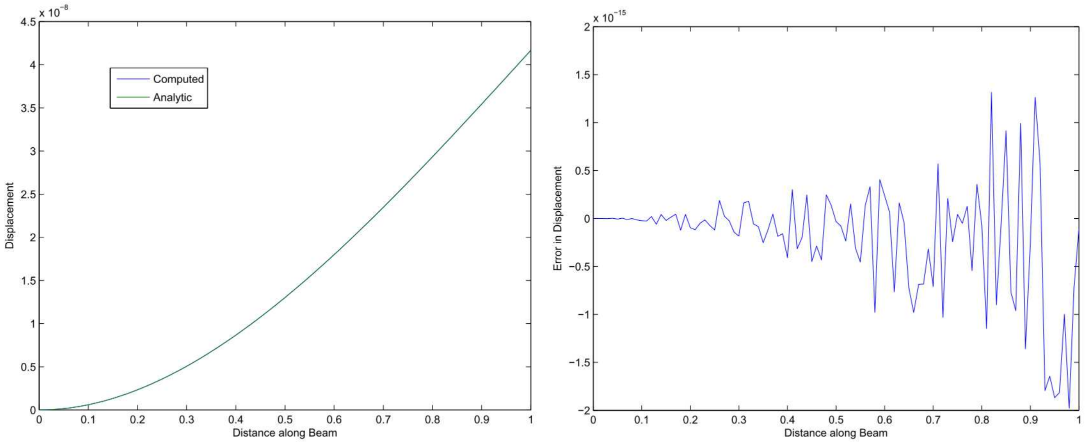


Figure 6.1: Comparison of Beam2 Bending

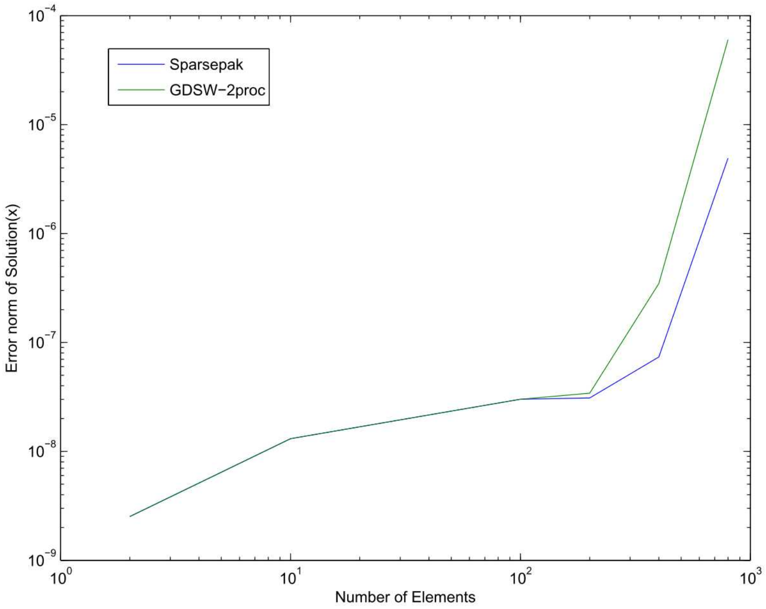


Figure 6.2: Beam2 Bending "Convergence". The plot shows the L_2 Norm of the error in w, divided by the L_2 norm of w as a function of the number of elements. Properly convergent solutions would decrease as the number of elements increase. While this solution is very accurate, it is not converging to the analytic solution as the number of elements increases.

6.2 Euler Beam Properties

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:

Figure 6.3: Geometry of Beam

Table 6.1: 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.

6.2.0.1 Analytical Solution

A MATLAB script was created to calculate the modal frequencies for a single span cantilevered beam using the following formula from Blevins:

$$f_i = \frac{\lambda_i^2}{2\pi L^2}\left(\frac{EI}{m}\right)^{\frac{1}{2}} \tag{6.1}$$

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

6.2.0.2 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 6.2

Table 6.2: 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

Natural frequencies that show N/A are twisting modes. Figure 6.4 shows the differences in calculated natural frequencies.

6.2.0.3 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 6.2 shows the comparison results.

6.2.0.4 References

Blevins, Robert D. “Formulas for Natural Frequencies and Mode Shape “, Krieger Publishing Company, 1984

For input deck see Appendix A.7.

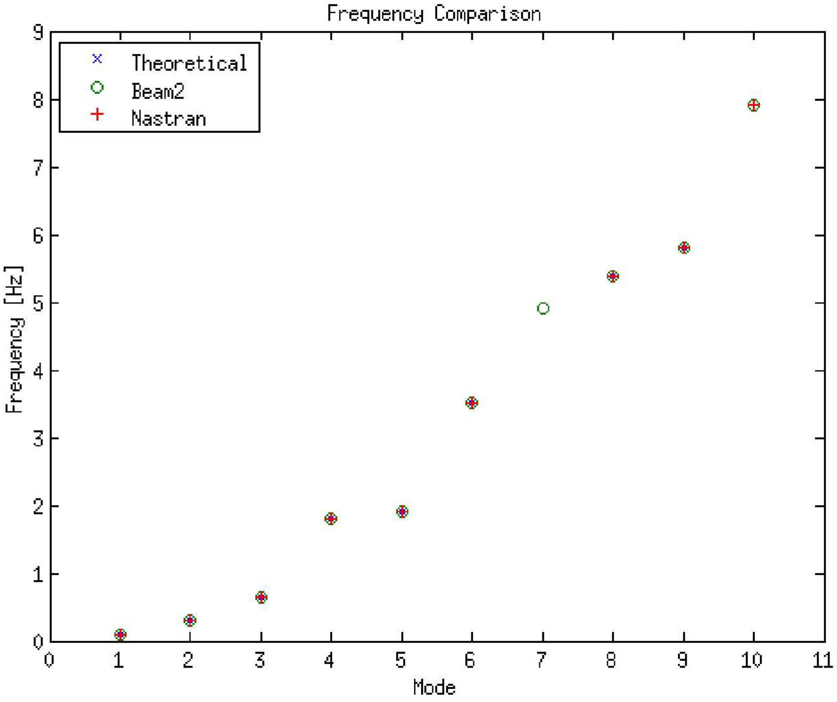


Figure 6.4: Frequency Comparison

6.3 A Navy Beam

The following test was used to verify that Sierra SD is using 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 Nbeam elements with the following geometry:

Figure 6.5: Geometry of Beam

Table 6.3: 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.

6.3.0.1 Analytical Solution

A MATLAB script was created to calculate the modal frequencies for a single span cantilevered beam using the following formula[8]

$$f_i = \frac{\lambda_i^2}{2\pi L^2}\left(\frac{EI}{m}\right)^{\frac{1}{2}} \tag{6.2}$$

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

6.3.0.2 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 6.4

Table 6.4: 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 that show N/A are twisting modes. Figure 6.6 shows the differences in calculated natural frequencies.

6.3.0.3 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 6.4 shows the comparison results. For input deck see Appendix A.8.

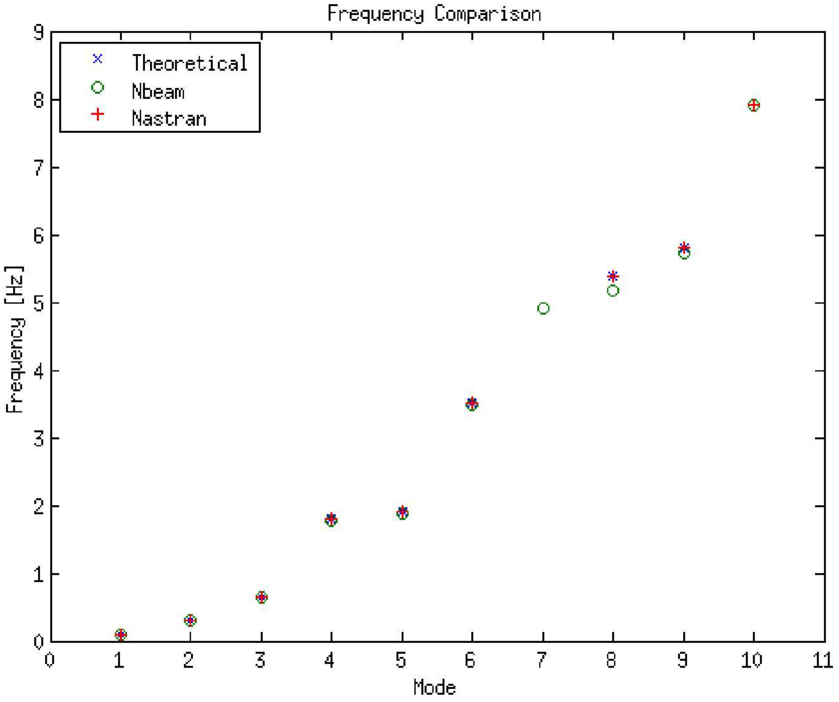


Figure 6.6: Frequency Comparison

6.4 Two Layered Hexshell

6.4.1 Problem Description

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

6.4.2 Verification of Solution

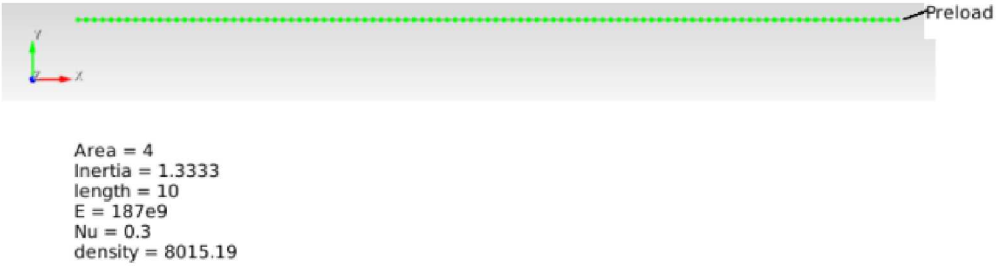
The mesh consists of a hexahedron of dimension $[-1/2, 1/2] \times [-1/2, 1/2] \times [-5/2, 5/2]$. The example is a step in a study of deflection versus layer thickness. Results have been compared to documented results,[1213] in the past. For input deck see Appendix A.9.

6.5 Preloaded Beam

The following test was used to verify that Sierra SD accurately acounts for an axial preload on a beam. This test was verified using three different references, two different analytical solutions (Shaker, 1975), (Carne, 1982), and an abaqus benchmark problem. Beam and shell elements were examined between the two tests. The problem was first analyzed with no preload using the same analytical solution and then modeled to verify that the system is functioning appropriately.

6.5.1 Test One

Test one is a verification of the analytical solution using beam elements. All parameters were incorporated using SI units. The beam parameteres are:



Area = 4
Inertia = 1.3333
length = 10
E = 187e9
Nu = 0.3
density = 8015.19

Figure 6.7: Geometry of Beam

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.

6.5.1.1 Analytical Solution

An axial preload has limited verification due to lack of closed form solutions, however; in the paper (Carne, 1982) an analytical solution can be used. Assuming pinned pinned constraints on the beam the natural frequencies are:

$$V_n = \left(\frac{n\pi}{L}\right)^2 \left(\frac{EI}{\rho A}\right)^{\frac{1}{2}} \left[1 - \frac{PL^2}{EIn^2\pi^2}\right]^{\frac{1}{2}} \tag{6.3}$$

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.

6.5.1.2 Computational Approach

This test case was modeled using Sierra SD.The eigenfrequencies 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 eigenfrequencies. The beam was partitioned into one hundred elements.

There are several parameters for this test. In order 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 6.5.

Table 6.5: 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

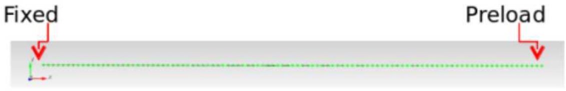
All modes are within 1.5 percent error between the analytical solution and Sierra SD.

6.5.2 Test 2

The following test is used to verify the static preload feature in Sierra SD. An Abaqus benchmark for an Eigenvalue analysis of a beam was found and compared against. The test consists of a cantilever beam, made of one hundred elements and a static axial force applied in the x direction.

6.5.2.1 Beam Elements

The beam was analyzed with and without the static preload. Figure 6.8 shows the geometry of the model.



Axial preload of 44482.
Beam Area=0.00002026
Length of beam 0.127
I1=I2 = 3.26907E-11

Figure 6.8: 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 6.6.

Table 6.6: 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 from Sierra-SD match up well with the benchmark problem in Abaqus.

6.5.2.2 Shell Elements

A similar test was created in Sierra SD using shell elements. This is a cantilever beam, made of 100 elements and a static axial force applied in the x direction, and was used and compared to the Abaqus benchmark. However, modification to the input deck is needed to ensure a similar structure of the model. The Abaqus model has a circular cross section with an inertia equal to $I = (\pi * r^4)/4$. When modeling shell elements a square cross section was used and a corresponding base and height was used to have an equivalent inertia. The inertia for a square cross section is $I = (b * h^3)/12$. Also, the beam was modeled with a pressure load, in order to distribute the load evenly.

The frequencies using shell elements in Sierra-SD and results in Abaqus are compared in table 6.7.

Table 6.7: Results Abaqus vs. Sierra-SD (shells)

	Abaqus	Sierra-SD
Without Preload		
Mode 1	212.4	212.793
Mode 2	1330.8	1327.73
Mode 3	3727.2	3689.16
With Preload		
Mode 1	1137.9	1141.66
Mode 2	3624.4	3621.86
Mode 3	6694.1	6636.3

A graphical representation of the data using shell elements in Sierra-SD and results in Abaqus is shown in Figure 6.9

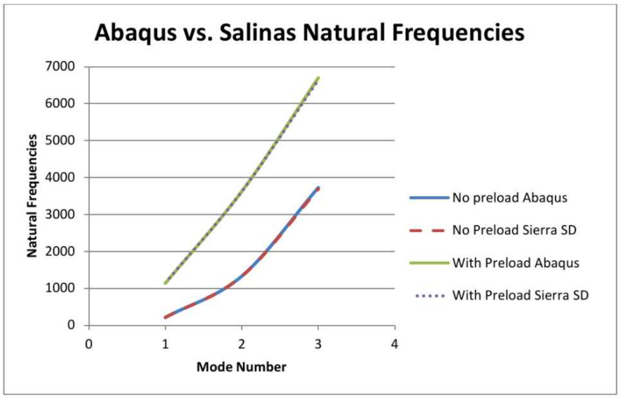


Figure 6.9: Shells Sierra SD vs. Abaqus

Even with slight modificaitons done to the input deck the results when using shell elements in Sierra-SD match up well with Abuqus.

The Abaqus benchmark problem can be found using this URL.

http://www.ilsb.tuwien.ac.at/v6.10/books/bmk/default.htm?startat=ch01s04ach37.html

6.5.3 References

Carne, Thomas G., Donald W. Lobitz, Arlo R. Nord, and Robert A. Watson. "Finite Element Analysis and Modal Testing of a Rotating Wind Turbine." (1982): 8-9. Sandia Report. Web.

Shaker, Francis J. "Effect of Axial Load on Mode Shapes and Frequencies of Beams." Lewis Research Center (1975): 1-9. Web.

For input deck see Appendix A.11.

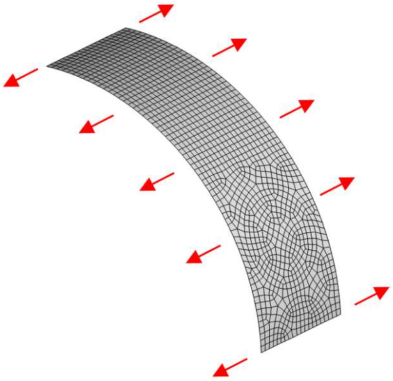


Figure 6.10: Partial Cylinder under Axial Stretch

6.6 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 6.10. 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:

$$\epsilon_{axial} = 0.010 \tag{6.4}$$

$$\epsilon_{hoop} = 0.003. \tag{6.5}$$

The analytical axial stress and hoop stress are:

$$\sigma_{axial} = \epsilon_{axial} * E = 10^4 \tag{6.6}$$

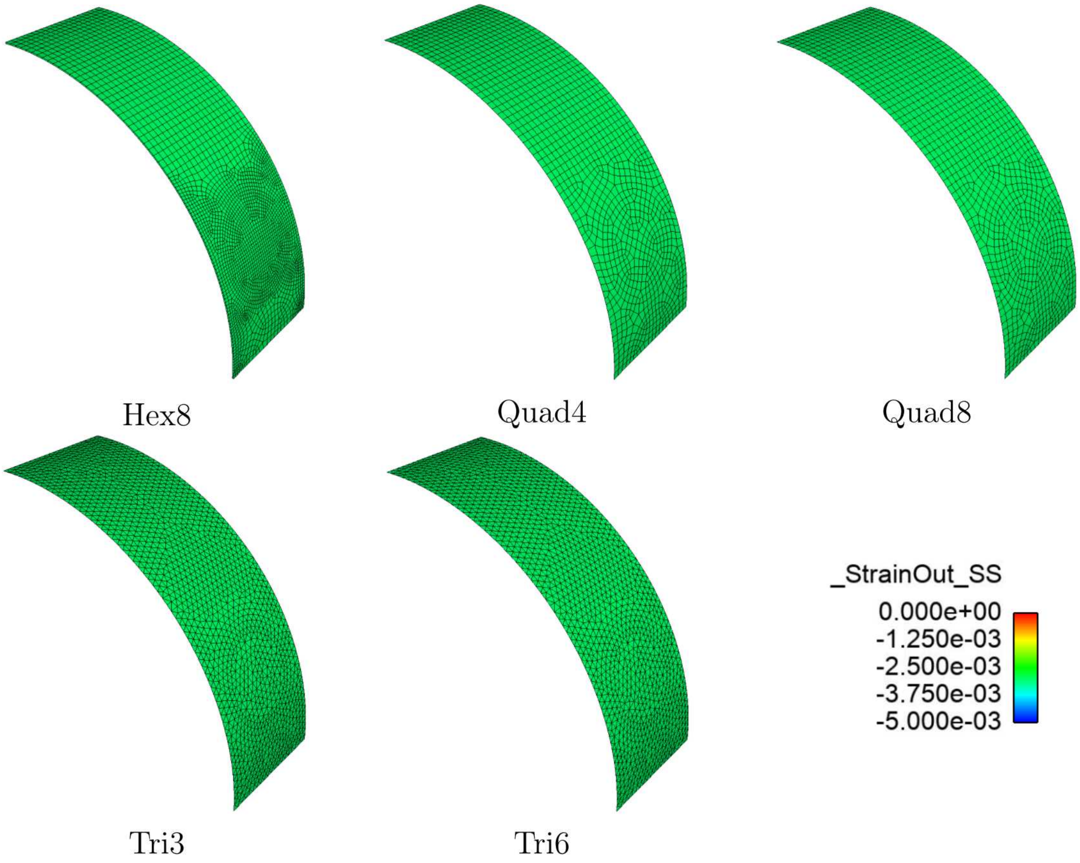
$$\sigma_{hoop} = 0.0. \tag{6.7}$$

The analytical strain energy density and total strain energy are:

$$SE_{density} = 0.5 * \sigma_{axial}\epsilon_{axial} = 50 \tag{6.8}$$

$$SE = SD_{density} * \frac{2ht\pi * r}{4} = 1.570754. \tag{6.9}$$

Post processing scripts are used to transfrom 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 6.11 shows the axial strain for each element type. Figure 6.12 shows the strain energy density for each element type. Figure 6.13 shows the axial stress for each element type. For input deck see Appendix A.12.


Hex8

Quad4

Quad8

Tri3

Tri6

Figure 6.11: Axial Strain for Partial Cylinder

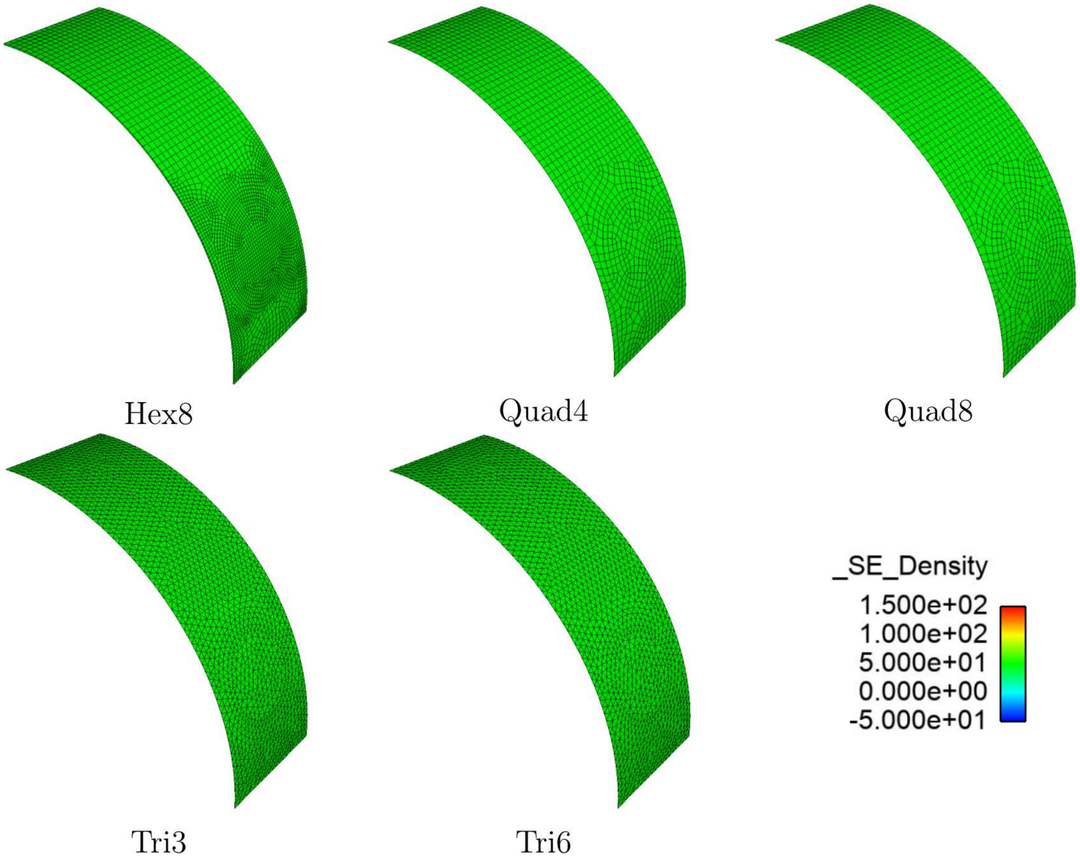


Figure 6.12: Strain Energy Density for Partial Cylinder

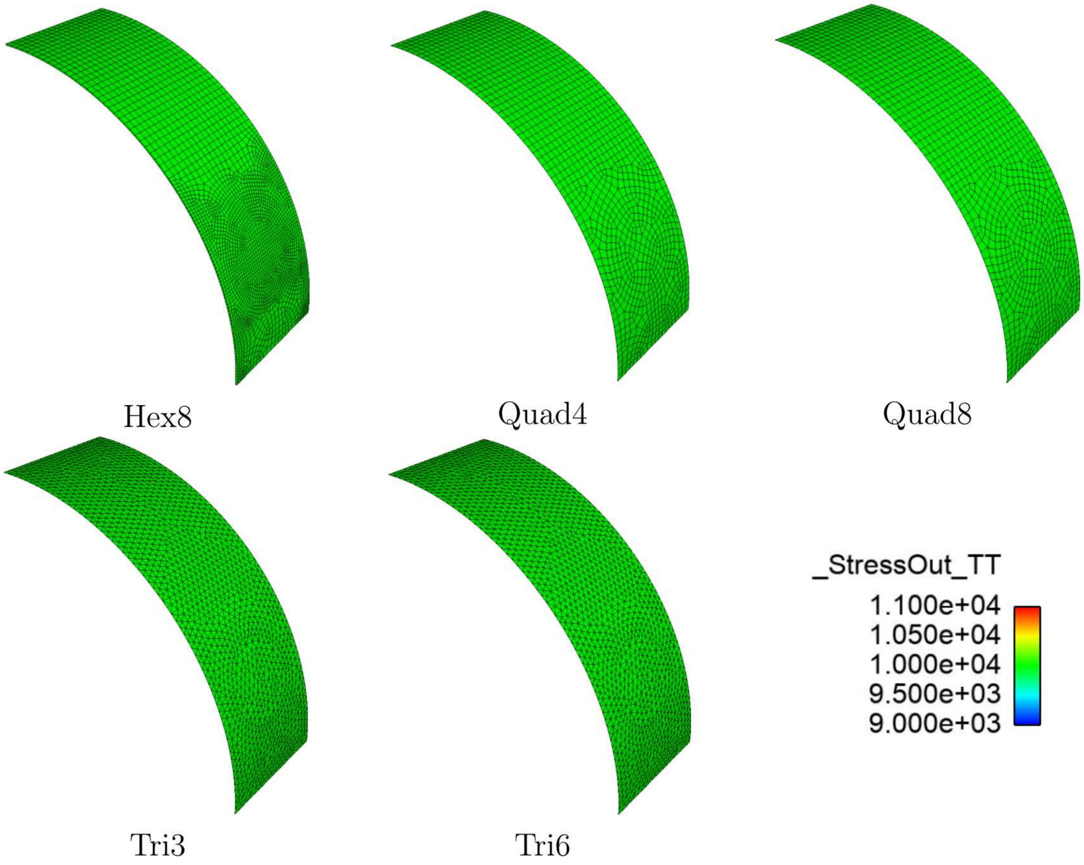

Hex8
Quad4
Quad8
Tri3
Tri6

Figure 6.13: Axial Stress for Partial Cylinder

6.7 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,

$$E_g = t\int_A \sigma_{lm}\left[\left(\frac{\partial\delta\boldsymbol{u}}{\partial x_m}\right)^T\frac{\partial\boldsymbol{u}}{\partial x_l} - \frac{1}{2}\sum_{\gamma=1}^{2}\left(\boldsymbol{e}_\gamma\frac{\partial\delta\boldsymbol{u}}{\partial x_l} + \boldsymbol{e}_l\frac{\partial\boldsymbol{u}}{\partial x_\gamma}\right)\left(\boldsymbol{e}_\gamma\frac{\partial\delta\boldsymbol{u}}{\partial x_m} + \boldsymbol{e}_m\frac{\partial\boldsymbol{u}}{\partial x_\gamma}\right)\right]dA \quad (6.10)$$

6.7.1 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

$$N_1 = (1-x)(1-y) \quad (6.11)$$
$$N_2 = x(1-y) \quad (6.12)$$
$$N_3 = (1-x)y \quad (6.13)$$
$$N_4 = xy. \quad (6.14)$$

The shape function derivatives are then

$$N_{1,x} = y-1 \quad (6.15)$$
$$N_{1,y} = x-1 \quad (6.16)$$
$$N_{2,x} = 1-y \quad (6.17)$$
$$N_{2,y} = -x \quad (6.18)$$
$$N_{3,x} = -y \quad (6.19)$$
$$N_{3,y} = 1-x \quad (6.20)$$
$$N_{4,x} = y \quad (6.21)$$
$$N_{4,y} = x \quad (6.22)$$

We have

$$\boldsymbol{u} = \sum_{i=1}^{3}(u_{1,i}N_1 + u_{2,i}N_2 + u_{3,i}N_3 + u_{4,i}N_4)\boldsymbol{e}_i, \quad (6.23)$$

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

$$\boldsymbol{u}_{,x} = \sum_{i=1}^{3}(u_{1,i}N_{1,x} + u_{2,i}N_{2,x} + u_{3,i}N_{3,x} + u_{4,i}N_{4,x})\boldsymbol{e}_i \quad (6.24)$$

$$\boldsymbol{u}_{,y} = \sum_{i=1}^{3}(u_{1,i}N_{1,y} + u_{2,i}N_{2,y} + u_{3,i}N_{3,y} + u_{4,i}N_{4,y})\boldsymbol{e}_i \quad (6.25)$$

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

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

$$-\frac{1}{2}t \int_A \sigma_{22} \left[(\boldsymbol{e}_1 \boldsymbol{u}_{,y} + \boldsymbol{e}_2 \boldsymbol{u}_{,x})(\boldsymbol{e}_1 \boldsymbol{u}_{,y} + \boldsymbol{e}_2 \boldsymbol{u}_{,x}) + (\boldsymbol{e}_2 \boldsymbol{u}_{,y} + \boldsymbol{e}_2 \boldsymbol{u}_{,y})(\boldsymbol{e}_2 \boldsymbol{u}_{,y} + \boldsymbol{e}_2 \boldsymbol{u}_{,y}) \right] dA \tag{6.26}$$

or,

$$\begin{aligned}
\frac{E_g}{t\sigma_{22}} &= \int_A \boldsymbol{u}_{,y}^T \boldsymbol{u}_{,y} dA && (6.27)\\
&-\frac{1}{2} \int_A (\boldsymbol{e}_1 \boldsymbol{u}_{,y})^2 dA && (6.28)\\
&- \int_A (\boldsymbol{e}_1 \boldsymbol{u}_{,y})(\boldsymbol{e}_2 \boldsymbol{u}_{,x}) dA && (6.29)\\
&-\frac{1}{2} \int_A (\boldsymbol{e}_2 \boldsymbol{u}_{,x})^2 dA && (6.30)\\
&-2 \int_A (\boldsymbol{e}_2 \boldsymbol{u}_{,y})^2 dA && (6.31)
\end{aligned}$$

6.7.1.1 $K_{1,1}$ 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,

$$\begin{aligned}
\boldsymbol{u}_{,x} &= N_{1,x} \boldsymbol{e}_1 && (6.32)\\
\boldsymbol{u}_{,y} &= N_{1,y} \boldsymbol{e}_1 && (6.33)
\end{aligned}$$

Then,

$$\begin{aligned}
\frac{E_g}{t\sigma_{22}} &= \int_A \frac{N_{1,y}^2}{2} dA && (6.34)\\
&= \int_A \frac{(x-1)^2}{2} dx\, dy && (6.35)\\
&= \left. \frac{(x-1)^3}{6} \right|_0^1 && (6.36)\\
&= \frac{1}{6} && (6.37)
\end{aligned}$$

6.7.1.2 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.

$$K_{11} = t\sigma_{22}/6 \tag{6.38}$$

$$Kg_{22} = -t\sigma_{22}/2 \tag{6.39}$$

$$Kg_{33} = t\sigma_{22}/3 \tag{6.40}$$

$$Kg_{12} = t\sigma_{22}/8 \tag{6.41}$$

$$\tag{6.42}$$

The maple script is available.

6.7.1.3 Rotations

The test in this directory runs only on a unit square in the xy plane. However, a related verification test 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 A.13.

6.8 Membrane Quad

A verification test was created for membrane elements in Sierra-SD. The geometry of this test is shown in Figure 6.14.

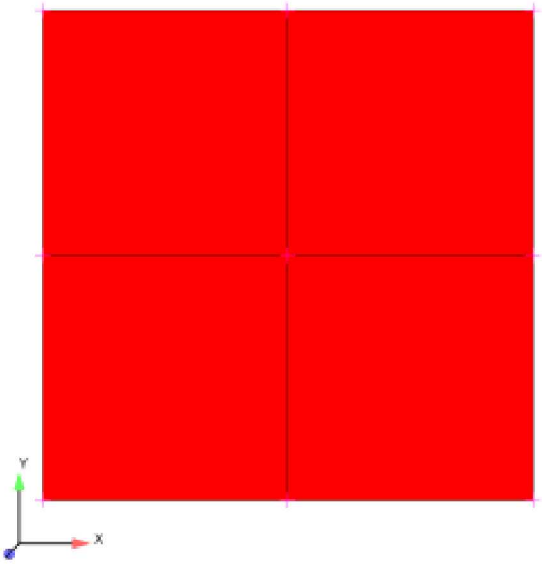


Figure 6.14: membraneGeometry

There is a total of 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. This is an eigen solution case with a total of fourteen modes. For verification the test in Sierra-SD was compared to the Abaqus finite element code. The Eigenvalue results are shown in Table 6.8. All modes are compared. There are nine rigid body modes in the model.

For input deck see Appendix A.14.

Table 6.8: 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

6.9 QuadM membrane Patch

A patch test that was created for a SierraSD membrane element. The geometry of this test is shown in Figure 6.15. There are a total of five boundary conditions constraining the model. First, all nodes 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 6.9.

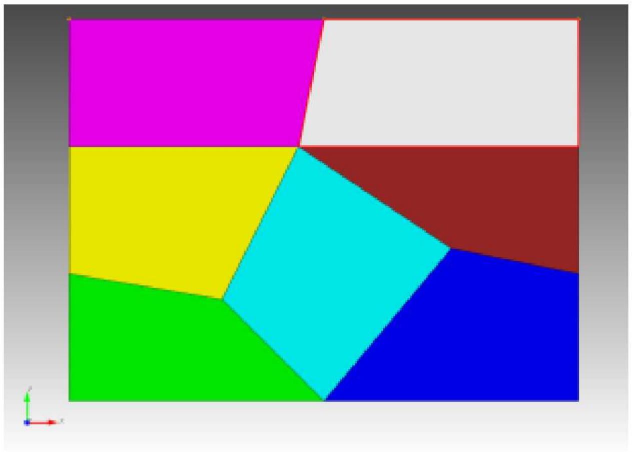


Figure 6.15: Patch Test Geometry

6.9.1 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.

6.9.2 Rotated Patch Test

Further verification was performed using the same patch test by rotating the test out of the XY plane, shown in Figure 6.16.

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

Table 6.9: Strain for Membrane Elements

Node Number	Strain
1	Fixed = 0
2	0.0250
3	0.0250
4	Fixed = 0
5	0.0250
6	Fixed = 0
7	0.0250
8	Fixed = 0
9	0.0250
10	0.0250
11	0.0250
12	0.0250
13	0.0250

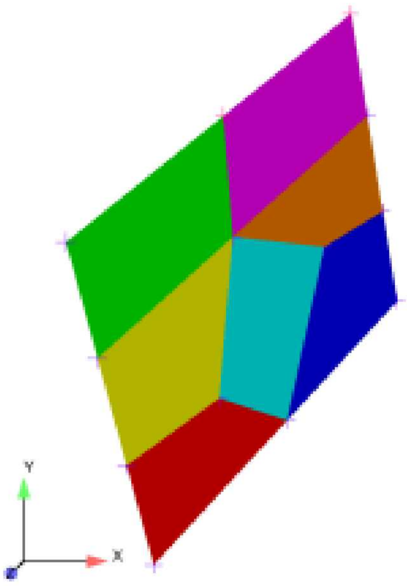


Figure 6.16: Test Geometry

modes. The first 10 modes for the rotated test are compared to the in plane patch test. The Eigenvalue results are shown in Table 6.10. As expected, the modes are the same in both cases and are invariant with respect to the rotation of the model.

Table 6.10: 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

6.9.3 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 6.11. The strain is constant for every node through out the model, therefore, verifying the patch test is working.

6.9.4 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 6.12. There are 12 rigid body modes in the model, and the remaining modes show an acceptable comparison of the two codes.

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

Table 6.11: Strain for Hex Elements

Node Number	Strain
1	Fixed = 0
2	Fixed = 0
3	0.0250
4	0.0250
5	Fixed = 0
6	Fixed = 0
7	0.0250
8	0.0250
9	0.0250
10	Fixed = 0
11	0.0250
12	Fixed = 0
13	Fixed = 0
14	Fixed = 0
Nodes 15-26	0.0250

Table 6.12: 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 fiber direction as expected. For input deck see Appendix A.15.

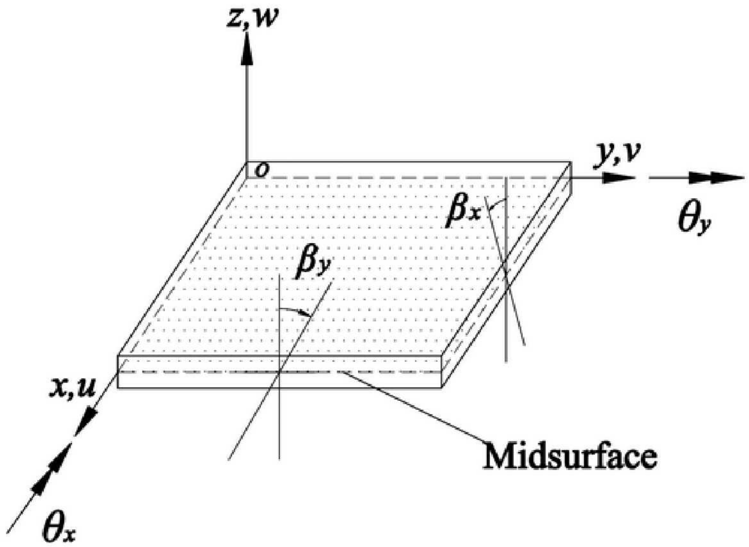


Figure 6.17: Transverse shear strains β_x and β_y allow cross sections to not remain on a plate perpendicular to fiber direction. This relaxation of the Kirchhoff hypothesis enables accurate study of thick plates and shells.

6.10 QuadS_GY Shear Membrane Shell

Verification of the QuadS_GY Element. The existing Salinas membrane element used for eigenmode/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.? 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 according to the 2-by-2 Gauss rule, whereas shear deformation is underintegrated at one central point. This mixed strategy avoids the "locking" effects caused by 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.

6.10.1 Eigenvalue analysis: Verification on a flat shell

In this section, we verify the new element using two procedures: 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 $E = 30$ MPa, the Poisson ration is 0.3, and density is 0.288 kg/m^3.

6.10.1.1 Isotropic

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

$$f_{ij} = \frac{\lambda_{ij}^2}{2\pi a^2}\left[\frac{Eh^3}{12\gamma(1-\nu^2)}\right]^{\frac{1}{2}}, \tag{6.43}$$

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.[?]).

6.10.1.1.1 Fixed-Fixed-Fixed-Fixed (FFFF) The bending eigenfrequencies of the plate for two different thickness values are reported in Tables 6.13 and 6.14. The shear-deformable shell element (QuadS_GY) results naturally diverge from bending theory for increasingly thicker sections.

Table 6.13: Eigenfrequencies for FFFF flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

	Analytiical	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)

6.10.1.1.2 Free-Free-Free-Free(FrFrFrFr) Tables 6.15 and 6.16 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.

6.10.1.1.3 Simply supported-Free-Free-Free (SFrFrFr) The natural frequencies associated with the lowest-frequency deformation modes are shown in Tables 6.17 and 6.18.

Table 6.14: Eigenfrequencies 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)

6.10.1.2 Orthotropic

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 6.43, also provided in Ref.[?] Results are summarized in Table 6.19. A graphical comparison of the (32) mode for two SD elements is shown in Fig. 6.18.

For input deck see Appendix A.16.

Table 6.15: Eigenfrequencies 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)

Table 6.16: Eigenfrequencies 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 6.17: Eigenfrequencies for SFrFrFr flat shell of thickness 0.001 m. Frequencies are in Hertz and discrepancies from theory are given in percentage between parenthesis.

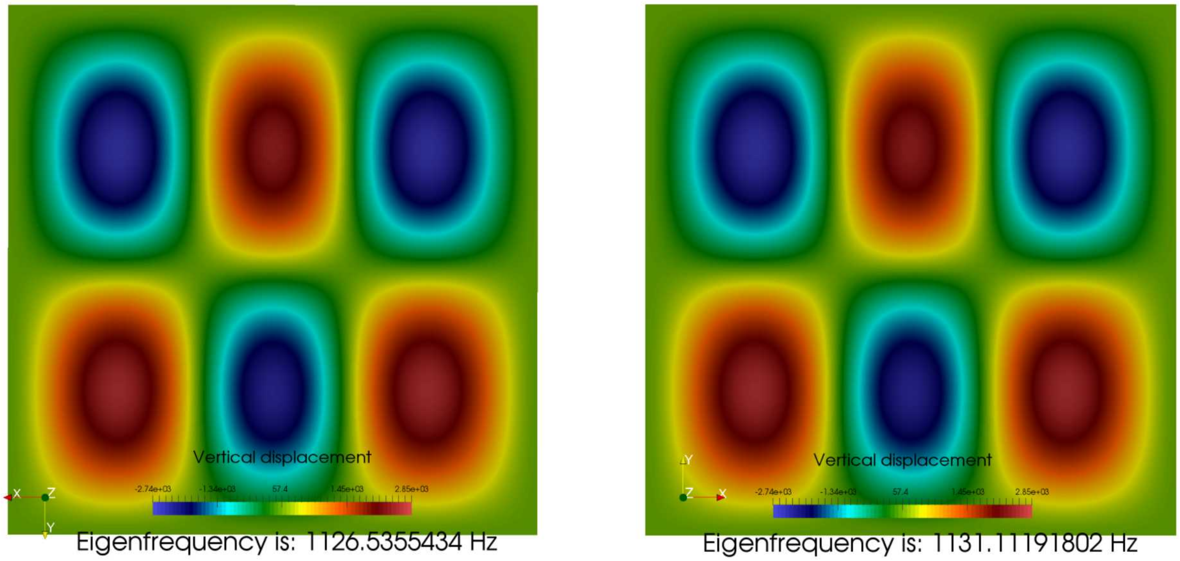
	Analytical	QuadT	QuadS_GY
1st mode	64.212	64.152	64.177
	(Ref.)	(0.09)	(0.05)
2nd mode	145.075	143.874	143.905
	(Ref.)	(0.83)	(0.81)
3rd mode	246.203	244.989	244.650
	(Ref.)	(0.49)	(0.63)
4th mode	252.384	250.912	249.830
	(Ref.)	(0.58)	(1.01)
5th mode	470.480	467.576	467.594
	(Ref.)	(0.62)	(0.61)
6th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

Table 6.18: Eigenfrequencies 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)

Table 6.19: Eigenfrequencies 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)



(a) QuadT (memb.+ bend.)

(b) QuadS_GY (memb.+ bend. + shear)

Figure 6.18: Comparison of (32) modes resulting from orthotropic material model (see Table 6.19).

6.11 QuadS_GY Shear Membrane Shell - Geometric Stiffness and Preload

Verification of the QuadS_GY Geometric Stiffness matrix and SierraSM Preload.

6.11.1 Verification of geometric stress stiffness matrix

The effect of geometric stiffness was evaluated by preloading a cantilever beam made up of shear-deformable shell elements. This example was also used for the verification of other Sierra-SD shell elements. This model consists of a beam clamped on one end and with applied axial pressure on the other end. Beam dimensions are 0.127 m (length), 0.0044504 m (width), and 0.0044504 m (thickness). The modulus of elasticity is 187 GPa, the Poisson ratio is 0.3, and density is 8015.19 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 6.20: 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 6.20:

- **SD shell**. In Sierra-SD, the pressure load is applied to the shelled beam and, with the resulting displacements, the sytem 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 eigenanalysis 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 6.20.

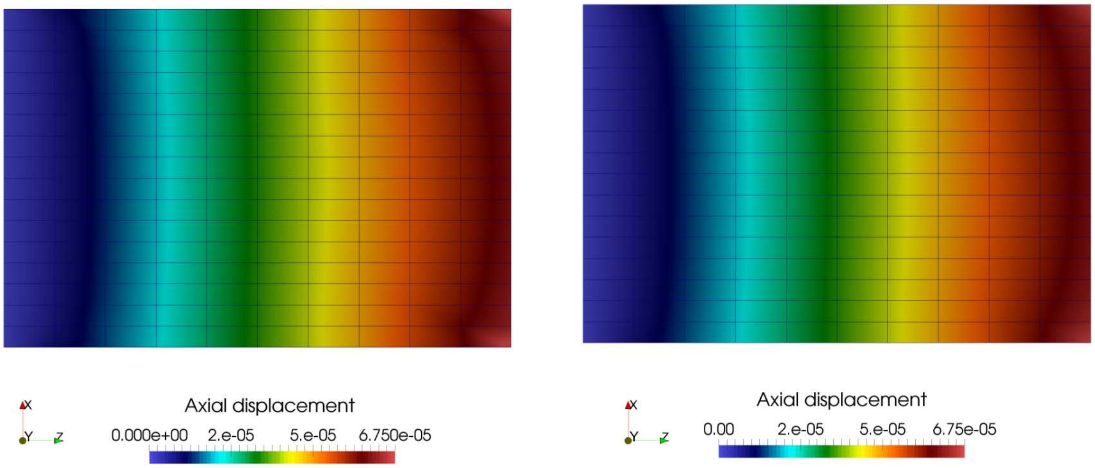
6.11.2 Verification Sierra-SM–Sierra-SD for small deformation

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

6.11.2.1 Isotropic shell

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. 6.19 and 6.20. 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.

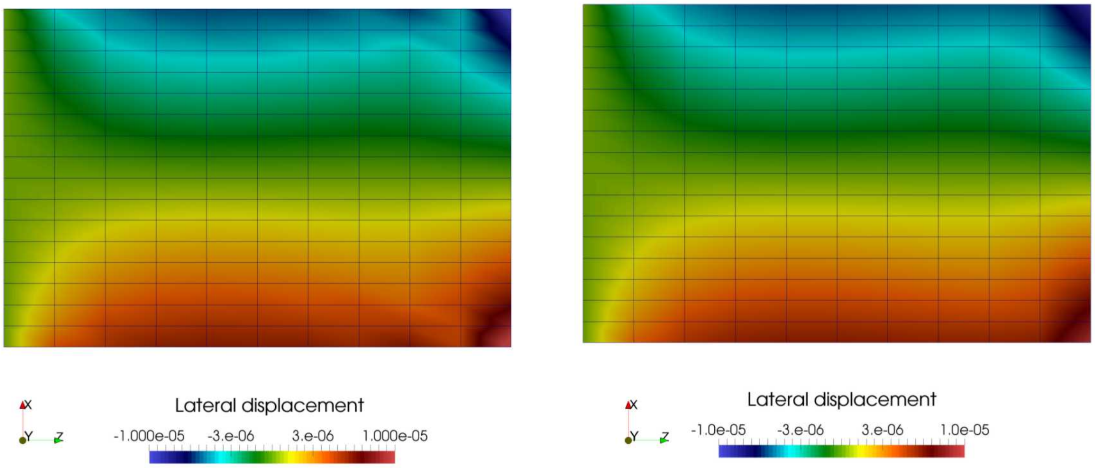
For input deck see Appendix A.17.



(a) Axial displacement in Sierra-SM

(b) Axial displacement in Sierra-SD

Figure 6.19: Comparison of axial displacement.



(a) Lateral displacement in Sierra-SM

(b) Lateral displacement in Sierra-SD

Figure 6.20: Comparison of lateral displacement.

6.12 Hex Membrane Sandwich

6.12.1 Isotropic Material

A simple plate model was constructed and analyzed using hex and membrane elements, shown in Figure 6.21.

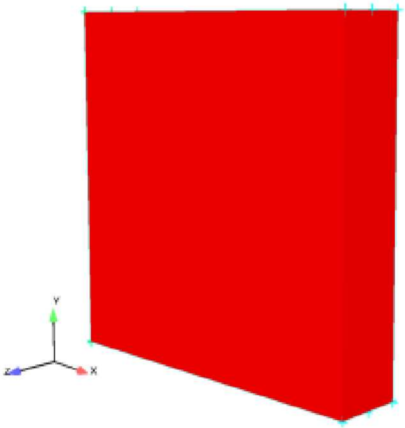


Figure 6.21: Test Geometry

The first test using this plate model was unpreloaded, and 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 6.21.

Table 6.21: 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 unpreloaded 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 6.22.

Table 6.22: 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

6.12.2 Orthotropic Material

The same plate model was tested using orthotropic material properties. The material elasticity tensor only provides stiffness in the x direction, with zero stiffness in the remaining directions.The first test was an unpreloaded model. The Eigenvalue results are shown in Table 6.23.

Table 6.23: 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, except that a 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 6.24.

Table 6.24: 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 A.18.

6.13 Higher Order Hex Acoustic Element Convergence

This section demonstrates a convergence study for the phex 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 6.22. 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

$$f_n = \frac{nc}{2L} = 16.6, 33.2, ... \tag{6.44}$$

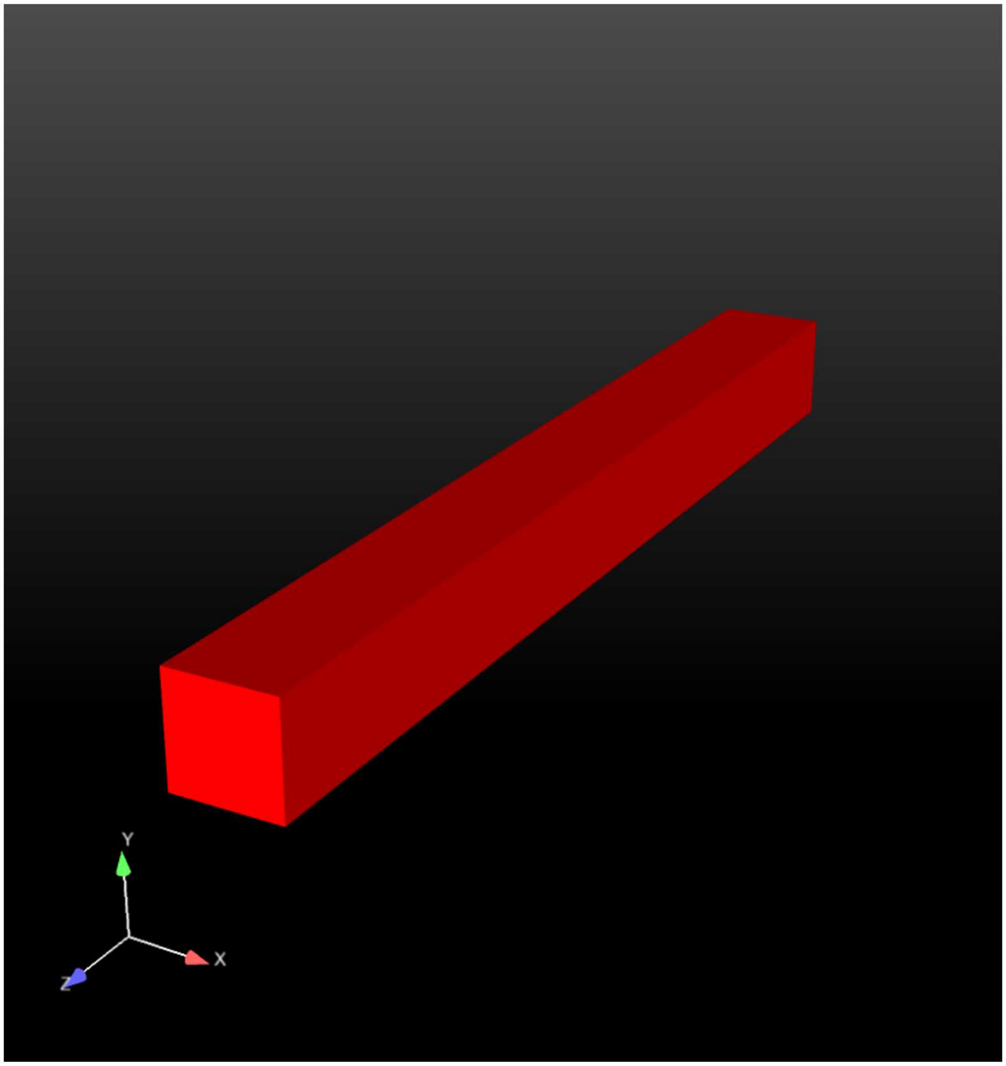


Figure 6.22: Waveguide Model for Convergence Study of P-hex elements.

Figure 6.23 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 6.23 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 $\frac{1}{h}$).

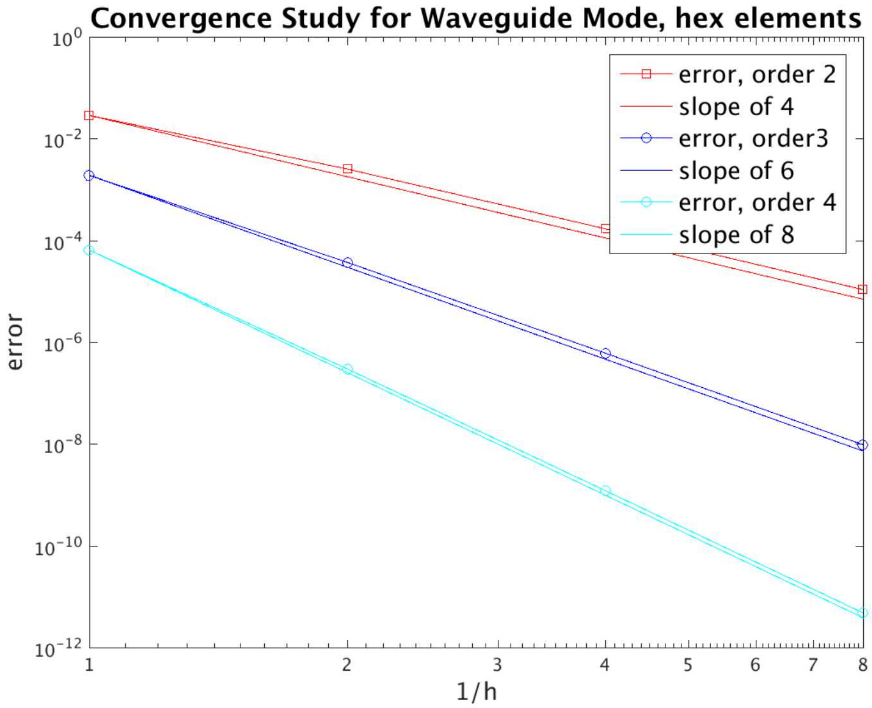


Figure 6.23: Convergence Study of P-hex elements.

For input deck see Appendix A.33.

6.14 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 6.24. 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

$$f_n = \frac{nc}{2L} = 16.6, 33.2, ... \tag{6.45}$$

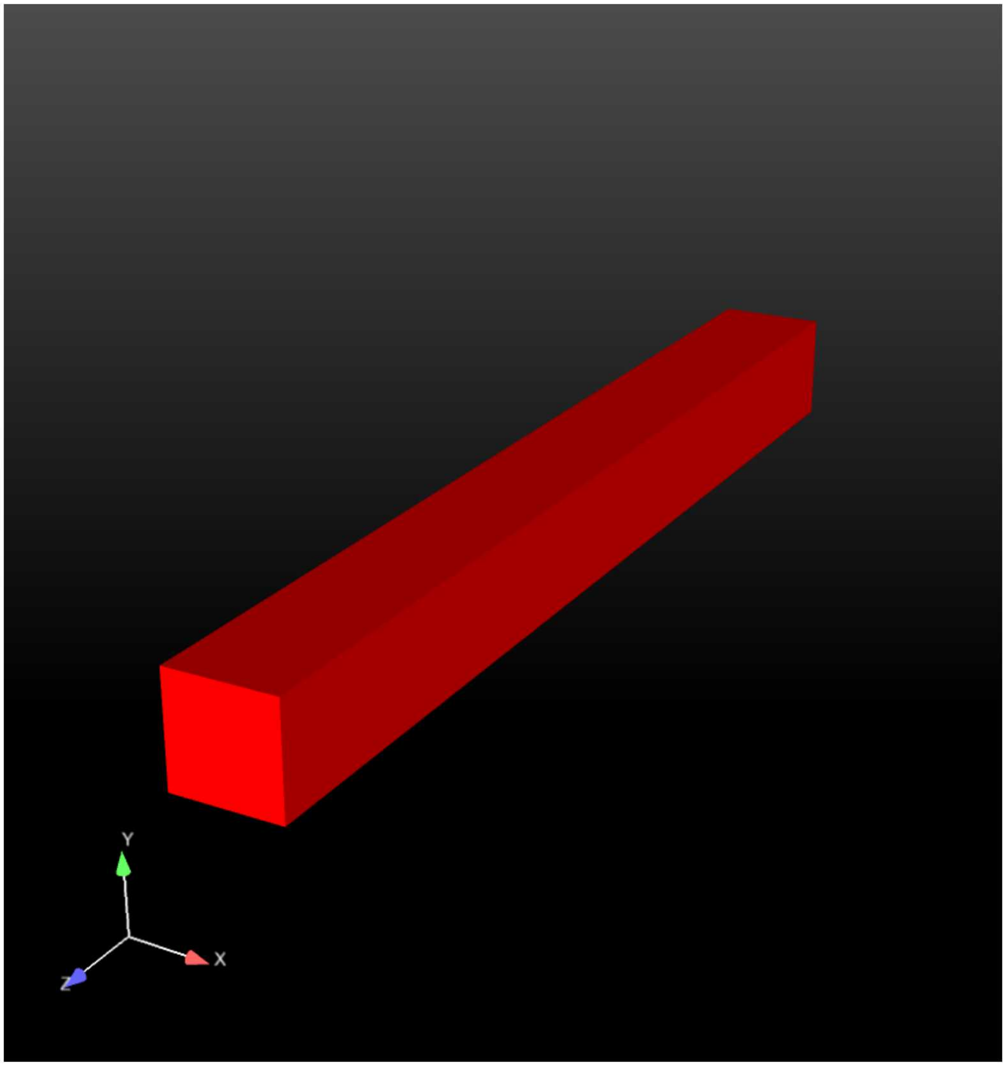


Figure 6.24: Waveguide Model for Convergence Study of P-tet elements.

Figure 6.25 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 6.25 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 $\frac{1}{h}$).

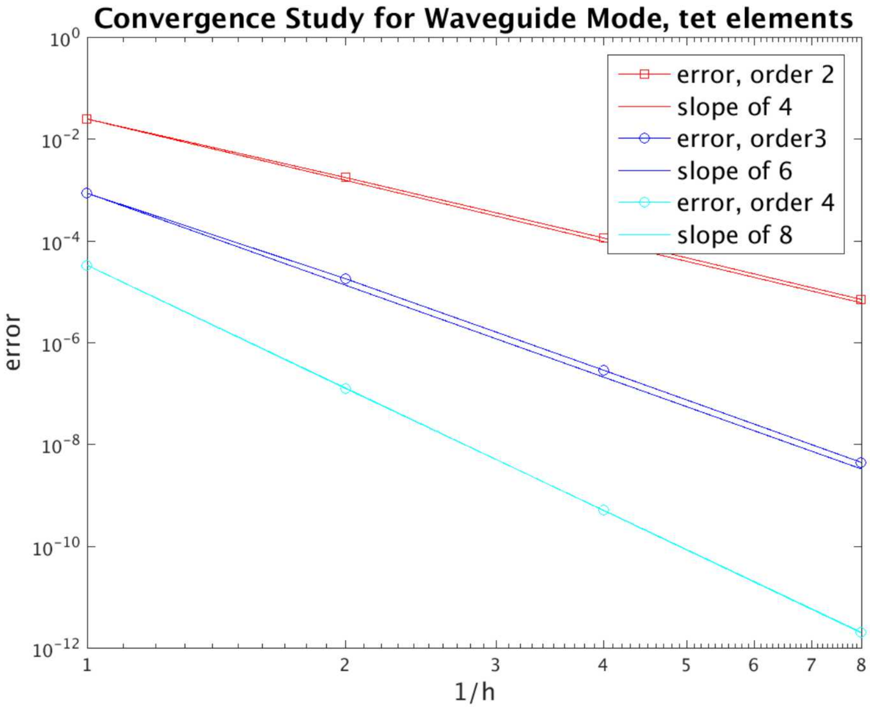


Figure 6.25: Convergence Study of P-tet elements.

For input deck see Appendix A.34.

6.15 P-elements on 1-D waveguide up to order 6

This section demonstrates a convergence study for the phex element, up to order 6. The geometry of the model is shown in Figure 6.26. It consists of an acoustic waveguide of length $L = 10.0(m)$, and cross sectional dimensions of $1.0(m)$. One end of the waveguide is assigned as a sideset with an absorbing boundary. The other end is the sideset assigned to where the load is applied, in this case an acoustic velocity. The node where the solution has been found is a corner node on the loaded sideset.

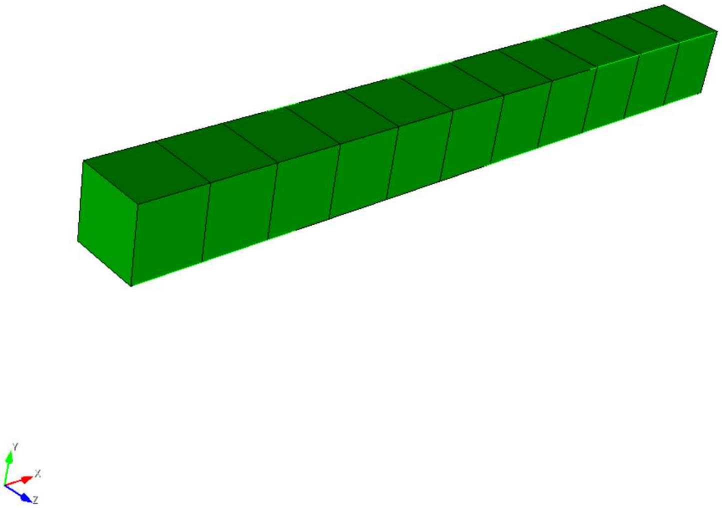


Figure 6.26: Waveguide Model for Convergence Study of P-hex Elements.

Figure 6.27 shows the convergence of the real part of acoustic pressure for the actual solution, as well as the value of expected convergence. It can be seen that the actual value quickly begins to converge to the theoretical value around order 3.

Figure 6.28 shows the convergence of the imaginary part of acoustic pressure. Similar to real part of acoustic pressure, the computed value of imaginary pressure quickly approached the expected value near order 3.

Figure 6.29 shows the convergence of acoustic impedance for the actual solution. As expected, the Impedance approaches the theoretical value around order 3. In Figure 6.30 we show the relative error for Apressure on a semilog plot.

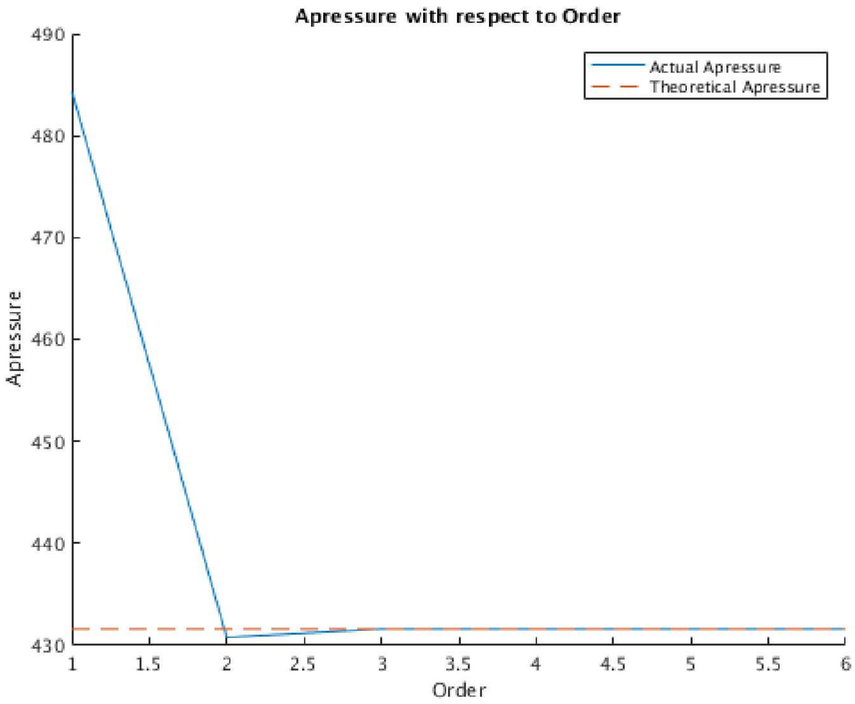


Figure 6.27: Convergence Study of P-hex elements-Apressure.

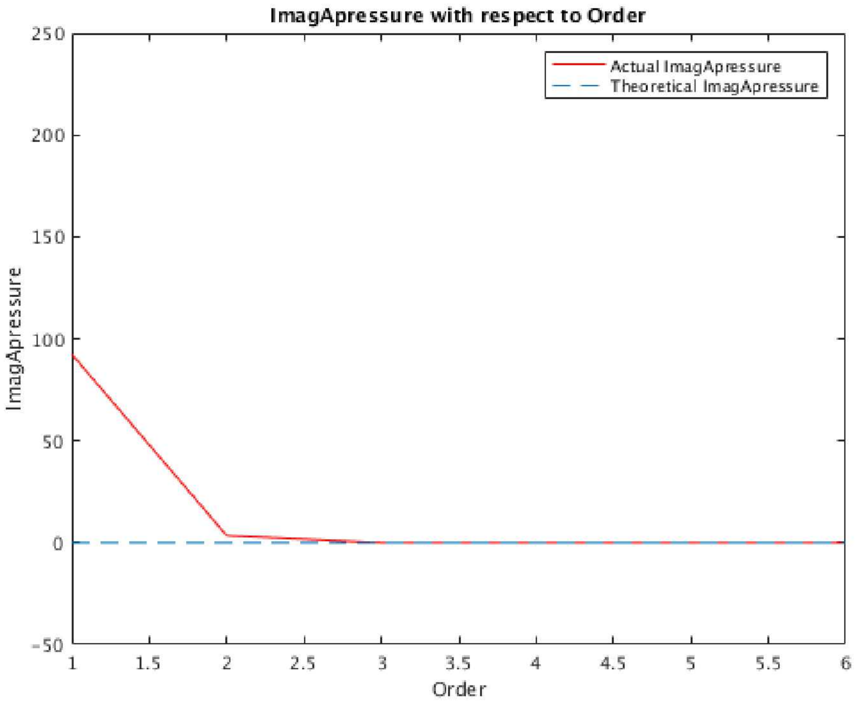


Figure 6.28: Convergence Study of P-hex elements-ImagApressure.

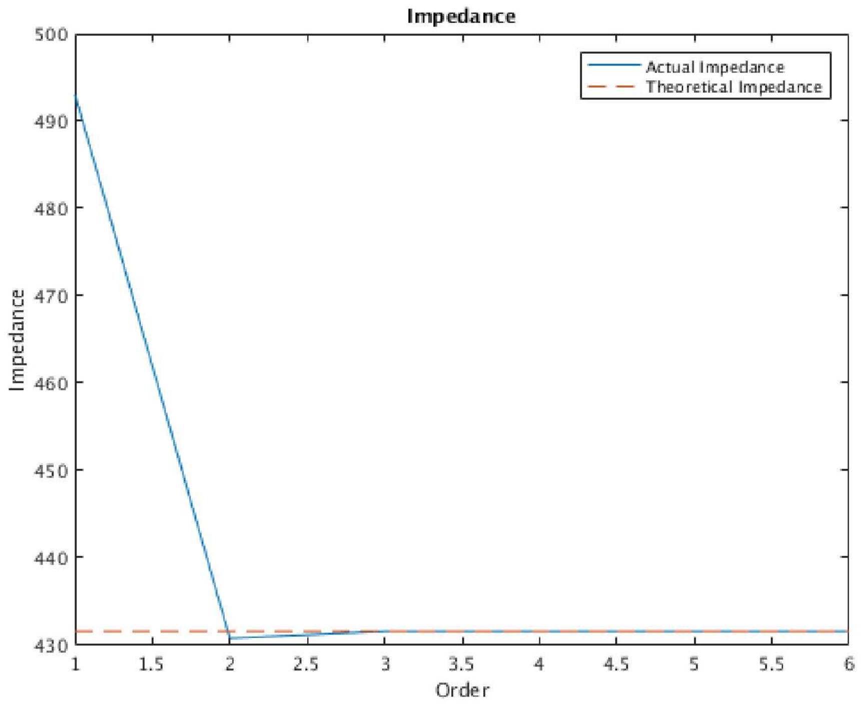


Figure 6.29: Convergence Study of P-hex elements-Impedance.

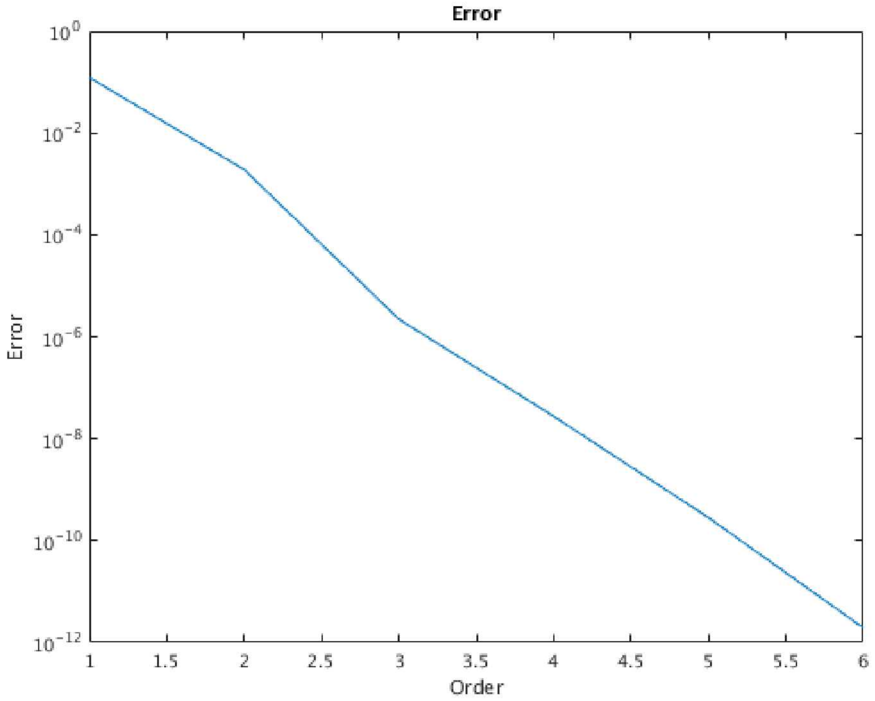


Figure 6.30: Convergence Study of P-hex elements-Relative Error.

For input deck see Appendix A.35.

6.16 P-elements on Acoustic Sphere for Multiple Refined Hex-Meshes

This section demonstrates a convergence study for the phex element, up to order 3 using several refinements of one mesh.

The geometry of the model is shown in Figure 6.31. It consists of an acoustic sphere of radius $r = 5.0(m)$ hollowed out with a smaller sphere of $r = 1.0(m)$. The outer surface is assigned as a sideset with an absorbing boundary. The inner surface is the sideset assigned to where the load is applied, in this case an acoustic velocity. The node where the solution has been found is a node on the inner surface of the sphere.

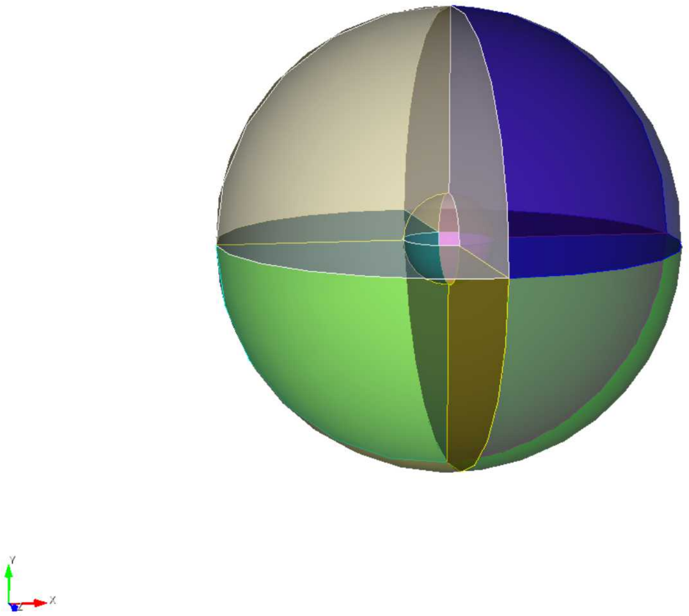


Figure 6.31: Acoustic Spherical Model for Convergence Study of P-hex Elements.

Figure 6.32 shows the convergence of the real part of acoustic pressure for the actual solution for each refinement of the mesh, as well as the value of expected convergence. From this figure, it can be seen that even up to order 3, convergence is not seen on the first refinement of the mesh created. This is because the course mesh on the sphere creates a sharp geometry and less of a smooth surface. The fourth refinement of the mesh approaches the expected value much more accurately. Higher order elements on the more refined meshes would most likely show closer convergence.

Figure 6.33 shows the convergence of the imaginary part of acoustic pressure. Similar to the real part of acoustic pressure, the computed value of imaginary pressure approaches the expected value more accurately as order increases for a highly refined mesh. This plot also shows how the geometry of the model can have a major effect on the accuracy of convergence.

Figure 6.34 shows the convergence of acoustic impedance for the actual solution. As expected, the Impedance approaches the theoretical value as order and refinement increase.

In Figures 6.35 and 6.36 we show the relative error for Apressure and ImagApressure for each refinement on a semilog plot.

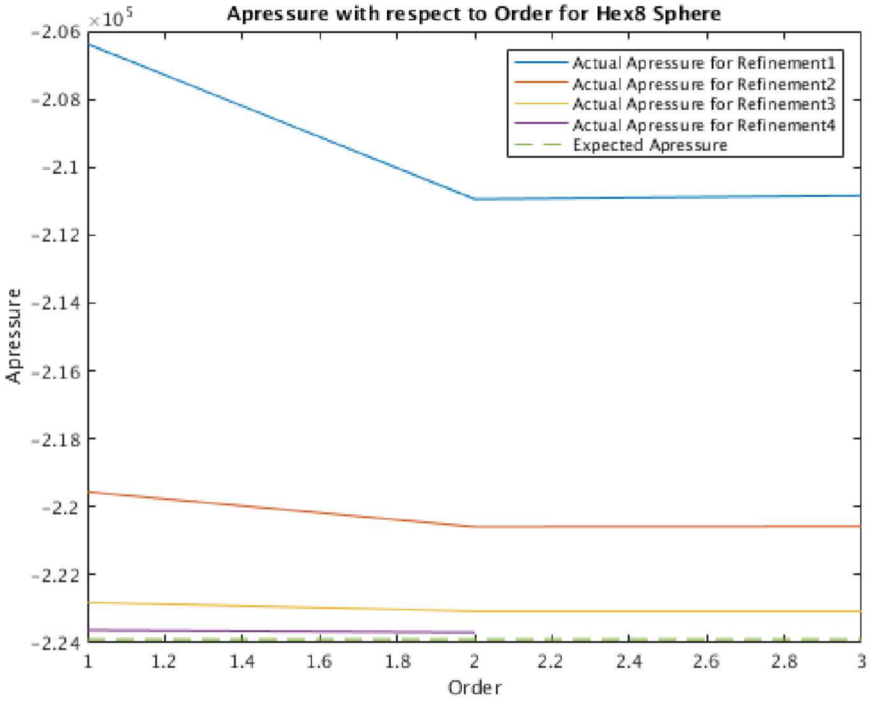


Figure 6.32: Convergence Study of P-hex elements-Apressure.

For input deck see Appendix A.36.

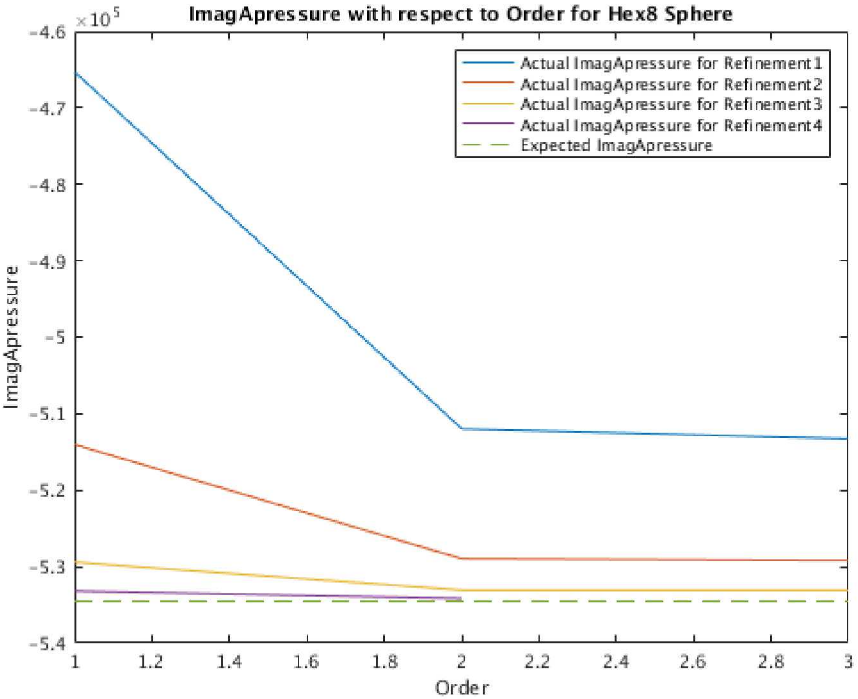


Figure 6.33: Convergence Study of P-hex elements-ImagApressure.

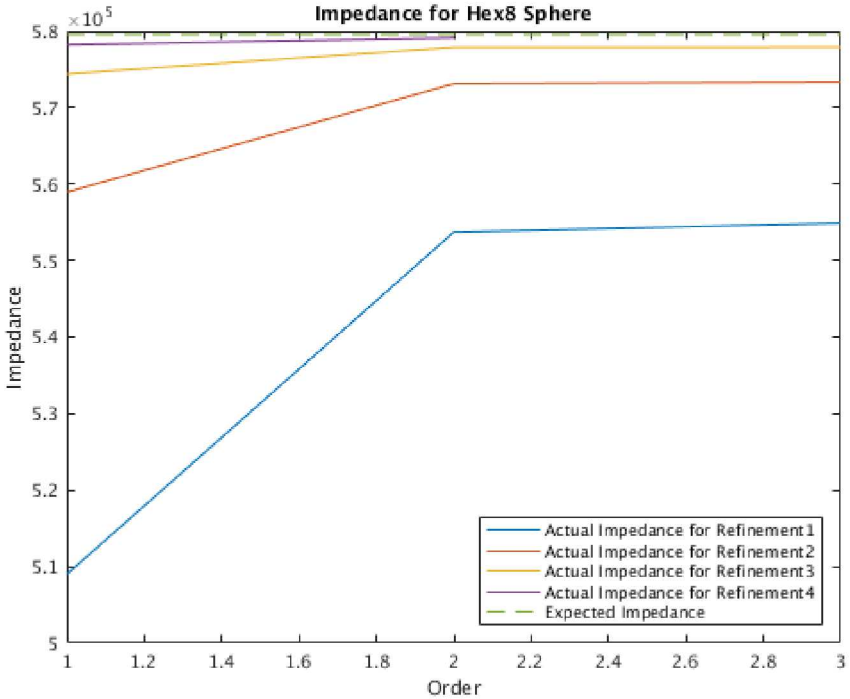


Figure 6.34: Convergence Study of P-hex elements-Impedance.

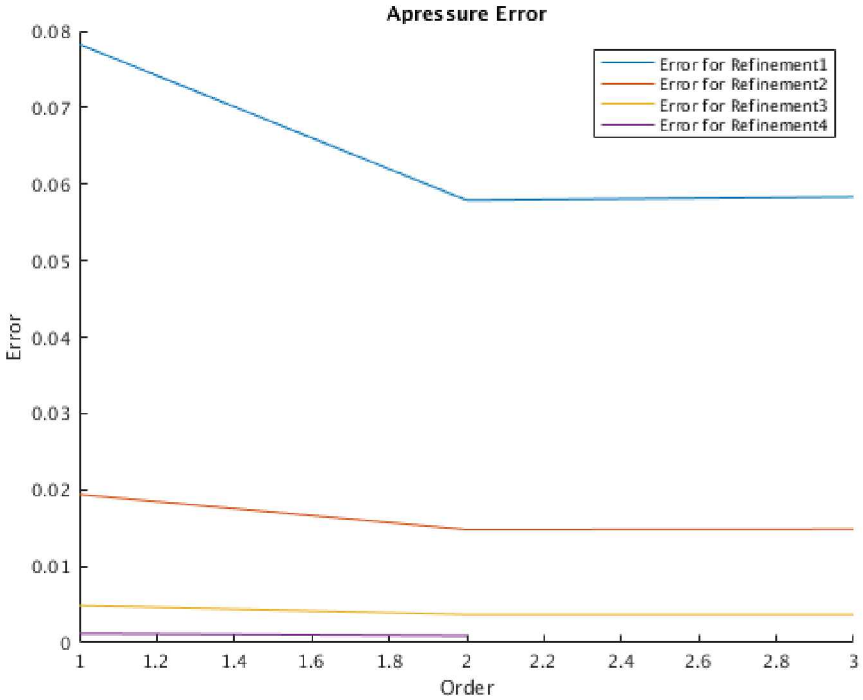


Figure 6.35: Convergence Study of P-hex elements-Relative Apressure Error.

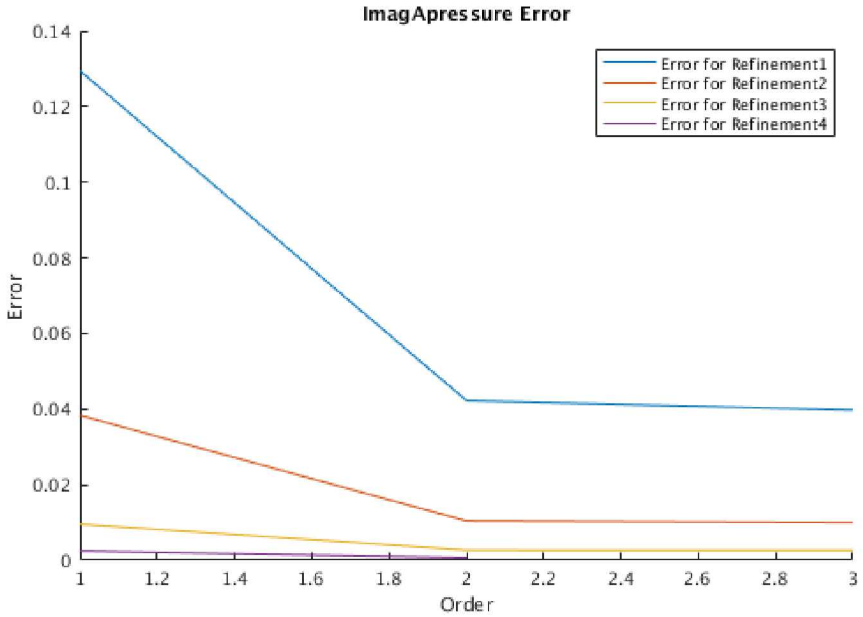


Figure 6.36: Convergence Study of P-hex elements-Relative ImagApressure Error.

6.17 P-elements on Acoustic Sphere for Multiple Refined Tet4-Meshes

This section demonstrates a convergence study for the ptet element, up to order 3 using several refinements of a tetra4 mesh.

The geometry of the model is shown in Figure 6.37. It consists of an acoustic sphere of radius $r = 5.0(m)$ hollowed out with a smaller sphere of $r = 1.0(m)$. The outer surface is assigned as a sideset with an absorbing boundary. The inner surface is the sideset assigned to where the load is applied, in this case an acoustic velocity. The node where the solution has been found is a node on the inner surface of the sphere.

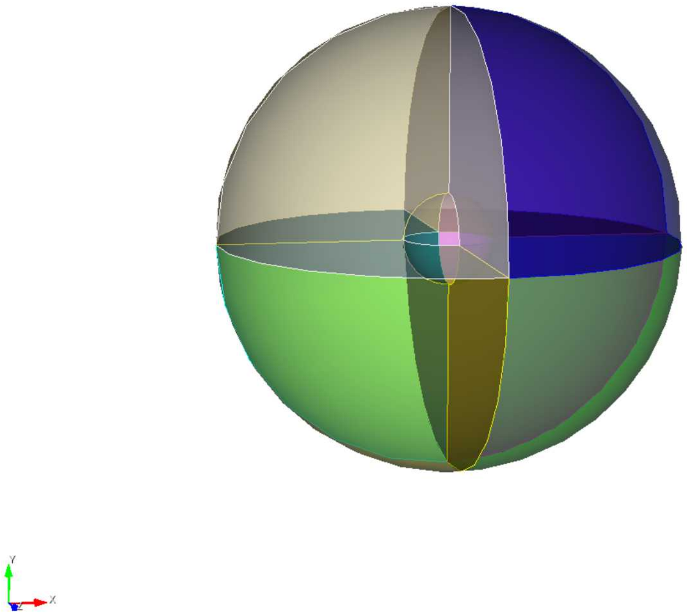


Figure 6.37: Acoustic Spherical Model for Convergence Study of P-tet Elements.

Figure 6.38 shows the convergence of the real part of acoustic pressure for the actual solution for each refinement of the mesh, as well as the value of expected convergence. From this figure, it can be seen that even up to order 3, convergence is not seen on the first refinement of the mesh created. This is because the course mesh on the sphere creates a sharp geometry and less of a smooth surface. It is important to note that the fourth refinement data is only for order 1. The purple line is extended through order 2 to give a visual of how close the fourth refined mesh is to the expected solution. Higher order elements on highly refined meshes would most likely show a more accurate convergence.

Figure 6.39 shows the convergence of the imaginary part of acoustic pressure. Similar to the real part of acoustic pressure, the computed value of imaginary pressure approaches the expected value more accurately as order increases for a highly refined mesh. However, like Apressure, the fourth refined mesh only has data for order 1 and the line has been extended for this plot in order to give a better visual of the results of using a highly refined mesh. This plot also shows how the geometry of the model can have a major effect on the accuracy of convergence.

Figure 6.40 shows the convergence of acoustic impedance for the actual solution. As expected, the Impedance approaches the theoretical value as order and refinement increase. Once again, the fourth refinement data point using order 1 has been extended.

In Figures 6.41 and 6.42 we show the relative error for Apressure and ImagApressure for each refinement on a semilog plot. Error for refinement 4 is effected by the extended point and is not completely accurate.

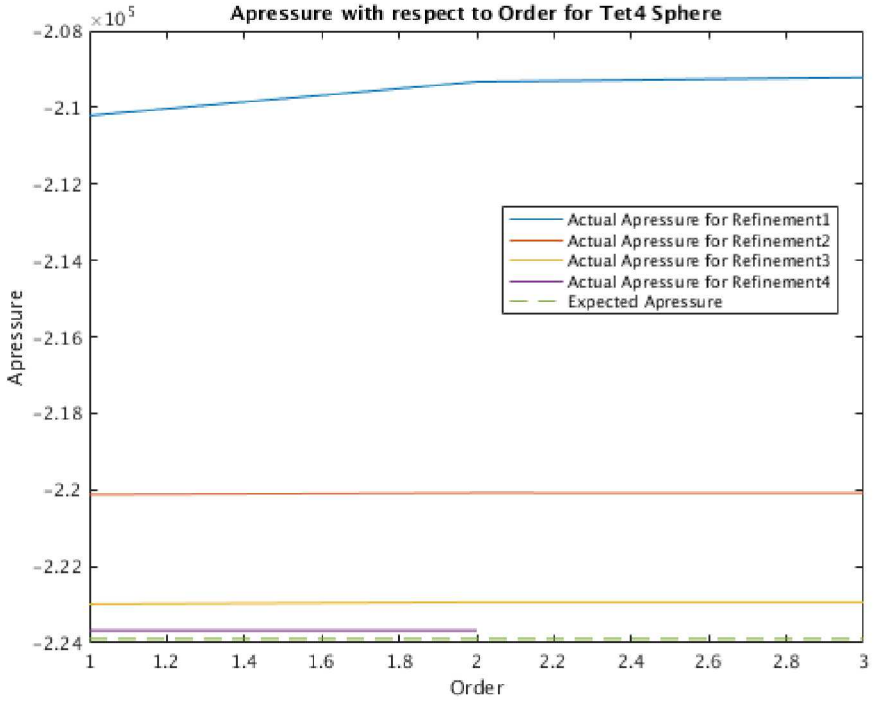


Figure 6.38: Convergence Study of P-tet elements-Apressure.

For input deck see Appendix A.37.

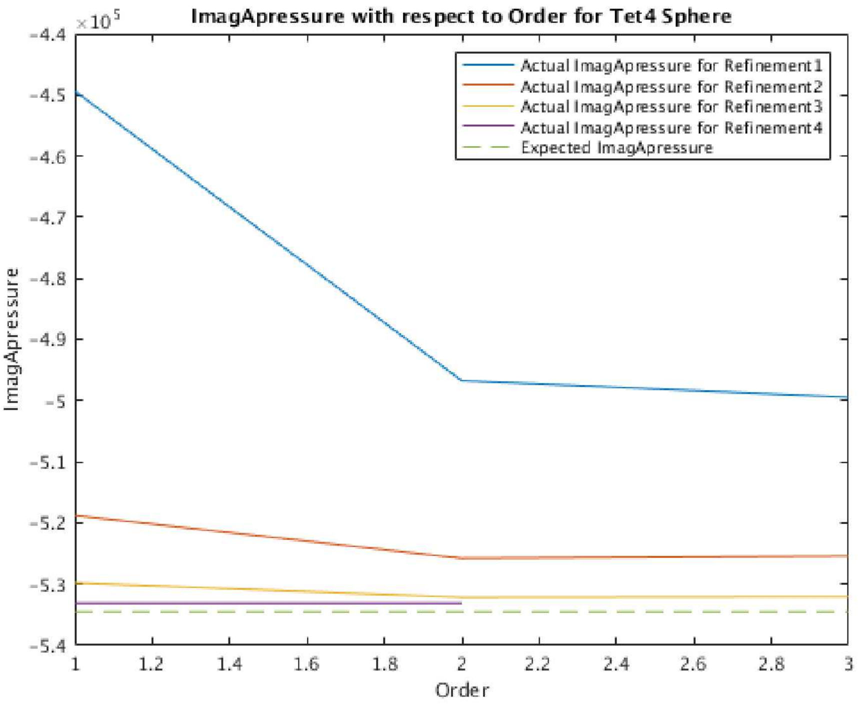


Figure 6.39: Convergence Study of P-tet elements-ImagApressure.

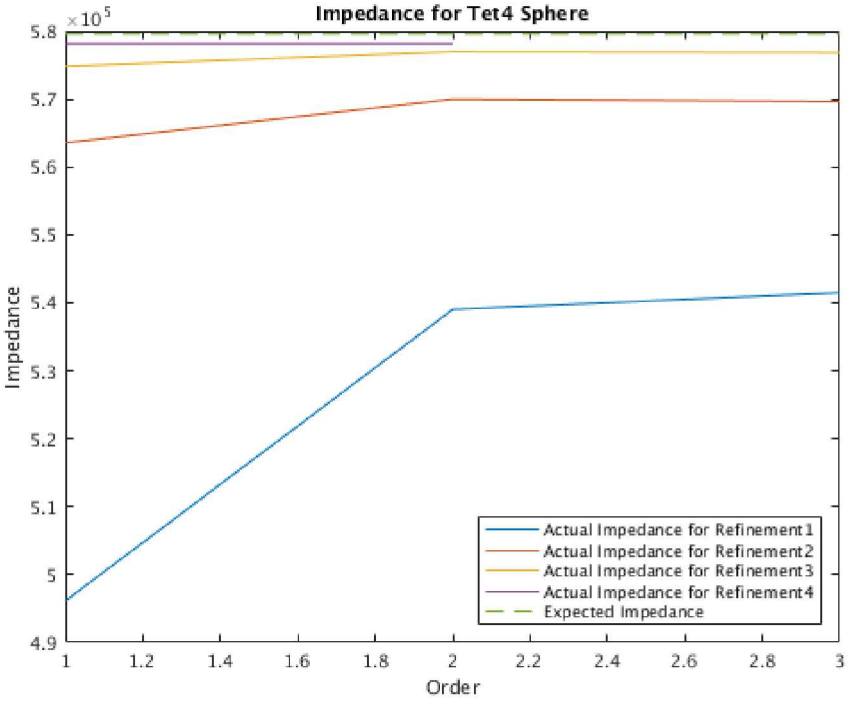


Figure 6.40: Convergence Study of P-tet elements-Impedance.

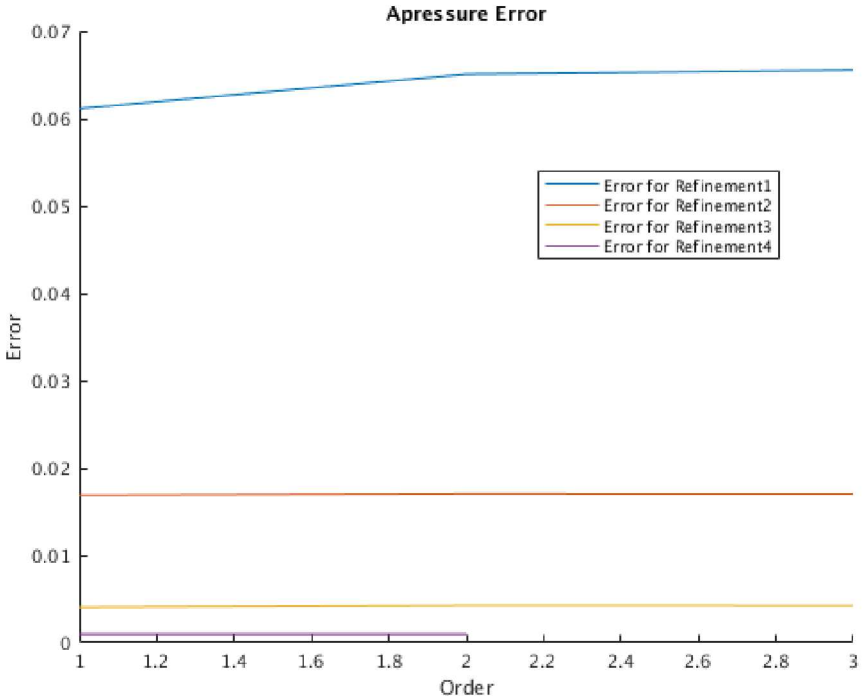


Figure 6.41: Convergence Study of P-tet elements-Relative Apressure Error.

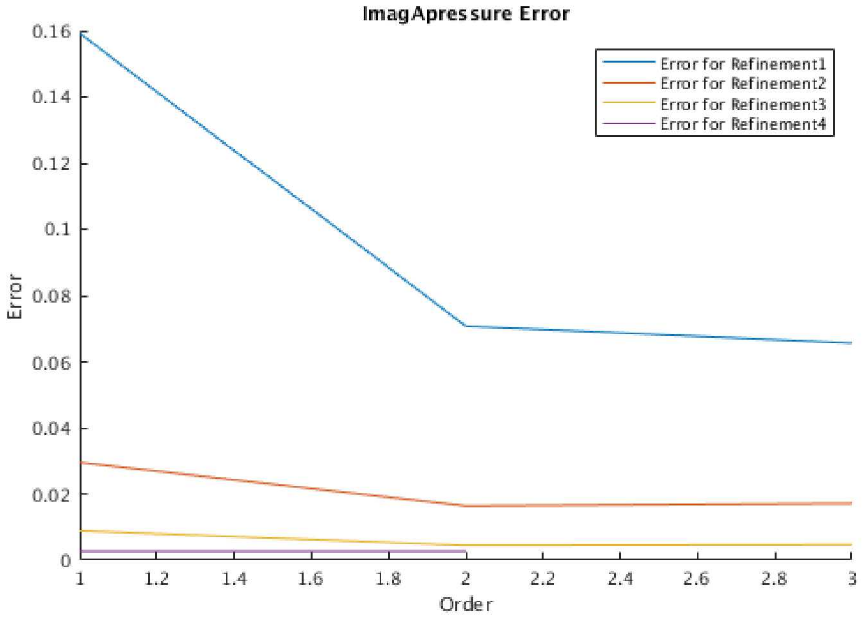


Figure 6.42: Convergence Study of P-tet elements-Relative ImagApressure Error.

6.18 Tied-Joint with Joint2G and Spring. Slip and Rigid

6.18.1 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 of 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.

6.18.2 Lap Joint Comparison

6.18.2.1 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 6.43. 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.

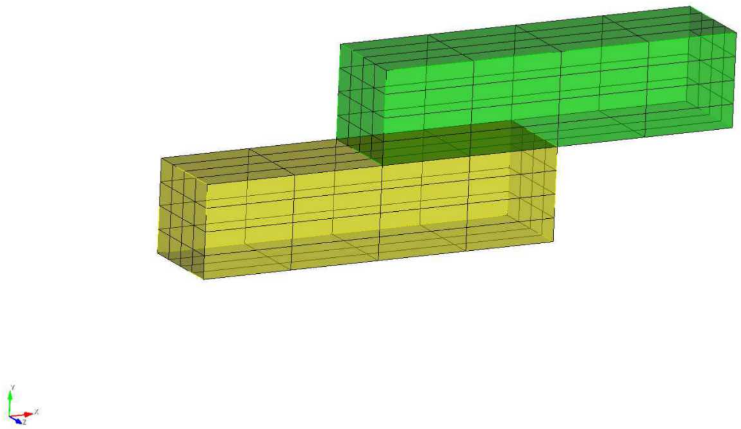


Figure 6.43: Tied-Joint Model Geometry

6.18.2.2 Building the Tied-Joint model

6.18.2.3 Non-slip

The exodus file of the original model as described in 6.18.2.1 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 6.44. 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.

```
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 6.44: Tied-Joint Non-Slip Input

6.18.2.4 Slip

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 in order 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 6.45. 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.

```
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 6.45: Tied-Joint Slip Input

6.18.3 Building the Conventional Model

6.18.3.1 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 6.46.

6.18.3.2 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 in order 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 6.47. 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 6.48 shows a section of the input file when using a spring connection for conventional slip.

```
Rigidset
        sideset 1
end
Rigidset
        sideset 2
end

Block 33
        Spring
                Kz = Elastic 1e9
                Kx = Elastic 1e9
                Ky = Elastic 1e9
end
```

Figure 6.46: Conventional Non-Slip Input

6.18.4 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 A.38.1 and Appendix A.38.2.

```
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 6.47: 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 6.48: Conventional Slip Input with Spring

6.19 Slide RBE2. Selected DOFS

This test exercises the RBE2 element as a slider. The model and results are shown in Figure 6.49. 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 6.49 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 6.50 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.

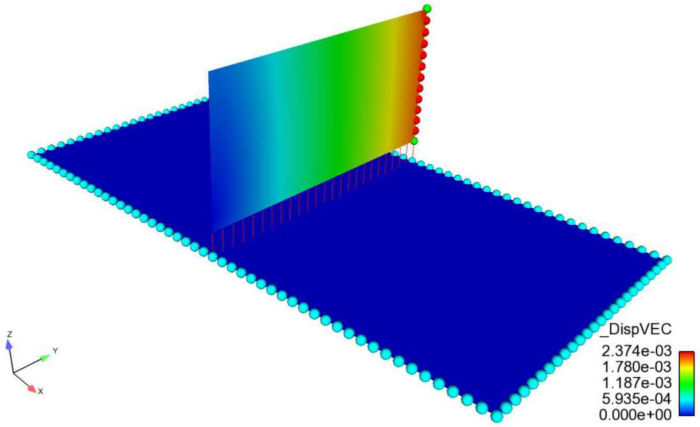


Figure 6.49: Model and Results of Selective DOF RBE2 Test

For input see Appendix A.40

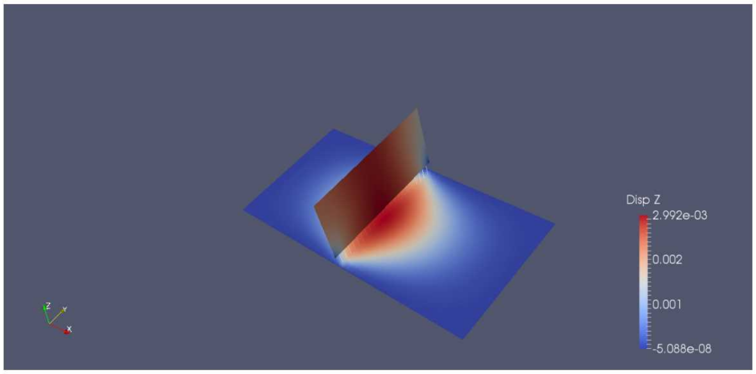


Figure 6.50: Model and Results of Orthogonally loaded Test

6.20 Thin Plate Bending

The model, shown in Figure 6.51, is a flat rectangular plate of dimension 48×24×0.5. The normal is in the Z coordinate direction. A uniform pressure is applied to the plate. Analytic expressions for the maximum displacement are found in Roark for the thin plate approximation, to which this should apply. The edges are clamped (no rotations for translations).

Table 6.25 compares the solutions from various methods and elements for this example.

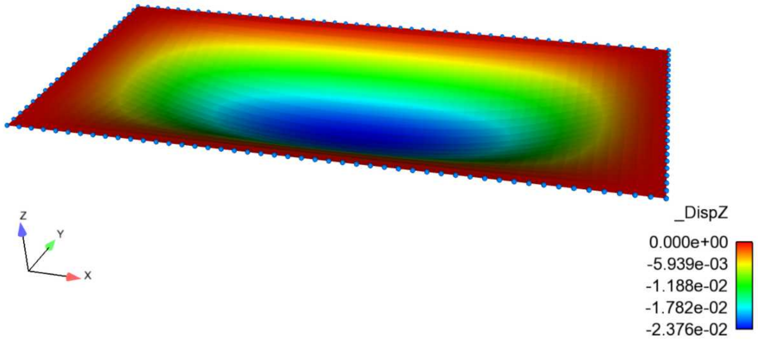


Figure 6.51: Thin Plate Bending. Geometry and Deformation

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

Table 6.25: Thin Plate Bending Center Point Solutions

For input see Appendix A.41

6.21 Spring Dashpot

6.21.1 Statics Verification: 1 translational dof at a time

6.21.1.1 Spring Translation in the X-Direction

In this example, the spring stiffness is $K = 3$, and the spring is driven by a force of $f = 7.5$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f/K.

6.21.1.2 Spring Translation in the Y-Direction

In this example, the spring stiffness is $K = 1$, and the spring is driven by a force of $f = 1$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f/K.

6.21.1.3 Spring Translation in the Z-Direction

In this example, the spring stiffness is $K = 1$, and the spring is driven by a force of $f = 1$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f/K.

6.21.1.4 Spring Translation in all Directions at Once

We now consider a spring with values in all 3 translational dofs. The axis of the spring is aligned with the x-coordinate direction, so the behavior of this system in each dof is uncoupled.

Spring Translation in the X-Direction
In the x-direction, the spring stiffness is $K_x = 1.2$, and the spring is driven by a force of $f_x = 2.4$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f_x/K_x.

Spring Translation in the Y-Direction
In the y-direction, the spring stiffness is $K_y = 1$, and the spring is driven by a force of $f_y = 1$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f_y/K_y.

Spring Translation in the Z-Direction
In the z-direction, the spring stiffness is $K_z = 1$, and the spring is driven by a force of $f_z = 3$.

After running the code, we find that the calculated solution is within 0.0001% of the exact solution, f_z/K_z.

6.21.2 Statics Verification: 1 rotational dof at a time

6.21.2.1 Spring Rotation about the X-Axis

In this example, the rotational spring stiffness is $K_r = 2.5$, and the spring is driven by a moment of $m = 6.5$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m/K_r.

6.21.2.2 Spring Rotation about the Y-Axis

In this example, the rotational spring stiffness is $K_r = 1.3$, and the spring is driven by a moment of $m = 2.6$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m/K_r.

6.21.2.3 Spring Rotation about the Z-Axis

In this example, the rotational spring stiffness is $K_r = 1$, and the spring is driven by a moment of $m = 1$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m/K_r.

6.21.2.4 Spring Rotation about all Axes at Once

We now consider a spring with values in all 3 rotational dofs. The axis of the spring is aligned with the x-coordinate direction, so the behavior of this system in each dof is uncoupled.

Spring Rotation about the X-Axis
The rotational spring stiffness about the x-axis is $K_{r_x} = 0$, and the spring is driven by a moment of $m_x = 4.6$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m_x/K_{r_x}.

Spring Rotation about the Y-Axis
The rotational spring stiffness about the y-axis is $K_{r_y} = 0$, and the spring is driven by a moment of $m_y = 3$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m_y/K_{r_y}.

Spring Rotation about the Z-Axis
The rotational spring stiffness about the z-axis is $K_{r_z} = 0$, and the spring is driven by a

moment of $m_z = 1$. After running the code, we find that the calculated solution is within 0.0001% of the exact solution, m_z/K_{r_z}.

6.21.3 Transient Dynamics Verification: 1 translational dof at a time

In the following examples, we investigate the translational response of the SpringDashpot element in each of the three coordiante directions. The equation of motion for such a system with mass M, damping B, and stiffness K is the following:

$$M\ddot{u} + B\dot{u} + Ku = f \tag{6.46}$$

The exact solution for a spring-dashpot system with a constant force f depends on the *damping ratio* $\zeta = \frac{B}{2\sqrt{km}}$. In all cases, the initial conditions $u(0) = d_0$ and $\dot{u}(0) = v_0$ are used, and we denote the *undamped natural frequency* ω_n as $\sqrt{K/M}$. For an *overdamped* system ($\zeta > 1$), the exact solution is:

$$\begin{aligned} u(t) = \frac{f}{K} + e^{-\zeta\omega_n t} \Bigg[& \frac{v_0 + (\zeta\omega_n + \omega_d)(d_0 - f/K)}{2\omega_d} e^{\omega_d t} \\ & - \frac{v_0 + (\zeta\omega_n - \omega_d)(d_0 - f/K)}{2\omega_d} e^{-\omega_d t} \Bigg] \end{aligned} \tag{6.47}$$

where the *damped natural frequency* ω_d is $\omega_n\sqrt{\zeta^2 - 1}$.

For an *underdamped* system ($\zeta < 1$), the exact solution is:

$$\begin{aligned} u(t) = \frac{f}{K} + e^{-\zeta\omega_n t} \Bigg[& (d_0 - f/K)\cos(\omega_d t) \\ & + \frac{v_0 + \zeta\omega_n(d_0 - f/K)}{\omega_d} \sin(\omega_d t) \Bigg] \end{aligned} \tag{6.48}$$

where $\omega_d = \omega_n\sqrt{1 - \zeta^2}$.

Finally, for a *critically damped* system ($\zeta = 1$), the exact solution is:

$$u(t) = \frac{f}{K} + \left\{ (d_0 - f/K) + [v_0 + \omega_n(d_0 - f/K)]\, t \right\} e^{-\omega_n t} \tag{6.49}$$

6.21.3.1 Spring-Dashpot Translation in the X-Direction

In this example, the spring-dashpot stiffness is $K = 4$, the damping is $B = 5$, and is driven by a force of $f = 10$.

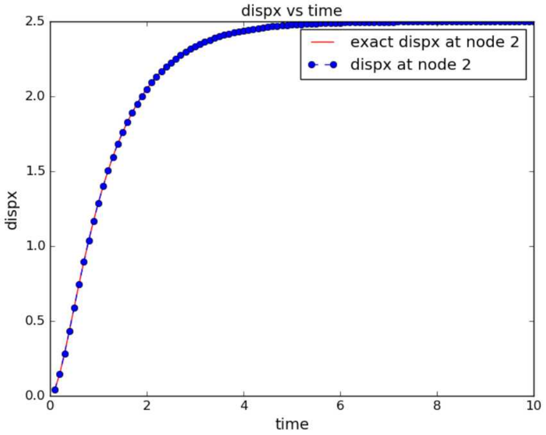


Figure 6.52: Comparison of exact and calculated x-displacement

6.21.3.2 Spring-Dashpot Translation in the Y-Direction

In this example, the spring-dashpot stiffness is $K = 4$, the damping is $B = 4$, and is driven by a force of $f = 10$.

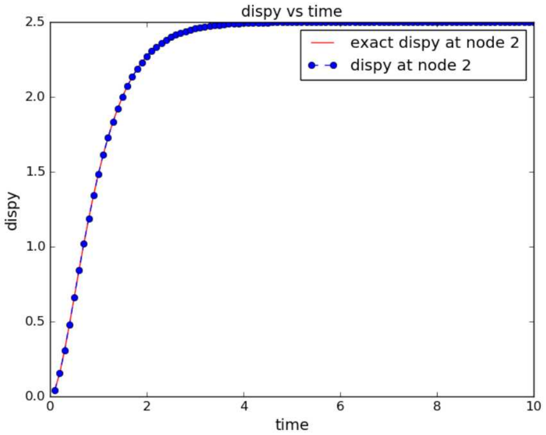


Figure 6.53: Comparison of exact and calculated y-displacement

6.21.3.3 Spring-Dashpot Translation in the Z-Direction

In this example, the spring-dashpot stiffness is $K = 4$, the damping is $B = 3$, and is driven by a force of $f = 10$.

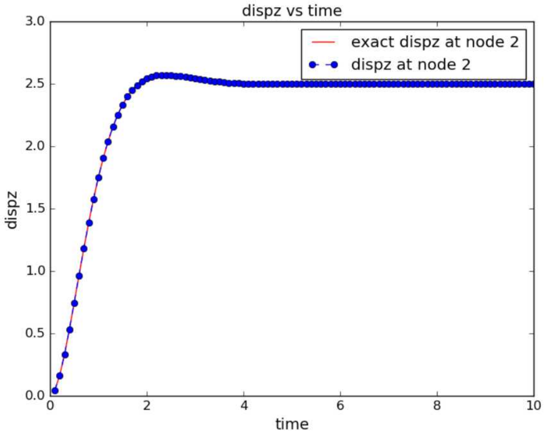


Figure 6.54: Comparison of exact and calculated z-displacement

6.21.4 Transient Dynamics Verification: all Translational and Rotational dofs at once

In this example, we consider a spring-dashpot with values in all 6 translational and rotational dofs. The axis of the spring is aligned with the x-coordinate direction, so the behavior of this system in each dof is uncoupled. The results are summarized below.

6.21.4.1 Spring-Dashpot Translation in the X-Direction

The spring-dashpot stiffness in the x-direction is $K = 6.5$, the damping is $B = 1.23$, and the x-component of the force is $f = 1$.

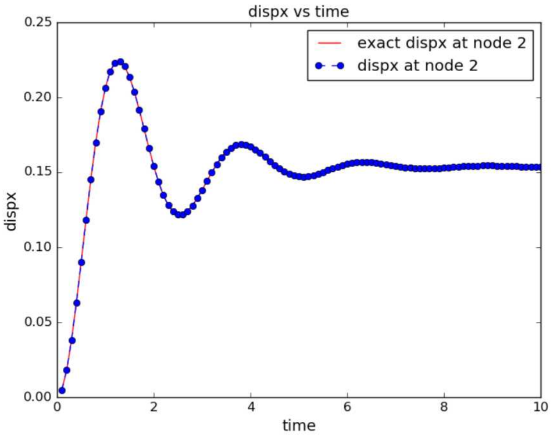


Figure 6.55: Comparison of exact and calculated x-displacement

6.21.4.2 Spring-Dashpot Translation in the Y-Direction

The spring-dashpot stiffness in the y-direction is $K = 7.8$, the damping is $B = 3.21$, and the y-component of the force is $f = 1$.

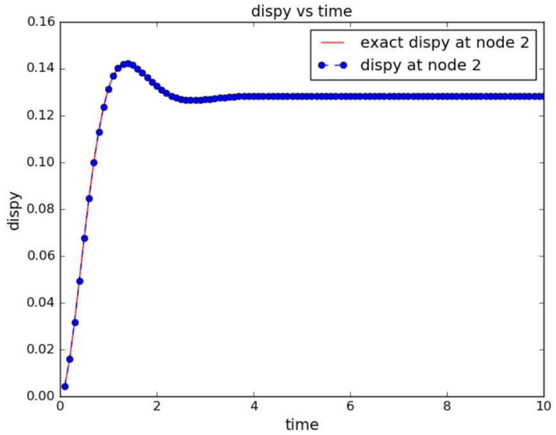


Figure 6.56: Comparison of exact and calculated y-displacement

6.21.4.3 Spring-Dashpot Translation in the Z-Direction

The spring-dashpot stiffness in the z-direction is $K = 5.23$, the damping is $B = 2.34$, and the z-component of the force is $f = 1$.

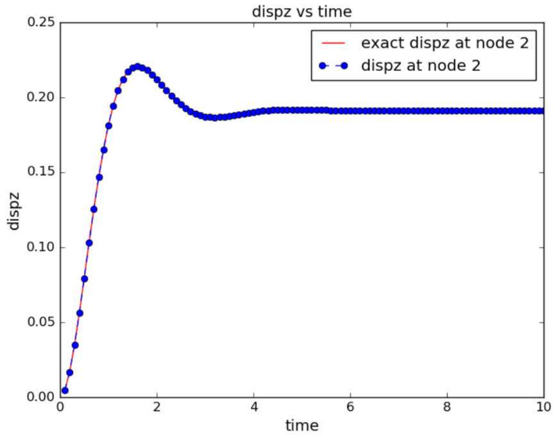


Figure 6.57: Comparison of exact and calculated z-displacement

6.21.4.4 Spring-Dashpot Rotation about the X-Axis

The spring-dashpot stiffness about the x-axis is $K = 8.75$, the damping is $B = 2.1$, and the x-component of the moment is $m = 1$.

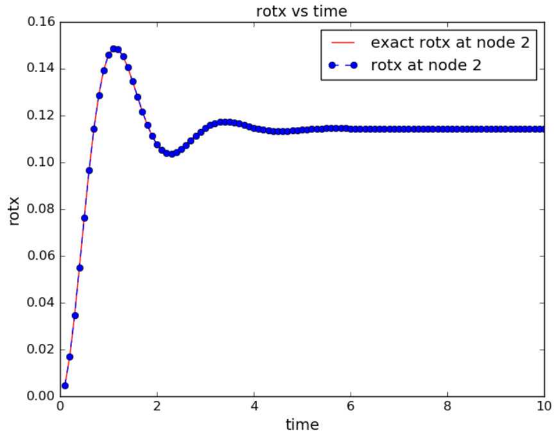


Figure 6.58: Comparison of exact and calculated x-rotation

6.21.4.5 Spring-Dashpot Rotation about the Y-Axis

The spring-dashpot stiffness about the y-axis is $K = 6.45$, the damping is $B = 1.32$, and the y-component of the moment is $m = 1$.

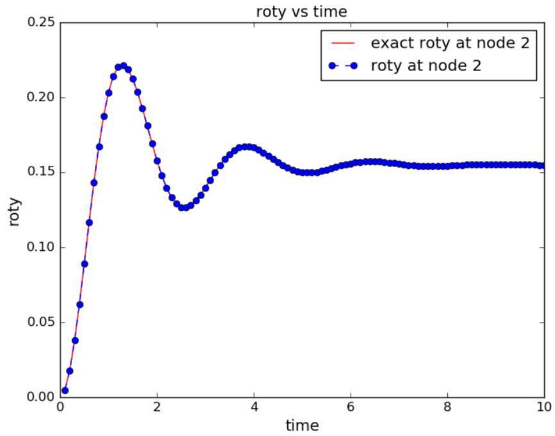


Figure 6.59: Comparison of exact and calculated y-rotation

6.21.4.6 Spring-Dashpot Rotation about the Z-Axis

The spring-dashpot stiffness about the z-axis is $K = 9.78$, the damping is $B = 2.3$, and the z-component of the moment is $m = 1$.

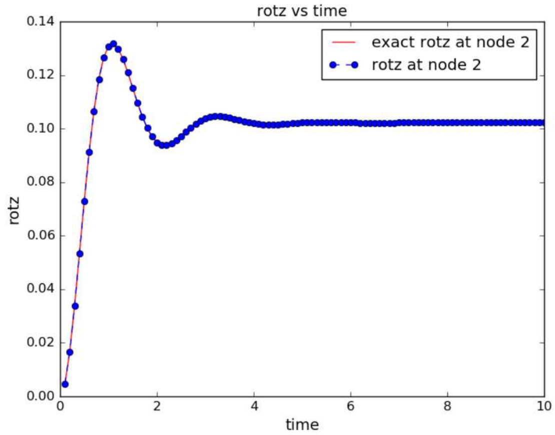


Figure 6.60: Comparison of exact and calculated z-rotation

6.21.5 Transient Dynamics Verification: all Translational and Rotational dofs at once, with a coordinate transformation

In this example, we consider a spring-dashpot with values in all 6 translational and rotational dofs. Additionally, a coordinate tranformation is utilized such that the axis of the spring is not alligned with the x-coordinate direction as seen in figure 6.61, leading to uncouped behavior in in the dofs of this system.

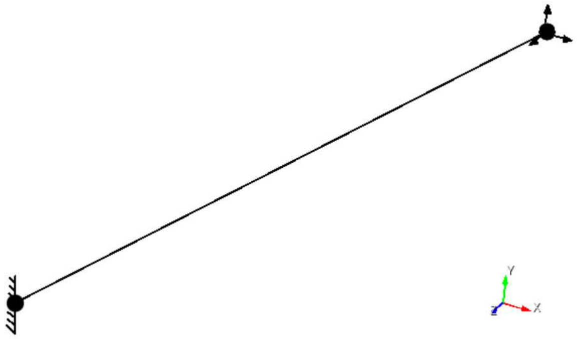


Figure 6.61: Figure showing the position, boundary conditions, and force applied to a spring in the global coordiante space {x,y,z}

For a given local (i.e. with respect to the spring element) coordinate system $l =$

$\{\hat{x'},\ \hat{y'},\ \hat{z'}\}$, we can transform the stiffness and damping terms to the global coordinate system using the following equations:

$$\begin{aligned} K^{(g)} &= R' K^{(l)} R \\ B^{(g)} &= R' B^{(l)} R, \end{aligned} \tag{6.50}$$

where K^g and B^g are the stiffness and damping matrices defined in the global coordinate system $g = \{\hat{x},\ \hat{y},\ \hat{z}\}$, and where the rotation matrix R is defined as

$$R = \begin{bmatrix} \hat{x'} \\ \hat{y'} \\ \hat{z'} \end{bmatrix} \tag{6.51}$$

In this example, the x-axis of the local coordinate system lies in line with the direction of the spring element as seen in figure 6.61, or $x' = \{1, 1, 1\}$, with the local origin at the fixed point of the spring, $\{0, 0, 0\}$. Additionally, we chose the (global) x-axis $x = \{1, 0, 0\}$ to be on the local x-y plane, or eqivalently for the local z-axis to be in the direction $\{0, 1, -1\}$. This local coordinate system leads to $x' = \{\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3\}$, $\hat{y'} = \{\sqrt{6}/3, -\sqrt{6}/6, -\sqrt{6}/6\}$, $\hat{z'} = \{0, \sqrt{2}/2, -\sqrt{2}/2\}$.

The translational spring-dashpot stiffness in the (local) x, y, and z-directions respectively are $K_x = 6.5$, $K_y = 7.8$, and $K_z = 5.23$. Likewise, the three components of the tranlational damping are $B_x = 1.23$, $B_y = 3.21$, and $B_z = 2.34$. The tranlational dofs of the spring-dashpot are driven by a force of $f = \{1,\ 1,\ 1\}$.

The rotational spring-dashpot stiffness about the (local) x, y, and z-axes respectively are $K_{r_x} = 8.75$, $K_{r_y} = 6.45$, and $K_{r_z} = 9.78$. Likewise, the three components of the rotational damping are $B_{r_x} = 2.1$, $B_{r_y} = 1.32$, and $B_{r_z} = 2.3$. The rotational dofs of the spring-dashpot are driven by a moment of $m = \{1,\ 1,\ 1\}$.

Since we have included damping in our analysis, over a long period of time, the translational solution will tend towards the static solution, ${K^{(g)}}^{-1} f$. Likewise, the rotational solution will tend towards ${K_r^{(g)}}^{-1} m$.

The displacement and rotational results in each of the global dofs are summarized below.

6.21.5.1 Spring-Dashpot Translation in the (global) X-Direction

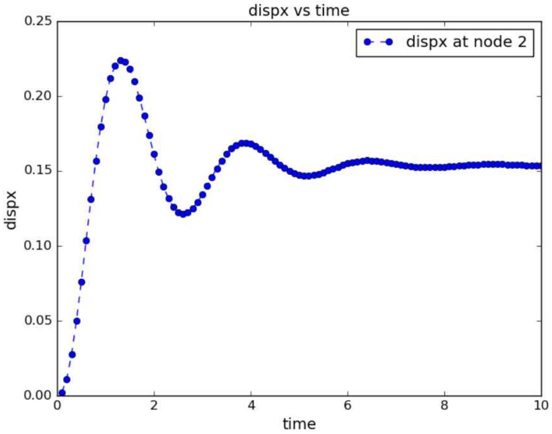


Figure 6.62: X-displacement over time

6.21.5.2 Spring-Dashpot Translation in the (global) Y-Direction

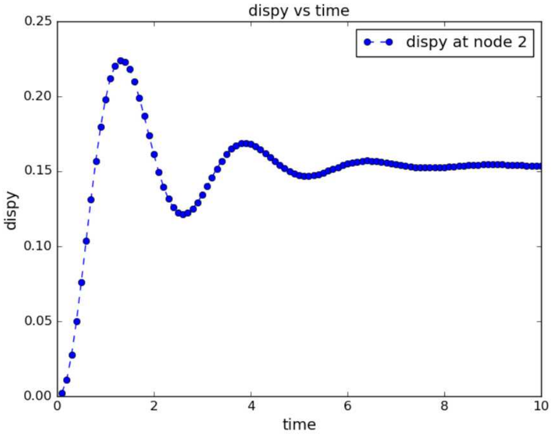


Figure 6.63: Y-displacement over time

6.21.5.3 Spring-Dashpot Translation in the (global) Z-Direction

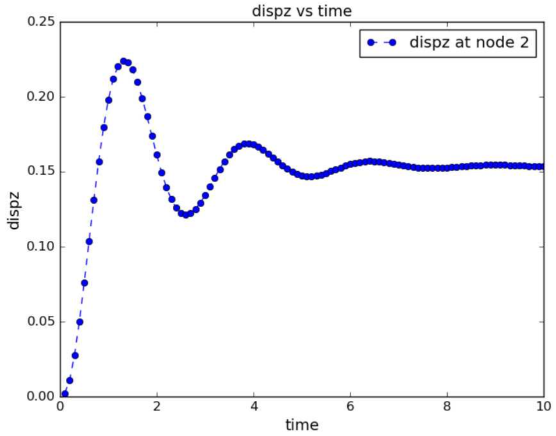


Figure 6.64: Z-displacement over time

6.21.5.4 Spring-Dashpot Rotation about the (global) X-Axis

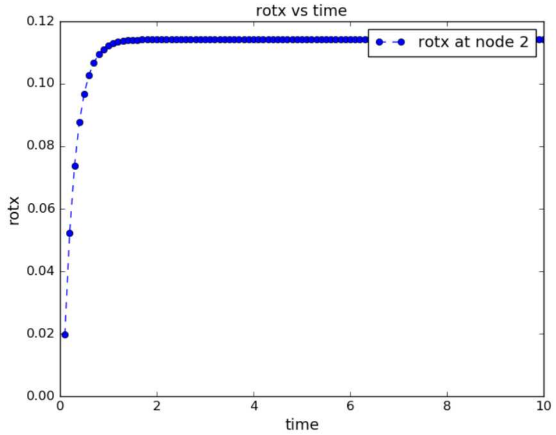


Figure 6.65: X-rotation over time

6.21.5.5 Spring-Dashpot Rotation about the (global) Y-Axis

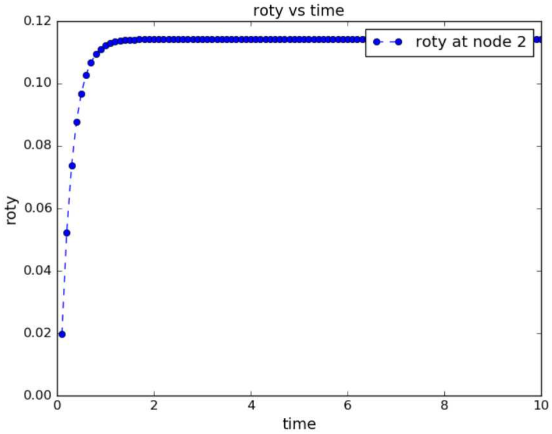


Figure 6.66: Y-rotation over time

6.21.5.6 Spring-Dashpot Rotation about the (global) Z-Axis

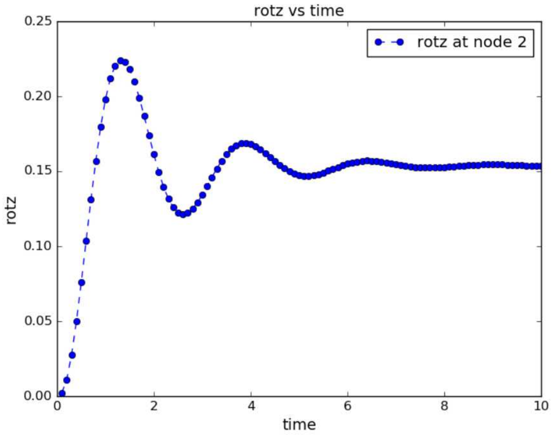


Figure 6.67: Z-rotation over time

For the general aprepro compatable input decks used in each of the three basic cases above (statics, transient dynamics, and transient dynamics with a coordiante transformation) see Appendix A.10.

Salinas$_{r}test_{v}erification_{m}anual_{62}1_{S}pringDashpot_{S}pringDashpotVerification_{S}pringDashpotVerifica$

Chapter 7

Solutions in Rotating Coordinate Frames

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

7.1 Rotating Dumbbell Statics

7.1.1 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 bar is massless. See Figure 7.1.

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.

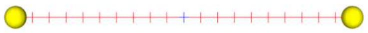


Figure 7.1: Dumbbell Geometry

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

7.1.2 Analytic Results

Each mass on either end of the rotating bar should experience only centrifugal loadings. 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 masses are located, as the other points are massless. For input deck see Appendix A.50.

7.2 Rotating Beam 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

$$\begin{aligned} F &= \Omega^2 \int_r^L x\, dm \\ &= \Omega^2 \int_r^L x \rho A\, dx \\ &= \rho A \Omega^2 (L^2 - r^2)/2. \end{aligned} \tag{7.1}$$

Thus, the axial stress at r is

$$\sigma(r) = F/A = \rho\Omega^2(L^2 - r^2)/2. \tag{7.2}$$

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

$$\epsilon(r) = \sigma(r)/E = \rho\Omega^2(L^2 - r^2)/(2E). \tag{7.3}$$

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

$$\begin{aligned} u(r) &= \int_0^r \epsilon(x)\, dx \\ &= \rho\Omega^2/(2E) \int_0^r (L^2 - x^2)\, dx \\ &= \rho\Omega^2/(2E)(L^2 x - x^3/3)|_0^r \\ &= \frac{\rho\Omega^2 L^3}{6E}[3(r/L) - (r/L)^3]. \end{aligned} \tag{7.4}$$

For input deck see Appendix A.51.

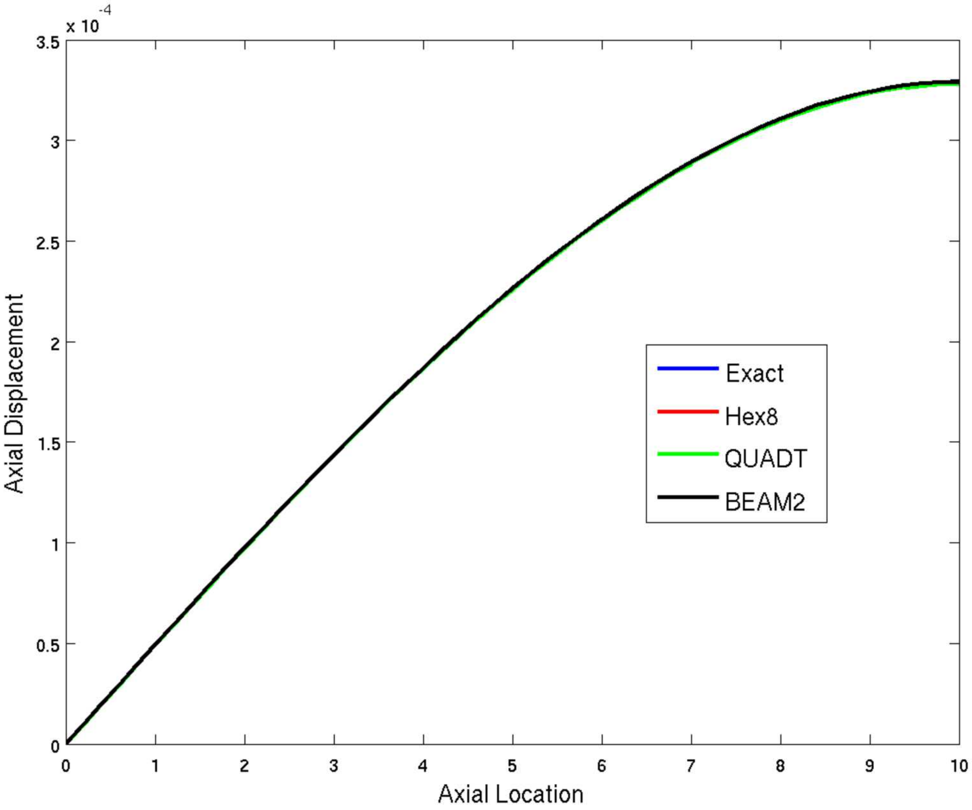


Figure 7.2: Comparisons of axial deformations with exact solution for a beam.

7.3 Rotating Shell Statics

The rotating shell example is similar to the rotating beam 7.2 with $E = 19.5 \times 10^{10}$, $L = 10$, $\rho = 7700$, $A = 1$, and $\Omega = 5$. Three different finite element meshes of the beam were constructed. The first one is a `HEX8` mesh with 50 elements in the x-direction and 5 elements in both the y and z-directions (the dimensions of the beam in the three coordinate directions are 10, 1, and 1). The second one is a quadrilateral shell mesh with 50 elements in the x-direction and 5 elements in the y-directions. The third one is a mesh of beams with 50 elements in the x-direction. Comparisons of axial deformations for three finite element analyses are shown in Figures 7.2 and 7.3. 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 A.52.

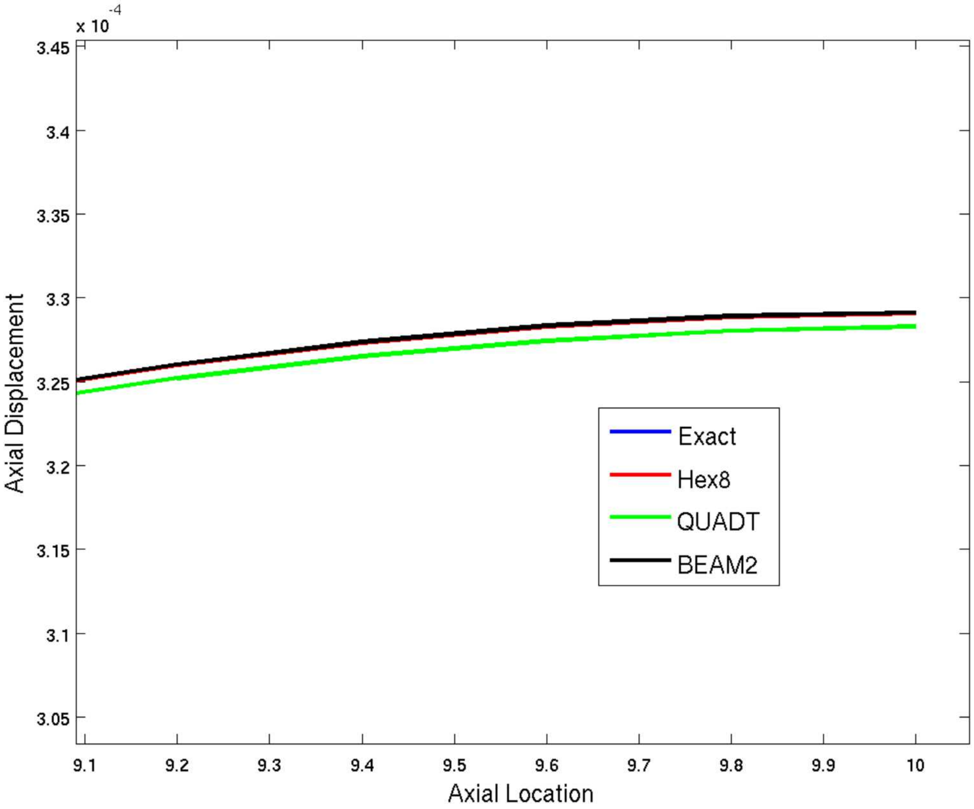


Figure 7.3: Zoomed in view of Figure 7.2 showing differences for QUADT elements.

7.4 Rotating Ring Statics

7.4.1 Introduction and Purpose

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 7.4 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.

7.4.1.1 Analytical Results

The resulting forces are given by,

$$\begin{aligned} \vec{F} &= \int \rho\vec{\Omega} \times \vec{\Omega} \times \vec{r}\,dV && (7.5) \\ &= 1.1^2 2.0\,(\rho V_n)\,\hat{r} && (7.6) \end{aligned}$$

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,

$$\begin{aligned} \vec{F} = & \quad \int \rho\frac{d}{dt}\vec{\Omega} \times \vec{r}\,dV && (7.7) \\ = & \quad 1.12.0\,(\rho V_n)\,\hat{r} && (7.8) \end{aligned}$$

and the resulting force is 0.2054/1.1 units outward.

7.4.2 What is tested

The test evaluates the following:

- The centrifugal force in a rotating system.
- The force on shells with rotational degrees of freedom. The moment should be zero.
- A coordinate translation.

It does *not* test,

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

For input deck see Appendix A.53.

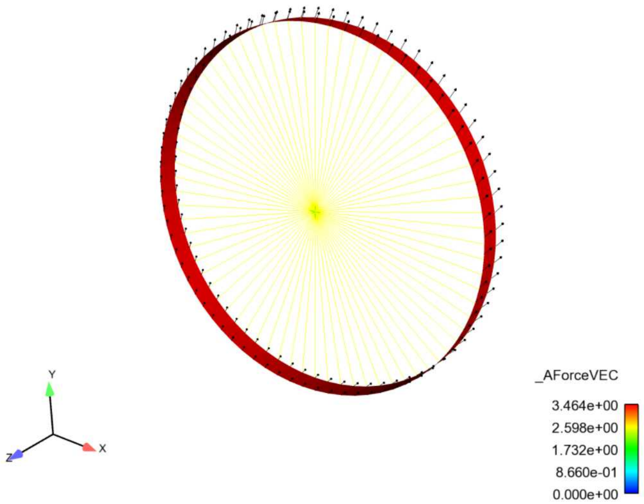


Figure 7.4: Rotating Ring Geometry and Results

7.5 Rotating Ring Acceleration

This is a variation on the static analyses of a rotating ring described in section 7.4. Here an angular acceleration is applied instead of the angular velocity in 7.4. By hand, the angular acceleration is .2054/1.1 force units. For input deck see Appendix A.54.

7.6 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,

$$\begin{aligned} F_{centrifugal} &= \vec{\Omega} \times (\vec{\Omega} \times \vec{r}) && (7.9) \\ &= [\Omega]^T [M] [\Omega] [r] && (7.10) \end{aligned}$$

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 7.5.

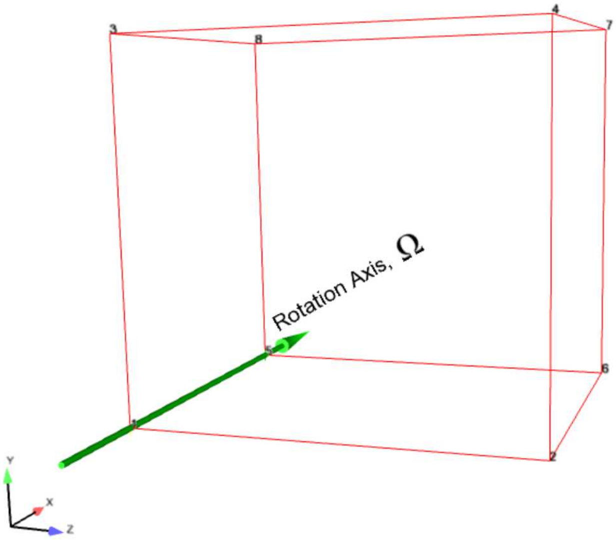


Figure 7.5: Rotating Hex Geometry

7.6.1 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 insure that the model reduction process is consistent.

3. We examine the loading of a superelement equivalent to the single hex element.

Each of these tests is described in a little more detail in what follows.

Single Hex Rotation

Equation 7.10 describes the load calculation for a single hex in a coordinate frame rotating at a constant angular velocity. The results of the loading have been examined visually for reasonable response, but no strict verification of these results are available. The loading vectors are not entirely radial (as expected).

The analysis is singular, i.e. the body has a zero energy mode and is free to rotate about the axis of rotation. Because of this, only the forces are evaluated - comparison of displacements could result in errors from inaccurate solution of the singular system. The force response is shown in Figure 7.6.

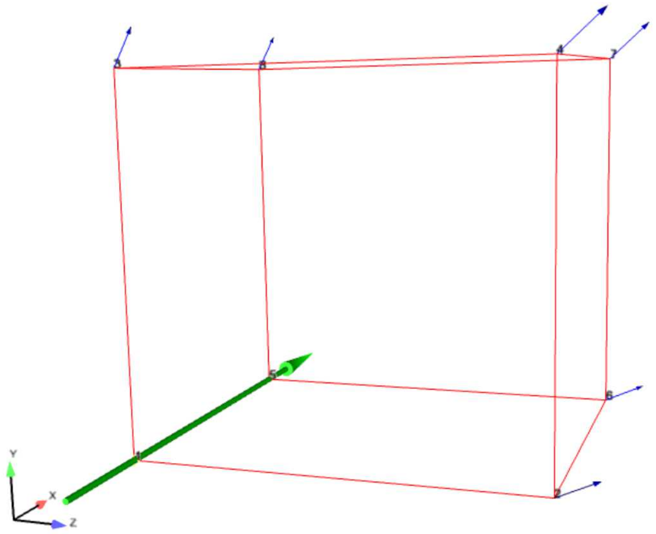


Figure 7.6: Rotating Hex Response

7.6.1.1 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.

7.6.1.2 Loading of a Superelement

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.

7.6.2 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.

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

When the full shape functions are available, this can be evaluated as a discretized linear algebra system (equation 7.10). 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 of 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 A.55.

7.7 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,

$$\begin{aligned}
u(r) &= \int_0^r \epsilon(x)\,dx \\
&= \rho\Omega^2/(2E) \int_0^r (L^2 - x^2)\,dx \\
&= \rho\Omega^2/(2E)(L^2 x - x^3/3)|_0^r \\
&= \frac{\rho\Omega^2 L^3}{6E}[3(r/L) - (r/L)^3]. \qquad (7.11)
\end{aligned}$$

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 7.7. 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 A.56.

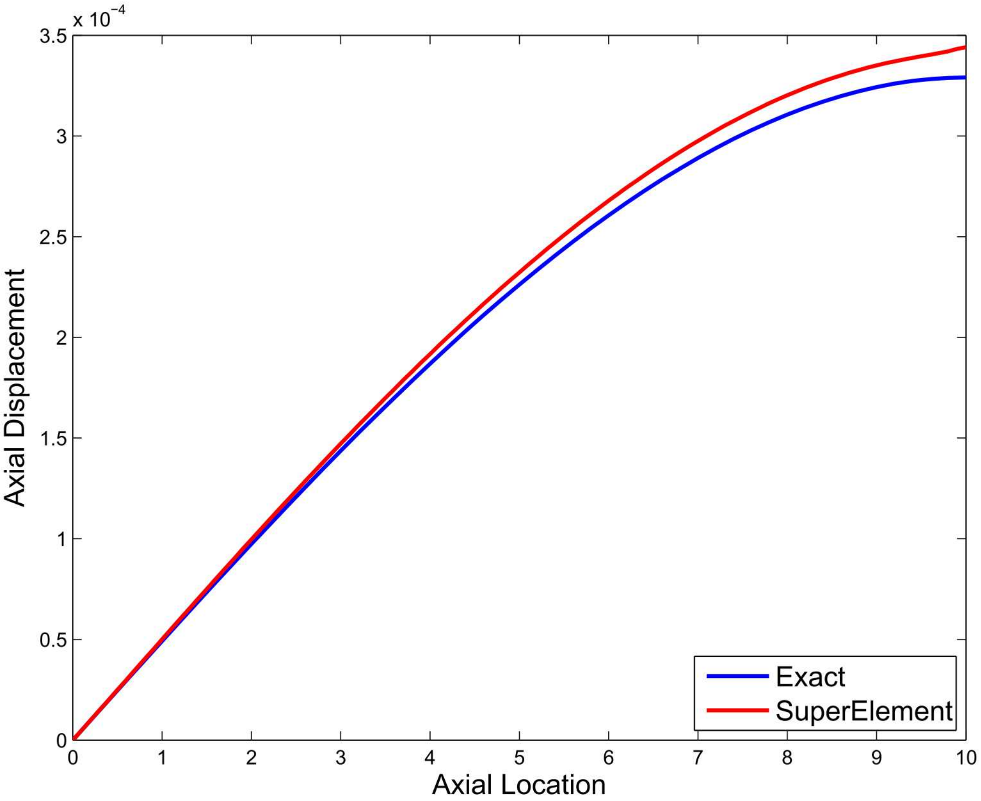


Figure 7.7: Comparisons of axial deformations with exact solution for a beam.

7.8 Point Mass in a Rotating Frame

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 7.8.

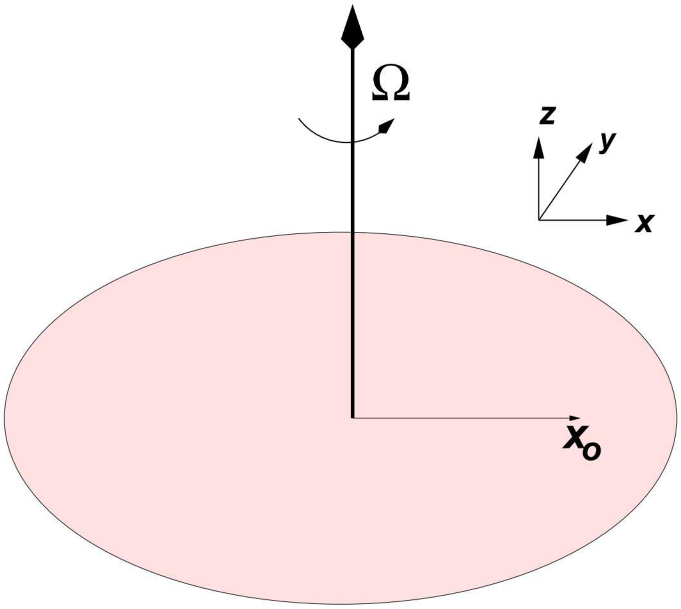


Figure 7.8: Rotating Frame Geometry

7.8.1 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,

$$x' = x_o \cos(-\Omega t) \quad (7.12)$$
$$y' = x_o \sin(-\Omega t) \quad (7.13)$$

7.8.2 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.

7.8.2.1 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.

7.8.2.2 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,

$$\begin{aligned} x &= x_o && (7.14)\\ y &= v_o t && (7.15)\\ &= x_o \Omega t && (7.16) \end{aligned}$$

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

$$\begin{aligned} r &= \sqrt{x^2 + y^2} && (7.17)\\ &= x_o\sqrt{1 + (\Omega t)^2} && (7.18)\\ \theta &= \tan^{-1}(y/x) && (7.19)\\ &= \tan^{-1}(\Omega t) && (7.20) \end{aligned}$$

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,

$$\tan^{-1}(\theta) \;\approx\; \theta - \frac{\theta^3}{3} + \frac{\theta^5}{5} + ... \tag{7.21}$$

$$\sqrt{1+\epsilon^2} \;\approx\; 1 + \frac{\epsilon^2}{2} \tag{7.22}$$

$$\tag{7.23}$$

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 7.9 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.

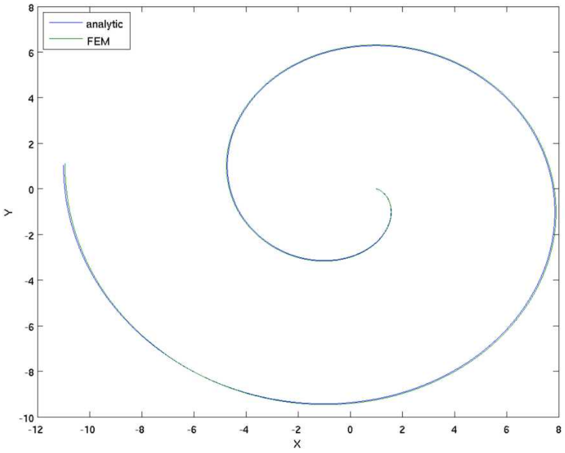


Figure 7.9: Point Mass initially at rest in rotating frame

7.8.3 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,

$$y = 0 \tag{7.24}$$
$$x = x_o(1 - 2\Omega t/\pi) \tag{7.25}$$

or,

$$r = x_o(1 - 2\Omega t/\pi) \tag{7.26}$$
$$\theta = 0 \tag{7.27}$$

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

$$x' = x_o(1 - 2\Omega t/\pi)\cos(\Omega t) \tag{7.28}$$
$$y' = x_o(1 - 2\Omega t/\pi)\sin(\Omega t) \tag{7.29}$$

For input deck see Appendix A.57.

This page intentionally left blank.

Chapter 8

Inverse Methods

8.1 Force Identification from Structural Acoustic Frequency Responses

The structural-acoustic frequency domain force identification capability is demonstrated using the synthetic response at three frequencies. The geometry of this test is shown in Figure 8.1.

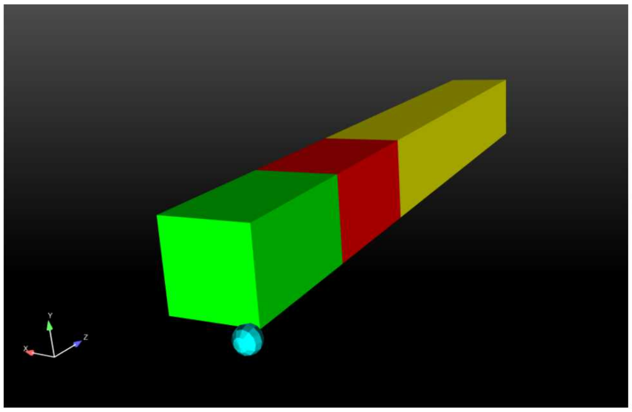


Figure 8.1: Force Inversion Test Geometry

The model contained three regions, as shown in Figure 8.1: two steel regions, represented by the red and green blocks, and a region of air, represented by the yellow block. Tied constraints were assigned at the steel-steel and steel-air interfaces to connect these regions. A concentrated mass, represented in cyan, was placed at one end of the model and was connected with a spring connection. Acoustic loading was applied to one side of the air region, shown in blue in Figure 8.2. The frequency of the acoustic loading was varied between 10 Hz, 20 Hz, and 30 Hz.

Synthetic input data was generated by performing a forward problem on the coupled air-steel model using known acoustic velocity amplitudes for two loading functions, F_1 and F_2. Displacement values were determined for a node set on the end of the green steel block, while acoustic pressure values were determined for a node set on the side of the air region. The data was then used in the inverse problem to verify that the code could recover the original

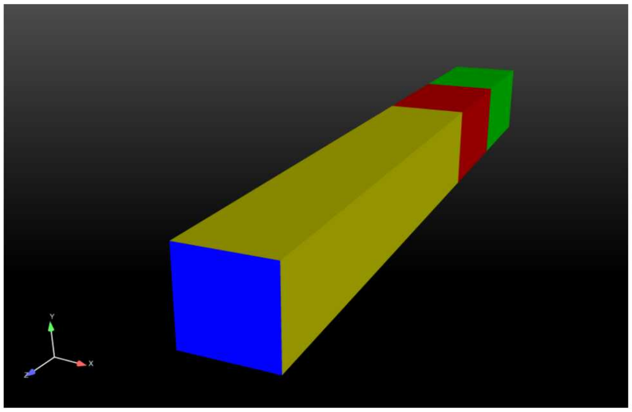


Figure 8.2: Side of model with applied acoustic loading represented in blue

acoustic velocity inputs for the two acoustic loading functions. The test, which used a full Newton algorithm with analytic Hessians, was analyzed by comparing the obtained acoustic velocity amplitudes to the expected values. Table 8.1 compares the expected acoustic velocity values to values obtained through the inverse problem (Exp/Obt).

Table 8.1: Acoustic Velocity Values

Frequency (Hz)	F_1 (Exp/Obt)	F_2 (Exp/Obt)
10	4.0/4.0	5.0/5.0
20	4.0/4.0	5.0/5.0
30	4.0/4.0	5.0/5.0

At each frequency tested, acoustic velocity values obtained in the inverse problem matched expected values of 4.00 and 5.00 for forcing functions F_1 and F_2, respectively. The test was performed both in serial and parallel. For each run, the ROL optimization performed three iterations, providing appreciable convergence for the objective function and gradient. Figure 8.3 shows the convergence behavior of the objective function and gradient values for the serial run; optimization results for the parallel run matched the serial run results.

For input deck see Appendix A.58.

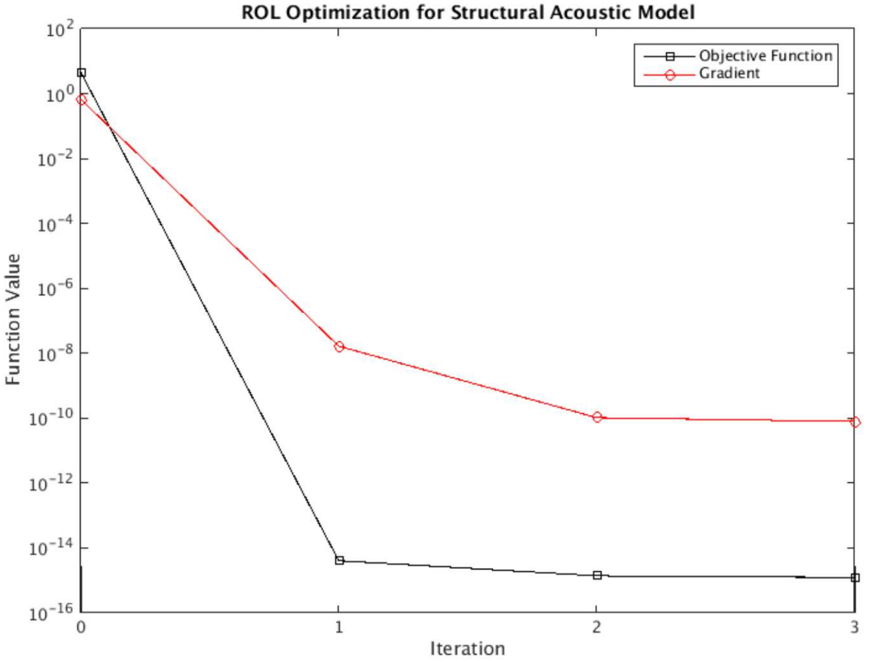


Figure 8.3: ROL Optimization of Objective Function and Gradient

8.2 Force Identification from Frequency Responses

The frequency domain force identification capability is demonstrated using the synthetic response and three frequencies. The cubic geometry of this test is shown in Figure 8.4.

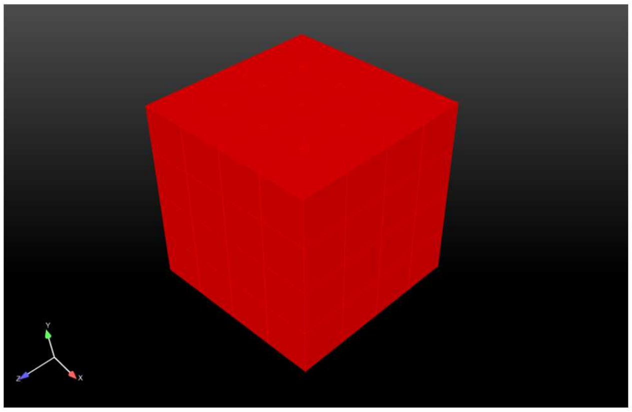


Figure 8.4: Force Inversion Test Geometry

One boundary condition constrained the model. An absorbing boundary was specified on one side, emulating a non-reflecting condition. Acoustic loading was applied to the opposite side, applied at frequencies of 1, 2, and 3 Hz. Figure 8.5 indicates sides of the model with specified boundary conditions:

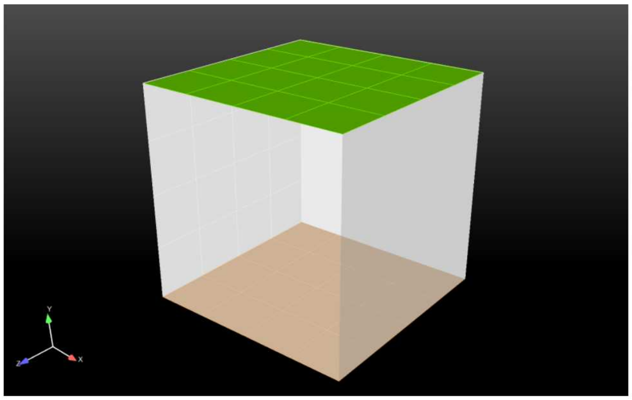


Figure 8.5: Sides with absorbing boundary side (green) and with acoustic loading (orange)

The model was discretized using 64 Hex-8 elements, arranged in a 4×4×4 cube. Synthetic input data for the inverse problem was generated by running a forward problem with the amplitudes shown in Table 8.2, and was comprised of acoustic pressure values specified at element nodes. This data was then used in the inverse problem to verify that the code could recover the original acoustic velocity inputs. The test, which used a full Newton algorithm with analytic Hessians, was analyzed by comparing obtain acoustic velocity amplitudes of three loading functions–F_1, F_2, and F_3– to the expected values. The expected values and obtained results (Exp/Obt) for the acoustic loading functions are shown in Table 8.2.

Table 8.2: Acoustic Velocity Values

Frequency (Hz)	F_1 (Exp/Obt)	F_2 (Exp/Obt)	F_31 (Exp/Obt)
1	10.0/10.0	5.0/5.0	1.0/1.0
2	10.0/10.0	5.0/5.0	2.0/2.0
3	10.0/10.0	5.0/5.0	3.0/3.0

Acoustic velocity for loading functions F_1 and F_2 matched the expected values of 10.0 and 5.0, respectively. The acoustic velocity of loading function F_3 demonstrated linear dependence on frequency, also as expected. Optimization using ROL allowed for appreciable convergence of the objective function and gradient; the convergence history of the objective function and gradient, minimized in three iterations, is illustrated in Figure 8.6:

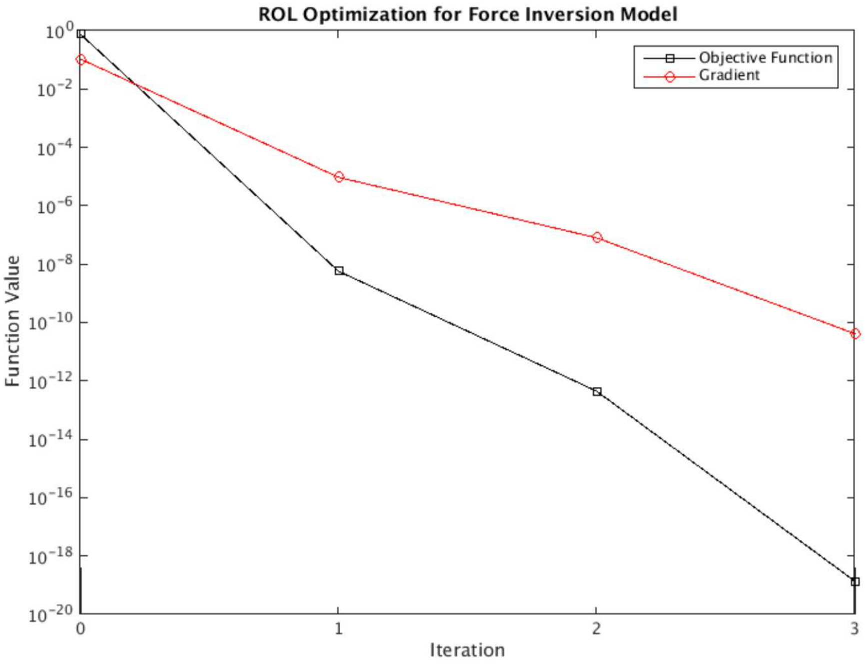


Figure 8.6: ROL Optimization of Objective Function and Gradient

For input deck see Appendix A.59.

8.3 Force Identification from Temporal Pressures

The temporal force identification capability is demonstrated using synthetic displacement data generated from a 1 Hertz pressure field. The cubic geometry of this test is shown in Figure 8.7.

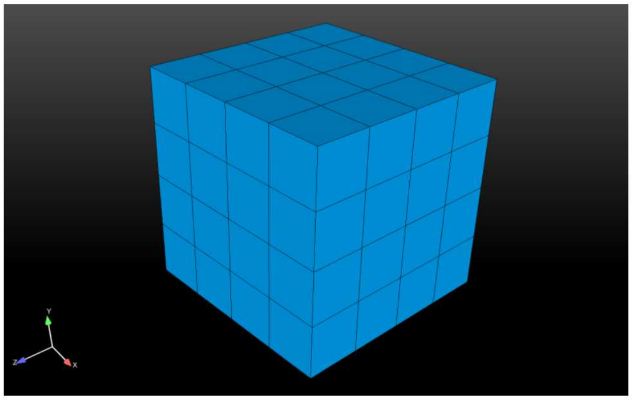


Figure 8.7: Pressure Inversion Test Geometry

The model consists of 64 Hex-8 elements, arranged in a 4×4×4 cube , and was assigned as a solid material. One boundary condition constrained the model, with one side assigned as a fixed boundary. A distributed pressure load acted normal to the side opposite of the fixed side. Figure 8.8 shows the sides with boundary (yellow) and loading (pink) conditions.

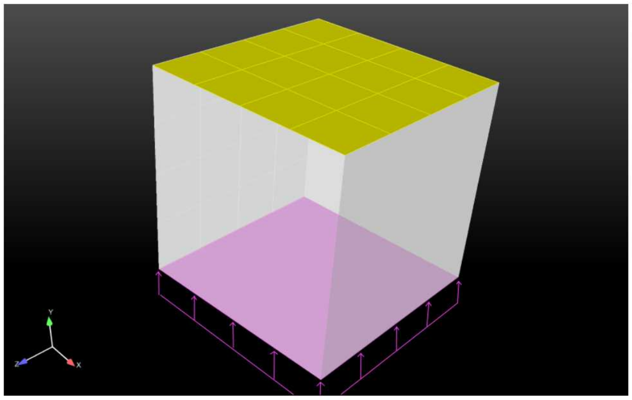


Figure 8.8: Fixed (yellow) and pressure-loaded (pink) sides in model

Synthetic input data was generated by performing a forward problem on the model, using a periodic distributed pressure load function with a known magnitude of 1 and frequency of 1 Hz. The data generated represented elastic displacements measured at element nodes. The data was used in the inverse problem to verify that the code could recover the original time history of the pressure loading. The test, which used a full Newton method with cubic interpolation linesearch, was analyzed by comparing the obtained pressure loading time-history with the original loading function. Figure 8.9 compares the inverse-problem results with the original function.

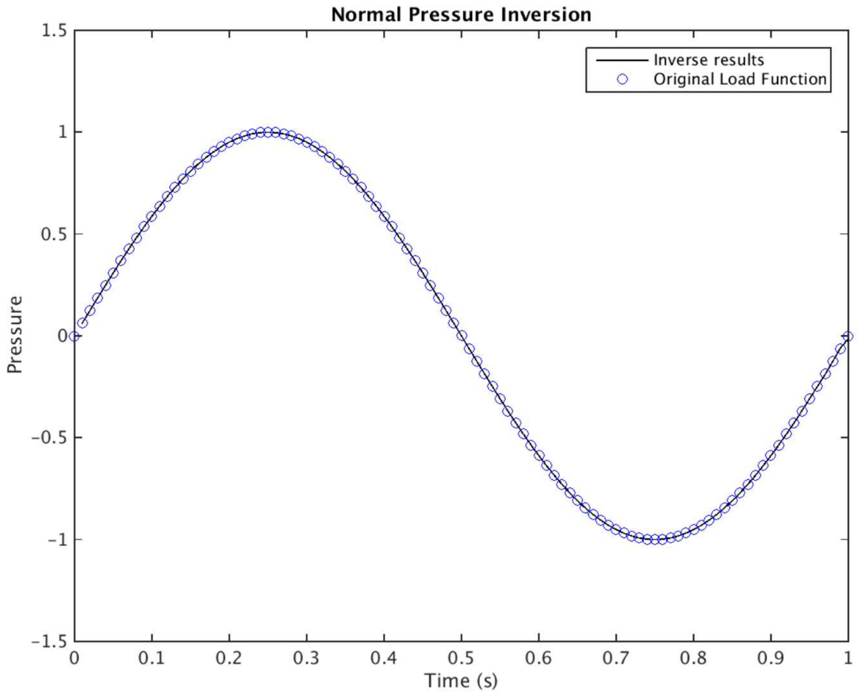


Figure 8.9: Inverse-problem results for elastic pressure loading time-history

As shown in Figure 8.9, the inverse problem results nearly exactly matched the original forcing function. The optimization method ran through four iterations and achieved significant convergence for the objective function and gradient. The test was conducted in both series and parallel; Figure 8.10 shows the convergence history for the serial run, though parallel results essentially matched the serial results.

For input deck see Appendix A.60.

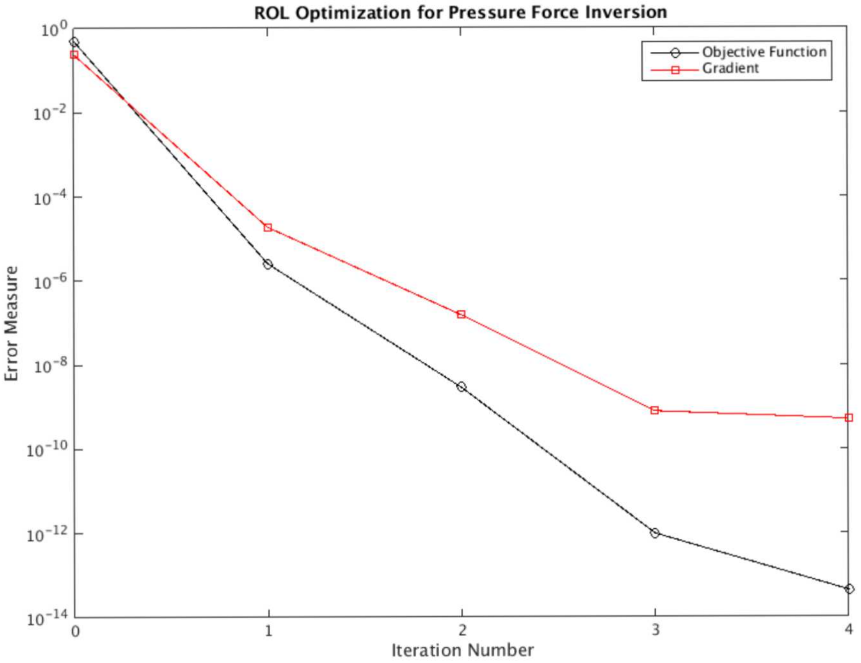


Figure 8.10: ROL Optimization of Objective Function and Gradient

8.4 Force Identification from Temporal Tractions

The temporal force identification capability is demonstrated using synthetic displacement data generated from a 1 Hertz traction field. The cubic geometry of this test is shown in Figure 8.11.

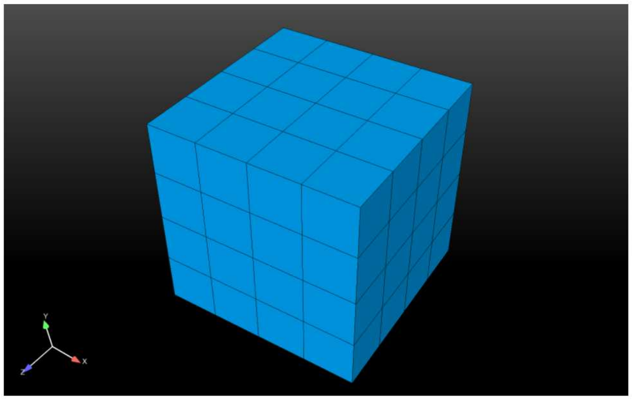


Figure 8.11: Force Inversion Test Geometry

The model consists of 64 Hex-8 elements, arranged in a 4×4×4 cube , and was assigned as a solid material. The model was constrained by one boundary condition, one side assigned as a fixed boundary. A traction load, including a normal component and two orthogonal shear components, acted on the side opposite of the fixed side and had a direction of $(x = 1, y = 2, z = 3)$. Figure 8.12 shows the sides with boundary (green) and loading (orange) conditions.

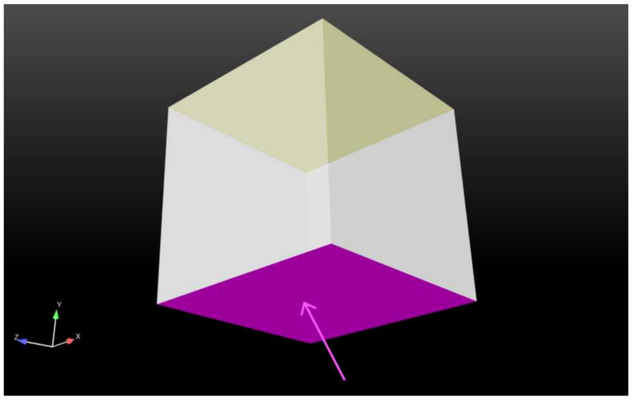


Figure 8.12: Fixed boundary (yellow) and traction loaded (pink) sides in model

Synthetic input data was generated by performing a forward problem on the model, using a periodic traction load with a known magnitude of 1 and frequency of 1 Hz. The data generated represented displacements measured for element nodes. The data was then used in the inverse problem to verify that the code could recover the original time history of the traction load. The test, which used a full Newton method with analytic Hessians,

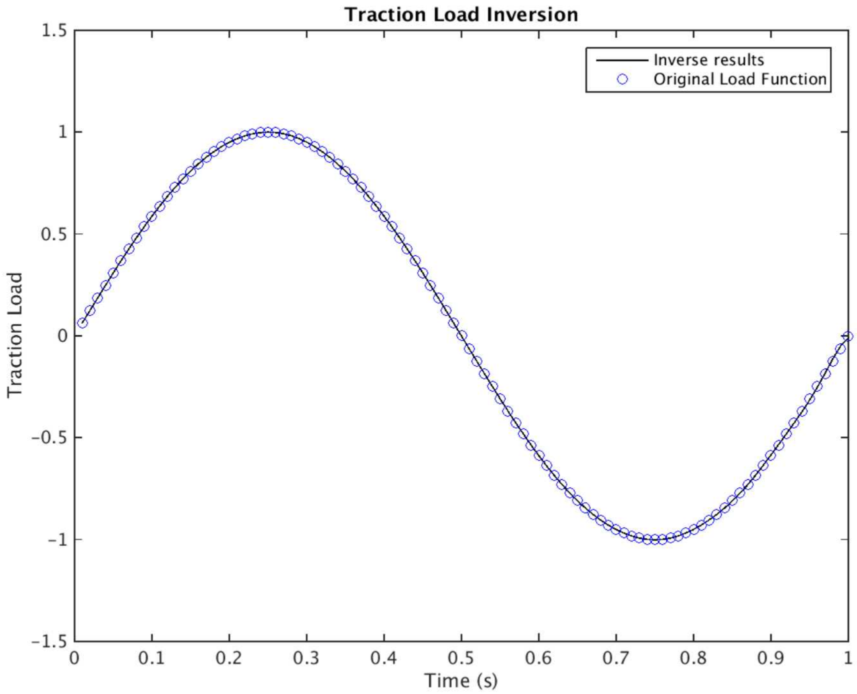


Figure 8.13: Inverse-problem results for traction loading time-history

was analyzed by comparing the obtained traction load time-history with the original loading function. Figure 8.13 compares the inverse-problem results with the original function:

As shown in Figure 8.13, the inverse problem results exactly matched the original forcing function. Optimization using ROL provided for significant convergence of the objective function and its gradient. The test was conducted in both series and parallel, each run undergoing three iterations; Figure 8.14 shows the convergence history for the serial run.

For input deck see Appendix A.61.

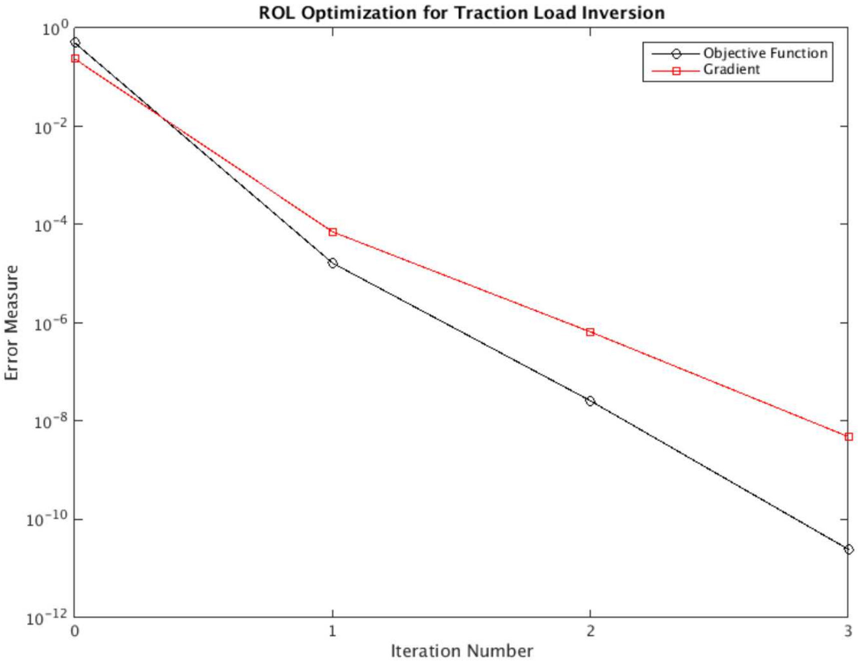


Figure 8.14: Convergence History for ROL Optimization of Transient Traction Load Inversion

8.5 Force Identification from Temporal Acoustic Pressures

The temporal force identification capability is demonstrated using synthetic displacement data generated from a 1 Hertz acoustic pressure field. The cubic geometry of this test is shown in Figure 8.15.

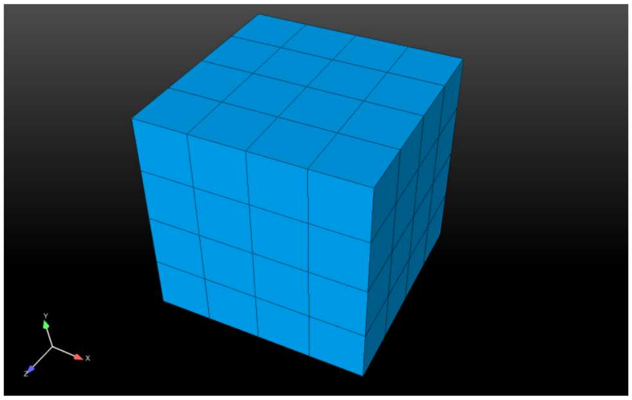


Figure 8.15: Force Inversion Test Geometry

The model consists of 64 Hex-8 elements, arranged in a 4×4×4 cube, and was specified as air for its material. One boundary condition constrained the model: one side was assigned with an absorbing boundary, creating a non-reflecting condition. Acoustic loading was applied to the opposite side, and consisted of one forcing function. Sides with boundary (green) and loading (orange) conditions are shown in Figure 8.16.

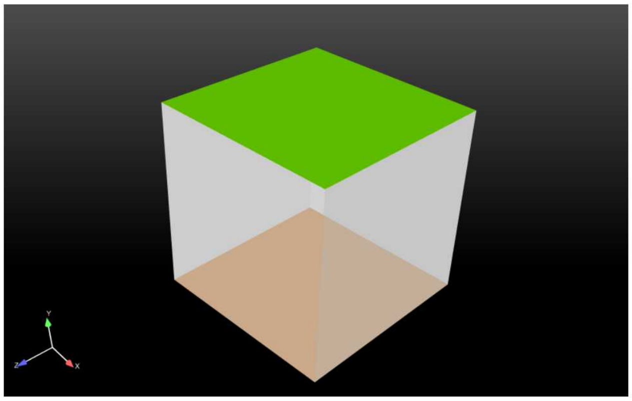


Figure 8.16: Side of model with applied acoustic loading represented in blue

Synthetic input data was generated by performing a forward problem on the model, using a forcing function with a known amplitude of 1 and frequency of 1 Hz. The data generated represented acoustic pressures measured for element nodes. The data was then used in the inverse problem to verify that the code could recover the original time history of the acoustic loading. The test, which used a full-Newton method with analytic Hessians, was analyzed by comparing the obtained acoustic loading time-history with the original loading function. Figure 8.17 compares the inverse-problem results with the original function:

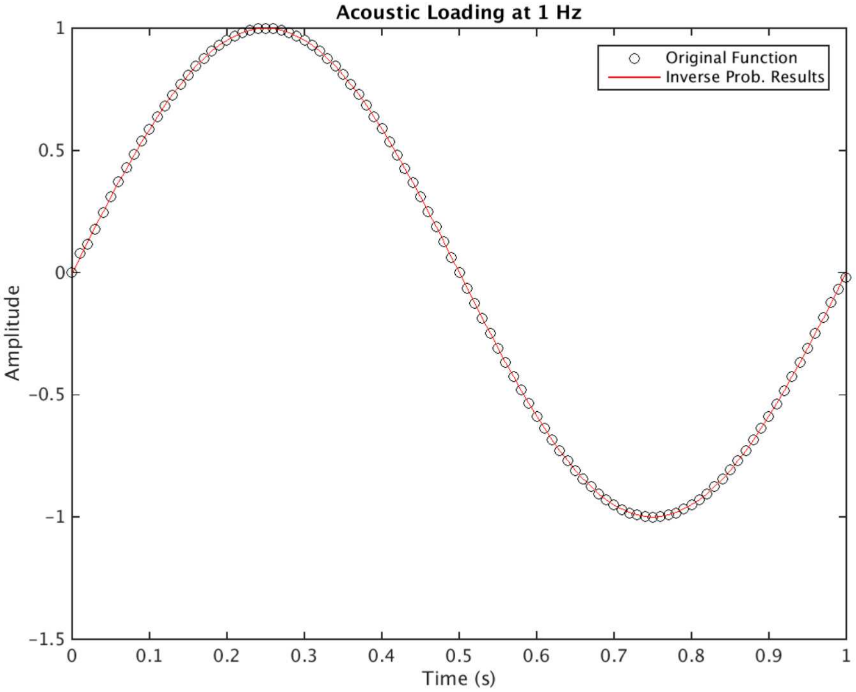


Figure 8.17: Transient Acoustic Pressure Identification

As shown in Figure 8.17, the inverse problem results exactly matched the original forcing function. The optimization of the objective function and gradient, run in both serial and parallel, performed four iterations and achieved appreciable convergence in both runs. Figure 8.18 shows the convergence history for the serial run.

For input deck see Appendix A.62.

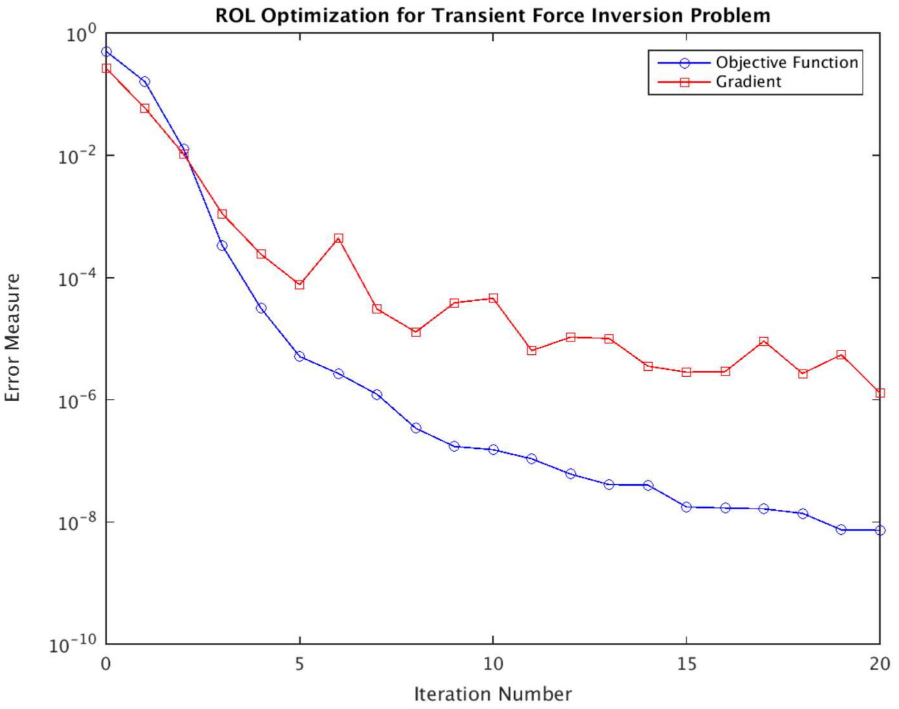


Figure 8.18: ROL Optimization of Objective Function and Gradient

8.6 Force Identification using Modal Transient

The temporal force identification capability for modal transient is demonstrated using synthetic displacement data generated from a 1 Hertz traction field, using the direct transient forward solution method. The cubic geometry of this test is shown in Figure 8.19.

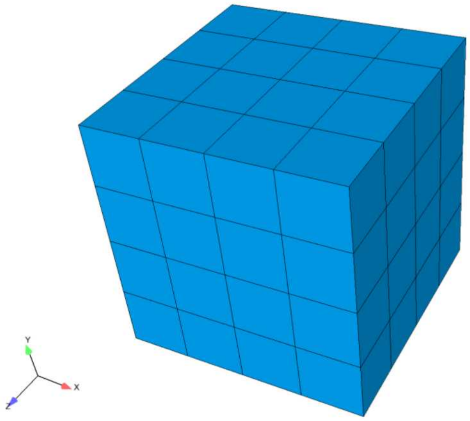


Figure 8.19: Force Inversion Test Geometry

The model consists of 64 Hex-8 elements, arranged in a 4×4×4 cube , and was assigned as a solid material. The model was constrained by one boundary condition, one side assigned as a fixed boundary. A traction load, including a normal component and two orthogonal shear components, acted on the side opposite of the fixed side and had a direction of $(x = 1, y = 2, z = 3)$. Figure 8.20 shows the sides with boundary (yellow) and loading (pink) conditions.

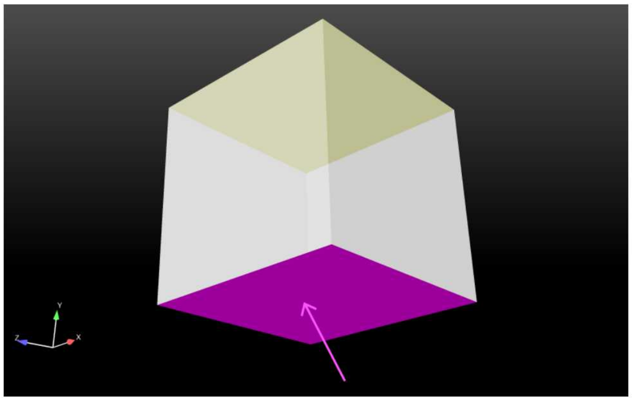


Figure 8.20: Fixed boundary (yellow) and traction loaded (pink) sides in model

Synthetic input data was generated by performing a forward problem (direct transient) on the model, using a periodic traction load with a known magnitude of 1 and frequency of 1 Hz. The data generated representes displacements measured for element nodes. The data is then used in the modal-transient inverse problem to verify that the code can recover the original time history of the traction load. The test, which used a full Newton method with analytic Hessians, was analyzed by comparing the obtained traction load time-history with the original loading function. Figure 8.21 compares the inverse-problem results with the original function, with increasing numbers of modes. Note that as the number of modes increase the ability for the modal transient solution to match the direct transient solution increases, and the magnitude of the objective function drops.

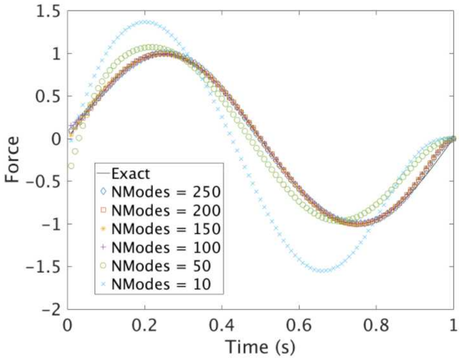


Figure 8.21: Inverse-problem results for traction loading time-history

Figure 8.22 shows the convergence of the final objective function with increased number of modes. Note that for this case, the objective function does not converge to zero, as not all of the modes can be calculated, so some truncation error exists in the problem. One possible solution is to use the modal transient optimization solution as the initial guess for a direct transient run. Figure 8.22 shows the convergence for the serial run. Note that the Tikhonov Parameter is used to maintain stability in the early time period.

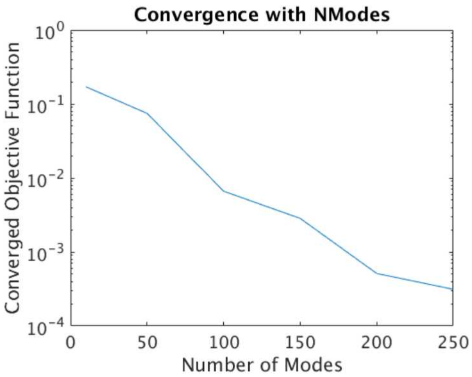


Figure 8.22: Convergence of Final Objective Function with Increasing Number of Modes

For input deck see Appendix A.63.

Chapter 9

High Cycle Fatigue and Damage

9.1 Random Vibration Moments

Stress moments are output from random vibration analyses for use in high cycle fatigue calculations. This is frequently useful as a faster alternative to calculating damage to a structure during a transient analysis.

9.1.0.1 Modalranvib Method

In the current modalranvib method, a model is given an input PSD and modal superposition is used in conjunction with the noSVD methodology in Salinas to give the 0^{th}, 2^{nd},and 4^{th} moments of Von Mises stress defined as:

$$M_i = \int \omega^i S_{\sigma\sigma} \mathrm{d}\omega$$

Where:

M_i	i^{th} moment of stress
ω	Frequency
$S_{\sigma\sigma}$	Stress PSD

Note: this is not the function used to calculate these values in Salinas. Instead, the noSVD method uses the equations outlined in reference.[14]

9.1.0.2 Transfer Function Method

The transfer function method directly calculates the stress PSD for each element using the equation below, then integrates the PSD across the frequency band.

$$S_{\sigma\sigma} = [\Psi][H_r(\omega)][\Phi]^T[S_{ff}][\Phi][H_r(\omega)]^T[\Psi]^T$$

Where:

Ψ	Stress modes
Φ	Eigen Vector
$H_r(\omega)$	Transfer Function
S_{ff}	Acceleration PSD

9.1.1 The Model

The model is shown in Figure 9.1. The green cutout represents the specific subset of elements that are being compared between the two previously described methods. The model is excited via a 3-axis random vibration signal (provided as a PSD to salinas) applied at a conmass. The conmass is attached to the rest of the model by rigid elements on the outer surface of the cylinder.

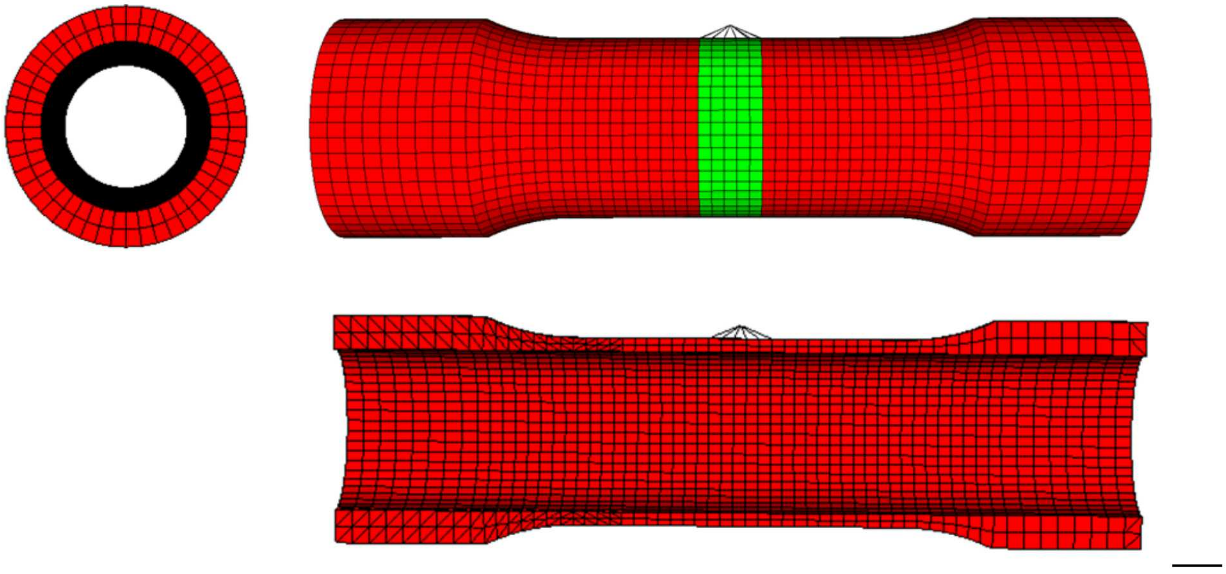


Figure 9.1: Model Geometry

For reference, the magnitudes of the first and second moment from Salinas are shown on the mesh in figure 9.2.

9.1.2 Acceleration PSD Comparison

Figure 9.3 illustrates acceleration power spectral responses at the input location (nodeset 10), and output location (nodeset 20). The input PSD is compared with the acceleration specified on the input. Because the inertial mass is only about 100 times the mass of the structure, resonances in the structure are seen in the driving point. Increasing the mass would reduce that resonance, but at the cost of decreased solution accuracy overall.

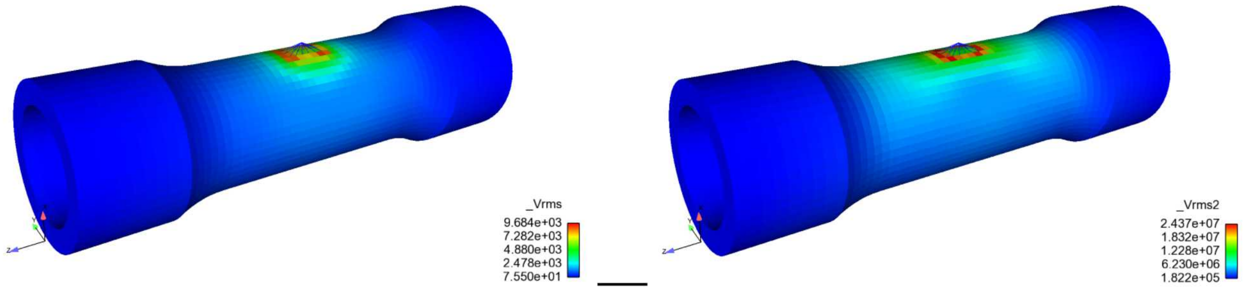


Figure 9.2: Salinas Output Colormap

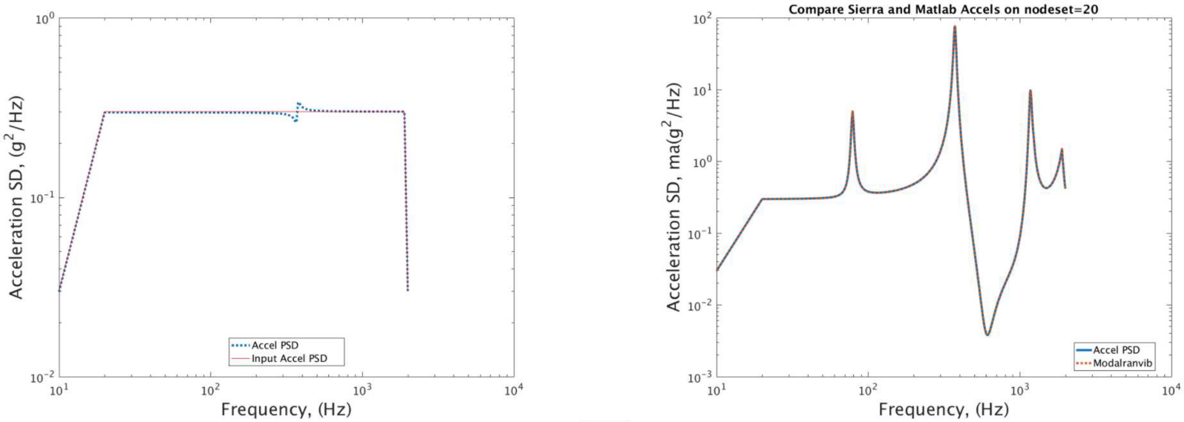


Figure 9.3: Acceleration PSD Comparison. The graph on the left is the driving point PSD output, and is nearly identical to the input specification. The graph on the right shows the acceleration on structural nodeset 20, which is located along the outer surface of the model, at the base of the fillet, and in-plane with the section view of figure 9.1.

9.1.3 Zero Moment Comparison

The RMS of the stress, which is defined as $\sqrt{M_0}$, is output from both calculation methods, and compared in Figure 9.4. Note that the error remains below 2x10^{-5}%.

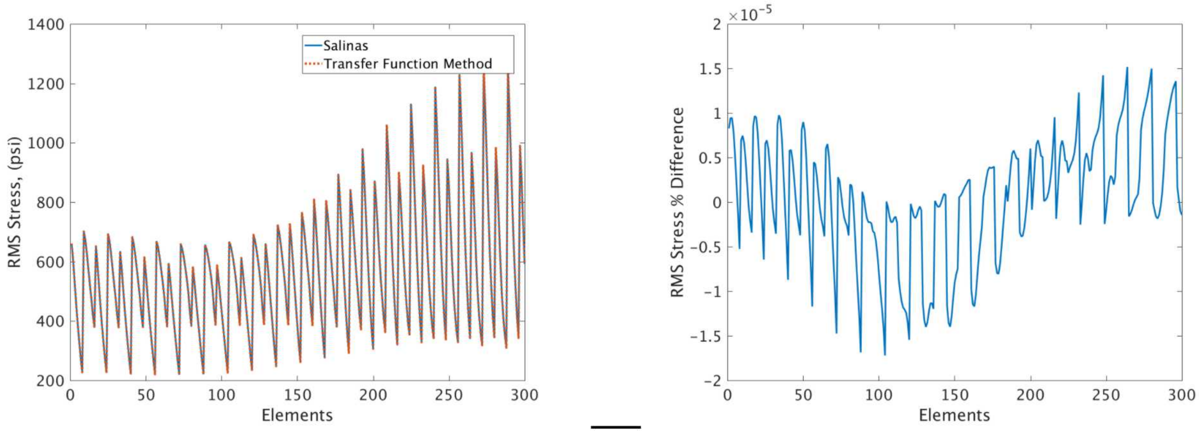


Figure 9.4: Stress RMS ($\sqrt{M_0}$) Comparison. The figure on the left shows the value of RMS stress across the elements in question, and the figure in the right shows the percent difference in the RMS stress between the two calculation methods. X-axis of both plots is the element number within the selected region.

9.1.4 Second Moment Comparison

The second moment of stress is output by Salinas as "VRMS2", which is defined as the square root of M_2. The data comparison between the two calculation methods is shown in figure 9.5. Note that the inclusion of the ω^2 term in the integration has exacerbated the error by several orders of magnitude, and is now on the order of 2%.

9.1.5 Fourth Moment Comparison

As with M_2, $\sqrt{M_4}$ is given by "VRMS4", which is plotted in Figure 9.6. Note that the error has increased with each successive moment calculation, and is now on the order of 2-10%. It is worth noting that not all damage methods use M_4, but it is not uncommon.

9.1.6 Zero-Crossing Rate Comparison

The zero-crossing rate is a common metric of stress cycling frequency in the frequency domain, and is calculated by $\nu_0 = \sqrt{\frac{M_2}{M_0}}$, or $\nu_0 = \frac{VRMS2}{VRMS0}$. As this value can be directly related to the damage calculated in an analysis, the error was examined and plotted in figure 9.7.

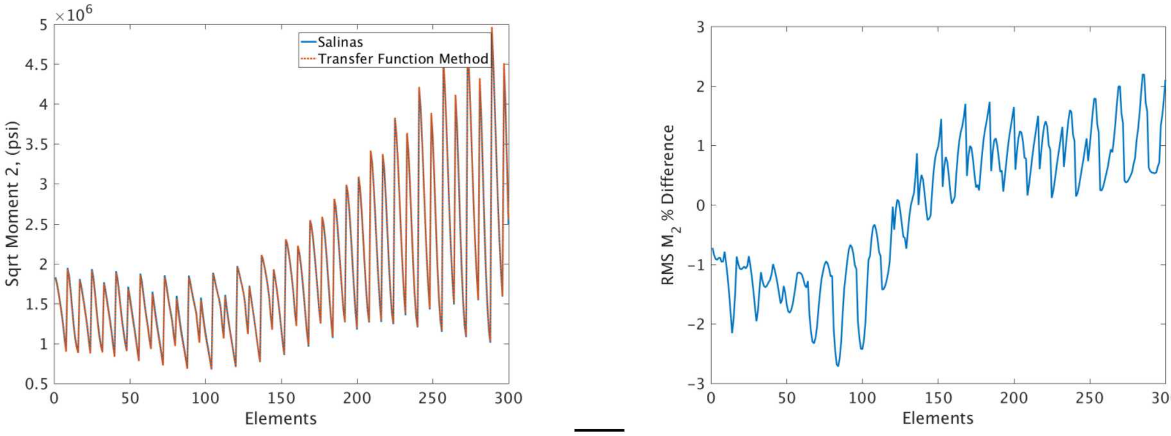


Figure 9.5: Stress RMS_2 ($\sqrt{M_2}$) Comparison. The figure on the left shows the value of VRMS2 stress across the elements in question, and the figure in the right shows the percent difference in the VRMS2 stress between the two calculation methods. X-axis of both plots is the element ID number within the selected region.

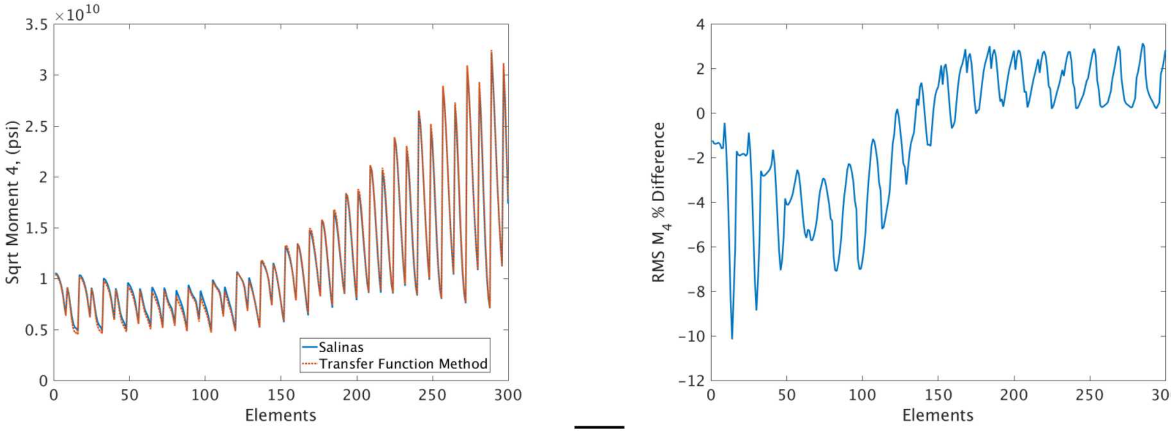


Figure 9.6: Stress RMS_4 ($\sqrt{M_4}$) Comparison. The figure on the left shows the value of VRMS4 stress across the elements in question, and the figure in the right shows the percent difference in the VRMS4 stress between the two calculation methods. X-axis of both plots is the element ID number within the selected region.

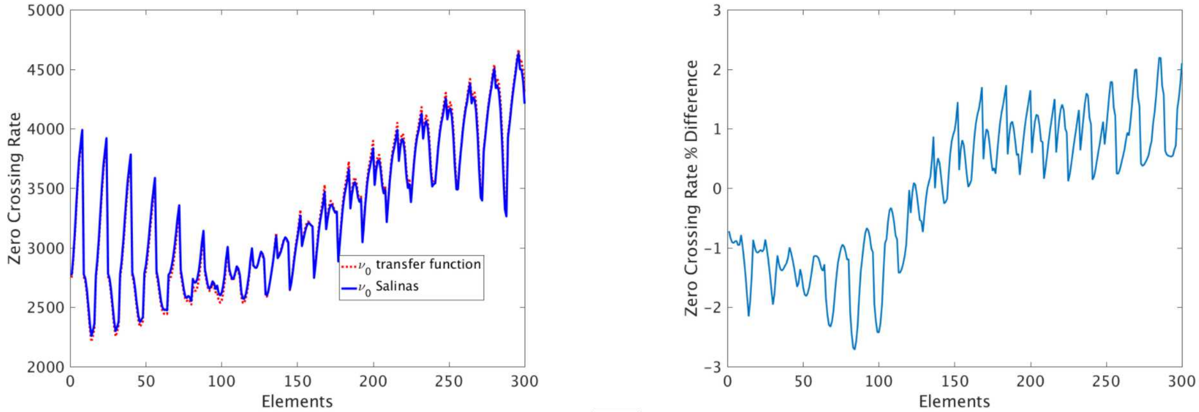


Figure 9.7: Zero Crossing Rate Comparison. The figure on the left shows the value of zero-crossing rates across the elements in question, and the figure in the right shows the percent difference in the zero-crossing rates between the two calculation methods. X-axis of both plots is the element ID number within the selected region.

For input deck see Appendix A.64.

9.2 Fatigue Output of Single DOF in Random Vibration

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, mutipoint 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^{TM} calculation found in "byhand.m". Each result is listed in following paragraphs.

9.2.1 Ensure Normalization of Eigenvectors

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`.

9.2.2 Determine the modal transfer functions, H_i

The physical force, F, is transformed to modal space by premultipling by ϕ^T. The modal transfer function at frequency ω describes the contribution of one mode to the resulting displacement.

$$\begin{aligned} u &= \sum_i^{Nmodes} H_i(\omega) F_i(\omega) && (9.1) \\ \text{where} & && (9.2) \\ H_i(\omega) &= \frac{1}{\omega^2 - \omega_i^2} && (9.3) \end{aligned}$$

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, u_i, is given by $u_i = H_i F_i \approx \phi^T F/\omega_i^2$ ≈6.2121e-9. The modal amplitude for FRF is not directly output, but the physical amplitude is output.

9.2.3 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, $H \approx$ 7.4286E-8. Likewise, the amplitude is the transfer function multiplied by the force. $U(\omega) = H(\omega) * Force \approx \phi F \phi^T/\omega_i^2 \approx$ 6.6857E-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 the top end of frequency band, $\ddot{U}(100) = 0.26394$.

Salinas computes 0.0026394 and 0.26394.

9.2.4 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. A similar relation exists for acceleration, but the integration includes ω^4, i.e.

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

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

$$\begin{aligned}
\Gamma &\approx 2\pi \sum_f H_q^2 S \Delta f w_i && (9.4)\\
&\approx 2\pi H_q^2 9.0[5\,10\,10\,10\,10\,10\,10\,10\,10\,5] && (9.5)\\
&\approx 2\pi(2.7249\text{e-}8)^2(9.0)(90) && (9.6)\\
&\approx 3.7789\text{e-}12 && (9.7)
\end{aligned}$$

Likewise

$$\begin{aligned}
\Gamma_A &\approx (2\pi)^5 \sum_f H_q^2 S f^4 \Delta f w_i && (9.8)\\
&\approx (2\pi)^5 H_q^2 9.0[10\,20\,30\,40\,50\,60\,70\,80\,90\,100]^4[5\,10\,10\,10\,10\,10\,10\,10\,10\,5] &&\\
&\approx (2\pi)^5(2.7249\text{e-}8)^2(9.0)2.0332\text{e}9 &&\\
&\approx 0.13306 && (9.9)
\end{aligned}$$

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 $(A_{rms}/X_{rms})^2 = 3.5212\text{E}10$.

9.2.5 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.

$$\begin{aligned}
\Gamma_V &\approx (2\pi)^4 \sum_f H_q^2 S f^2 \Delta f w_i && (9.10)\\
&\approx (2\pi)^3 H_q^2 9.0\,[10\,20\,30\,40\,50\,60\,70\,80\,90\,100]^2[5\,10\,10\,10\,10\,10\,10\,10\,10\,5] &&\\
&\approx 5.5447\text{E-}7 && (9.11)
\end{aligned}$$

Salinas has $VRMS = 1.1384\text{E}2$ and $VRMS2 = 4.3607\text{E}4$. Therefor $(VRMS2/VRMS)^2 = 1.4673\text{E}5$, 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`.

9.2.6 Fatigue Solution

We verify the fatigue analysis on a single, 1x1x1 Hex8 element. 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.

9.2.6.1 Assumptions

We begin the solution with a previously verified random vibration solution with results in Table 9.1.

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

Table 9.1: Input Moments

We also *construct* a fictitious material with fatigue parameters (i.e. S-N curves) that make computation simple. The S-N curve is represented in Figure 9.8. 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 9.2.

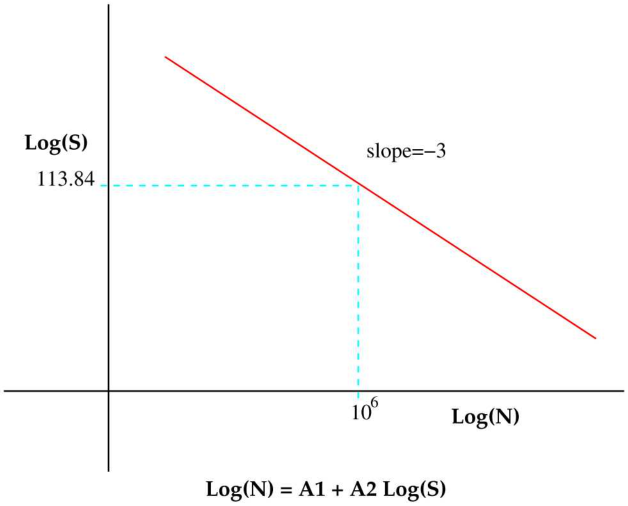


Figure 9.8: S-N Curve for Fictitious Material

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

Table 9.2: Fatigue Material Parameters

9.2.6.2 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.

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

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 A.65.

9.2.7 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.

9.2.7.1 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 × 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.

9.2.7.2 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	✓

9.3 Fatigue Output of Dogbone Test

9.3.1 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.

9.3.2 Methodology:

The dog-bone specimen described by Anes et al.[15] 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.

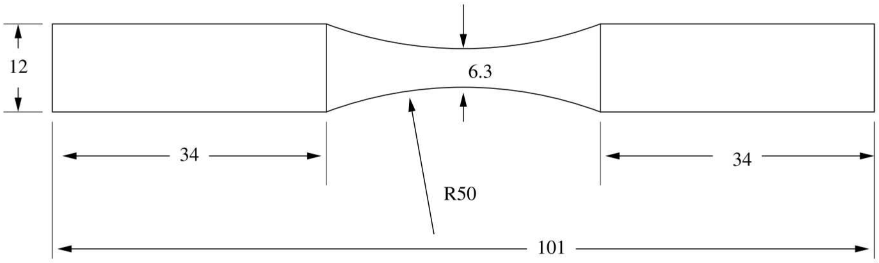


Figure 9.9: Dog-bone Specimen Dimensions (mm)

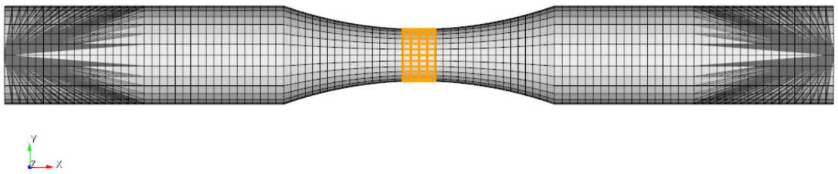


Figure 9.10: Boundary conditions of mesh

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 9.11, with details in Tables 9.3 through 9.5. 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.

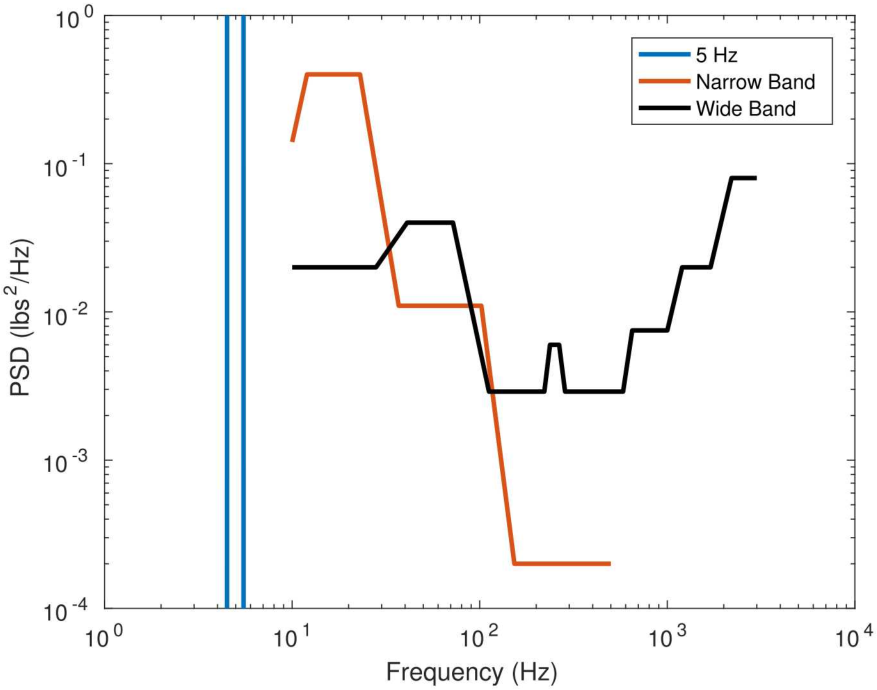


Figure 9.11: Power Spectral Density of Input Force

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.

Table 9.6 shows the preliminary results of the 5 Hz test of frequency and time domains. Sierra/SD and Siesta results are very close for this model. However, it is important to note

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 9.3: 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 9.4: 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 9.5: 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 9.6: Preliminary 5 Hz Results. Ranges indicate spatial changes.

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 in increased in length, 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 9.7. 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 9.7: 5 Hz test after adjustments. Ranges indicate spatial variation.

9.3.2.1 Narrowband and Wide-Band Evaluation

Tables 9.8 and 9.9 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 9.8: Narrow-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 9.10 compares these results. Overall, the comparison is good, and well within the differences of the other methods.

9.3.2.2 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.

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 9.9: Wide-Band Test Results. Ranges indicate spatial variation.

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 9.10: Maximum of Siesta and Sierra/SD Computions

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 9.12 provides some information on the convergence of the solution as these parameters are varied. The figure on the left shows variation of the narrowband damage, D_{NB}, as damping is increased. 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 affect 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 A.66.

9.4 Fatigue Output of Pinned Shell

Both a narrowband and wideband example are evaluated. The verification test ensures the following.

- 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 inpt. This is easier to evaluate for narrowband processes.

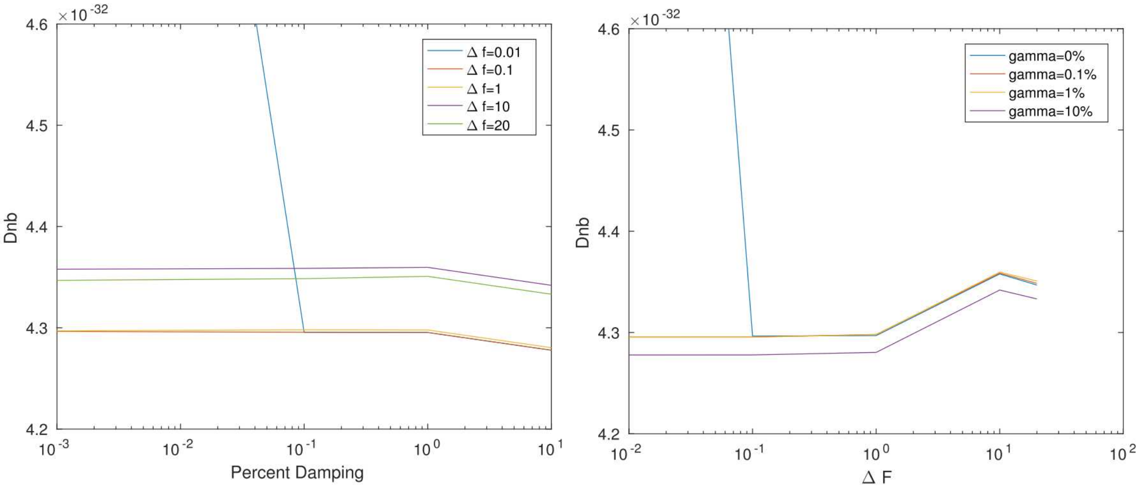


Figure 9.12: Convergence of PSD Integration

- 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 rainfail calculations. We also have no convergence study, either with mesh, or with modes.

9.4.1 Narrow Band Pinned Plate

The model is a simple rectangular plate, shown in Figure 9.13. The plate is 10 units in

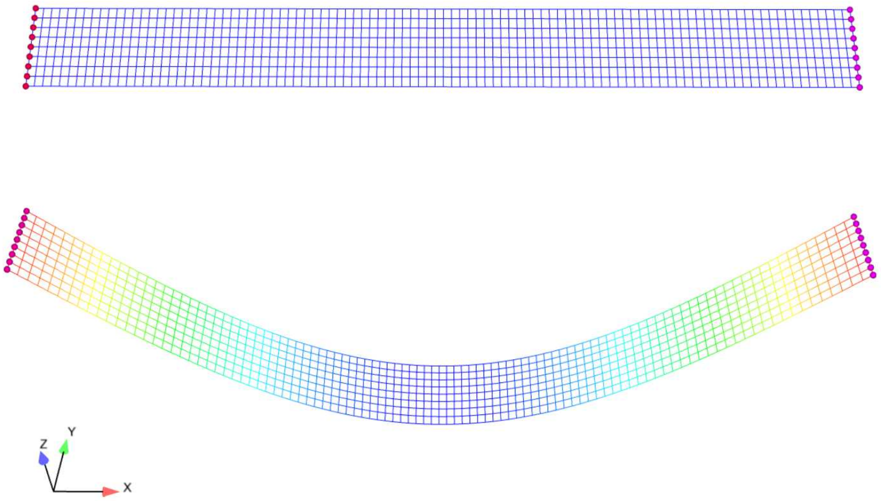


Figure 9.13: Pinned Plate Geometry, and First Mode

X, 1 unit in 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 9.13. The +/- X surfaces are pinned, with no other Dirichlet boundary conditions. Loading is a uniform pressure in the $-Z$ direction. The narrowband loading in shown in Figure 9.14, 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.

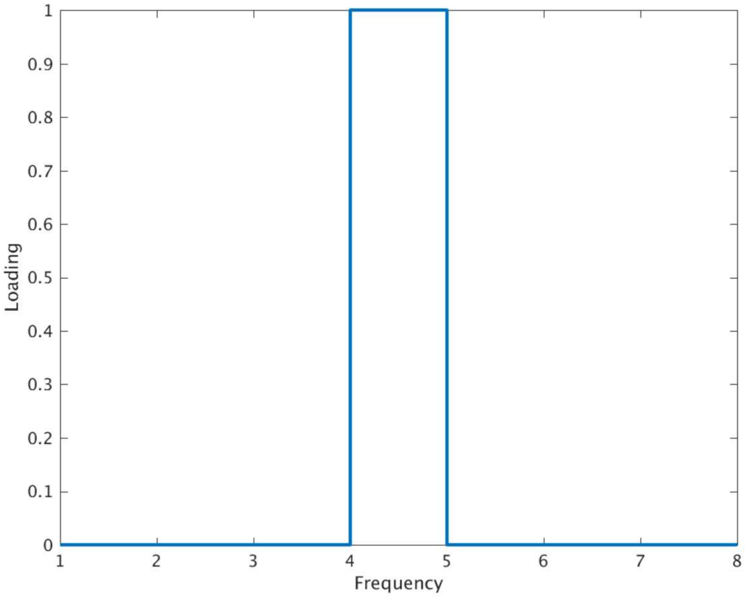


Figure 9.14: Pinned Plate. Random Vibration Loading

9.4.1.1 Statics:

The static response on the bottom surface, to a uniform pressure load is shown in Figure 9.15. Stress on the top surface is the negative of this, and there is no stress on the midplane.

9.4.1.2 FRF:

The input is modified, and a modal FRF computed from 0.01 to 8 Hz, as shown in Figure 9.16. The stress response is very similar to the static solution, as evidenced in Figure 9.17. 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.

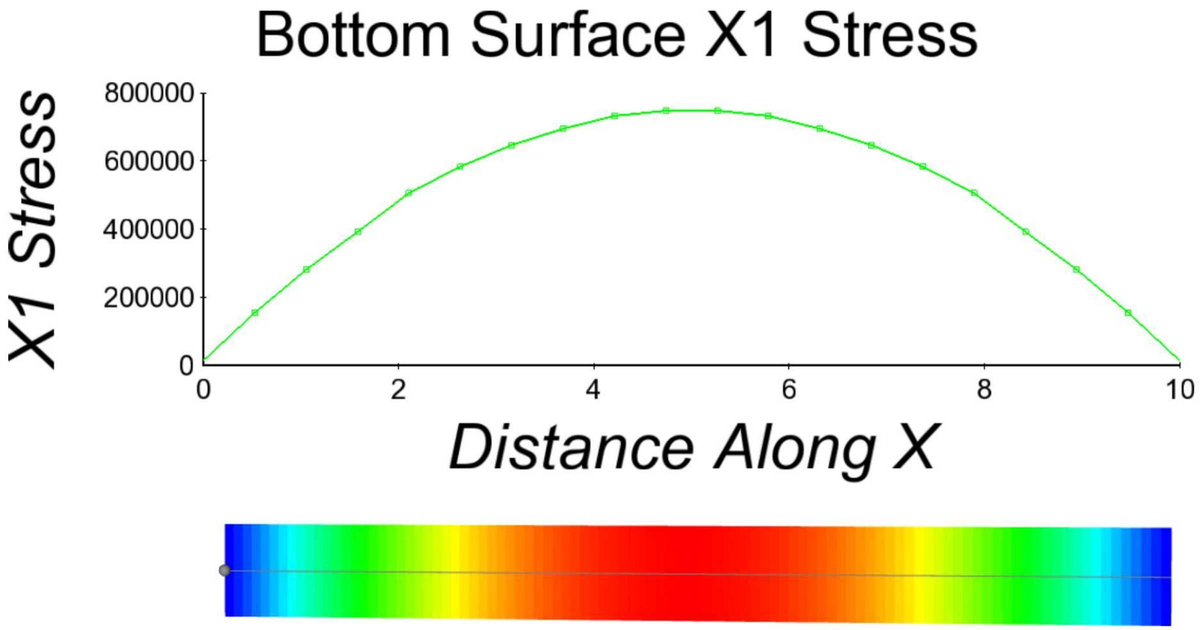


Figure 9.15: Pinned Plate. Statics Response

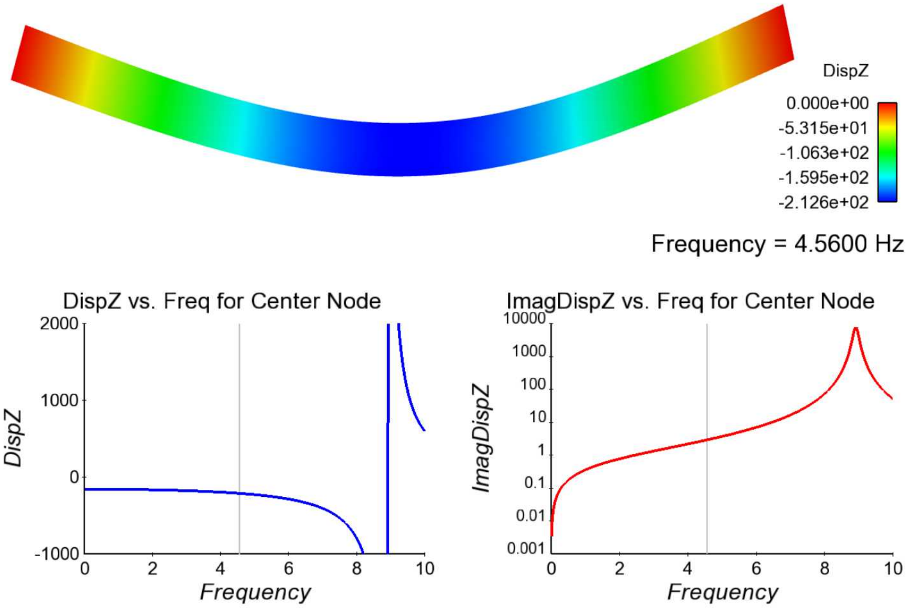


Figure 9.16: Pinned Plate. Modal FRF Response

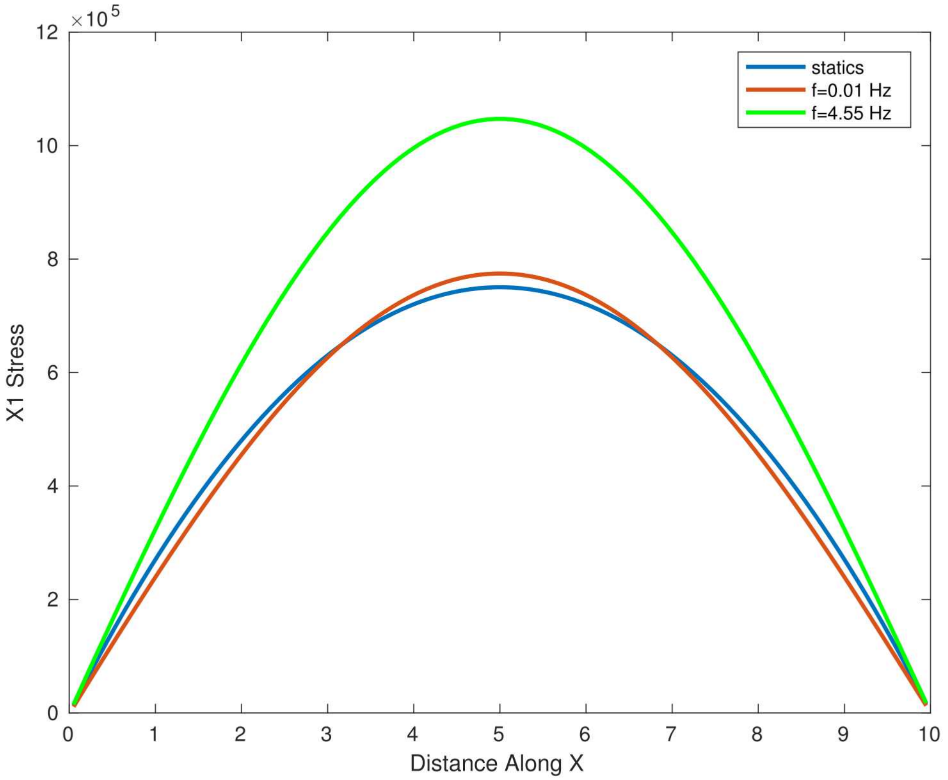


Figure 9.17: Pinned Plate. Comparison of Static and FRF Solutions

9.4.1.3 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.

9.4.1.4 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.

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

The value from the output is $D_{NB} = 1.291125933 \times 10^{-5}$.

9.4.2 Wideband Calculations

Wideband calculations use the same model as narrowband. Only a single mode is retained as shown in Figure 9.13, however the band selected is from 10-100 Hz. Figure 9.18 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.

✓ 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 of these methods is applied here for the element 51, which is found at the center of the plate where the stress is maximum.

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

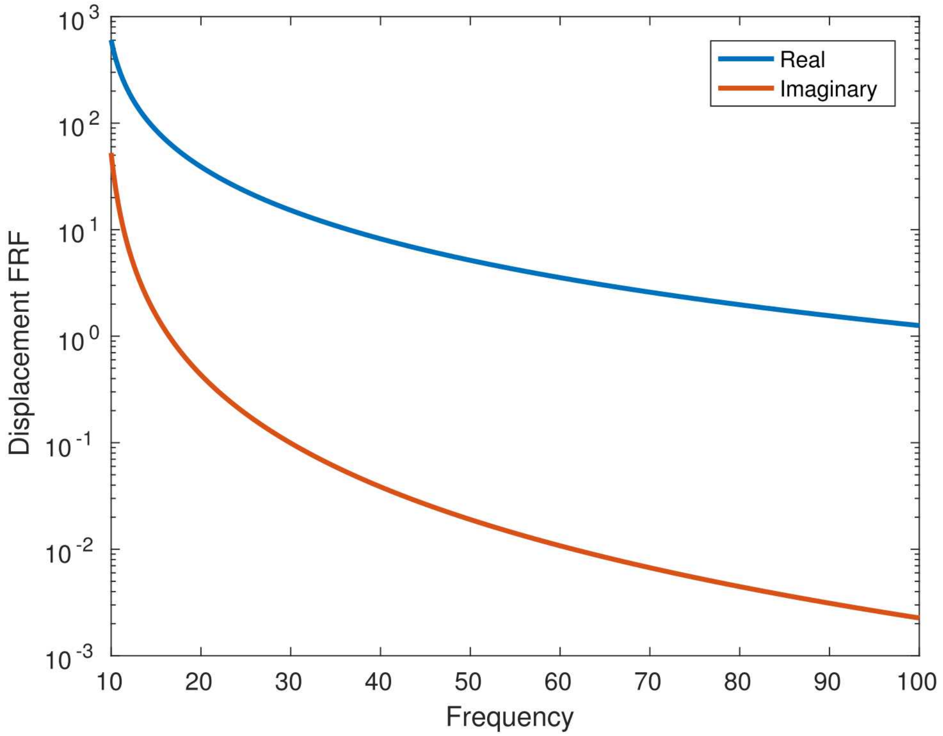


Figure 9.18: Pinned Plate. Wide Band FRF Response

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

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

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 9.19. 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.

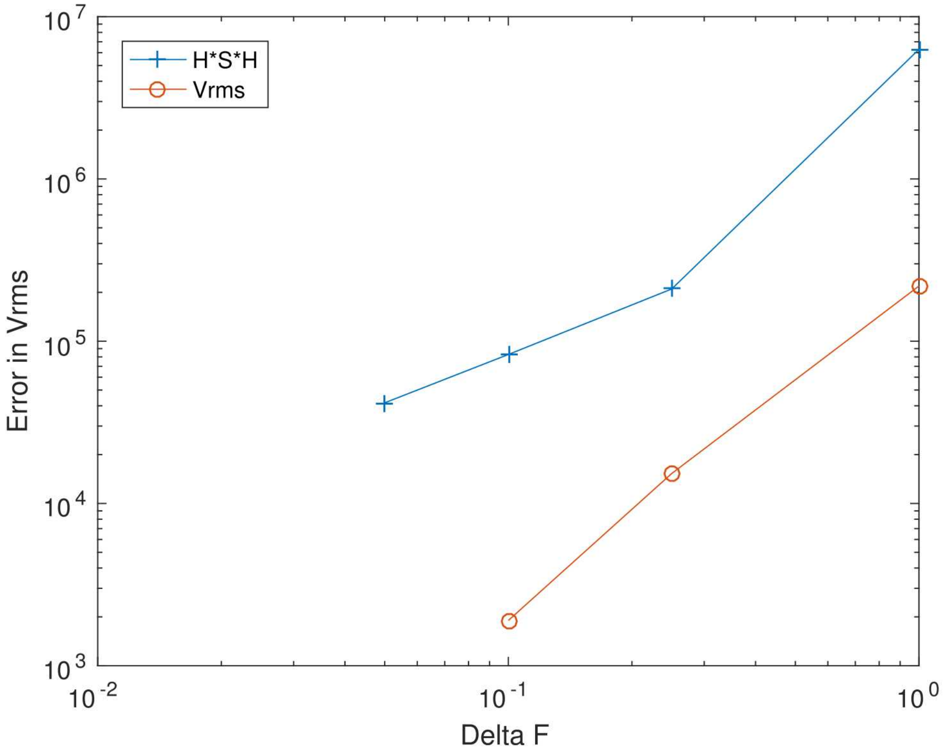


Figure 9.19: Convergence of Frequency Integrals

For input deck see Appendix A.67.

This page intentionally left blank.

Chapter 10

Legacy Sierra/SD Verification Problems

10.1 Element Verification Tests

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's nuclear weapons program.

The tests described in this document were performed in support of release 1.1 of **Sierra/SD** . This covers capabilities for linear structural dynamics, linear statics, and linear transient dynamics. Specifications and requirements for this release are identified in the Requirements document,[16] and summarized in a technical report.[17]

Verification tests can never cover the full aspects of the software. Analysis shows that there are simply too many paths through the software to ever adequately cover all such paths (see *Beizer*[18] or *Myers*[19]). 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 only indirectly.

10.1.1 Element Patch Tests

The element patch tests in this study are derived from MacNeal's monograph.[20] 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 of 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*.[20]

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 10.1.

Table 10.1: 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

10.1.2 Element 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* [20]). 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 10.2, 10.3 and 10.4 show results for rectangular, trapezoidal, and parallelogram shaped elements, respectively.

Table 10.5 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 $E = 10^7$, the Poisson ratio is 0.25. The tip load is of unit magnitude.

Table 10.6 shows results for a cantilever beam that twist a total of 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 10.2: 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 10.3: 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 10.7 through 10.10 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 has either a width-to-height aspect ratio of 1.0 or 5.0. The plate height is 4.0. The plate thickness is 0.01 for solid elements (Hex20, Hex8, Hex8b, et10, Tet4, and Wedge6) and 0.0001 for shell elements (QuadT, Tria6, Tria3, and TriaShell). The Young's Modulus is 1.7472x10^7 and the Poisson ratio 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 midside 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 not particularly well-suited for the Tet elements (Tet10 and Tet4) as the aspect ratios of the elements is very large due to the small thickness. Nastran's Tet10 performs in a similar fashion to **Sierra/SD** 's Tet10 on the remaining problems in this section.

Table 10.11 shows the results for the Scordelis-Lo Roof tests. This test involves a curved plate. The radius of curvature is 25.0 and the associated arc 80 degrees. The length of the

Table 10.4: 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 10.5: Curved Beam Tests

Element Type	**In Plane**	**Out of Plane**
Hex20	0.879	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	1.01	0.893
Tria3	1.07	0.864
TriaShell	0.185	0.895

plate is 50.0 and the thickness 0.25. The straight edges of the plate are free and the curved edges are constrained so that they are not able to move in the plane in which the curved edge is contained. The loading is a traction in the z-direction on the face of the plate of magnitude 90.0 per unit area. The 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 10.12 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 10.13) 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

Table 10.6: Twisted Beam Tests

Element Type	In Plane	Out of Plane
Hex20	1.01	0.995
Hex8	14.2	10.7
Hex8b	0.744	0.741
Tet10	1.00	1.01
Tet4	0.0947	0.162
Wedge6	0.0877	0.283
QuadT	31.5	25.0
Tria6	19.6	15.5
Tria3	30.9	24.6
TriaShell	11.4	8.99

Table 10.7: 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

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.

10.1.3 Element 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[21] is not directly applicable to unstructured meshes, related methods can

Table 10.8: 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 10.9: 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

be used to determine truncation error (see *Alvin*[22] for example). Some detail is provided in Appendix C.

Convergence testing is usually 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 usually considered sufficient. Convergence testing is performed as part of this verification suite to provide consistency with verification efforts in other Sandia codes.

In its simplist 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.

Table 10.10: 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 10.11: 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	0.563	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	0.0608	0.0847	0.0999	0.113	0.130
QuadT	1.58	1.05	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

The convergence tests for the Hex8 elements was the static deformation of a cantilevered beam. The meshes employed are shown in Figure 10.1 and the appropriate plot of convergence error is show in Figure 10.2. 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. That one needs to go to extremely small meshes to achieve this geometric convergence would appear to result from focusing on convergence at a single point – a very rigorous criterion.

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 10.3. Here we see that the convergence rate is -3.8, very close to the theoretical value.

The convergence test of the Tet10 element was a bit more difficult. Here it is impossible to refine the mesh through sectioning to create new elements all of approximately the same

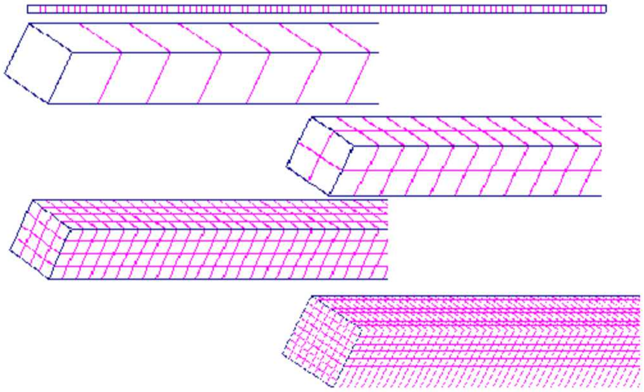


Figure 10.1: Meshes for convergence test for Hex8 elements

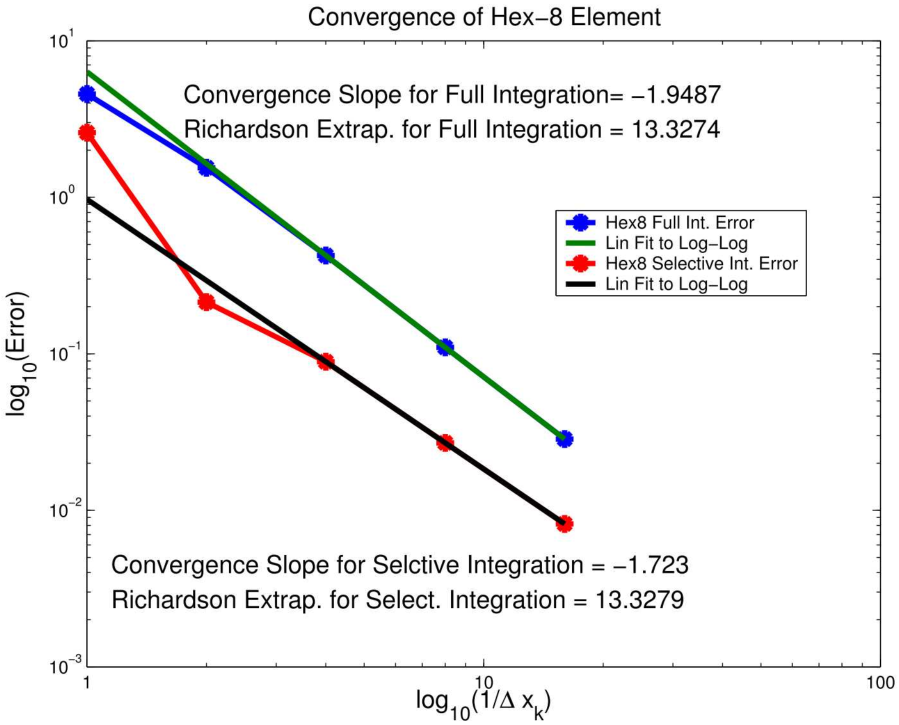


Figure 10.2: As the meshes are progressively refined, the slope of the log-log plot of the error approaches -2, as predicted by theory.

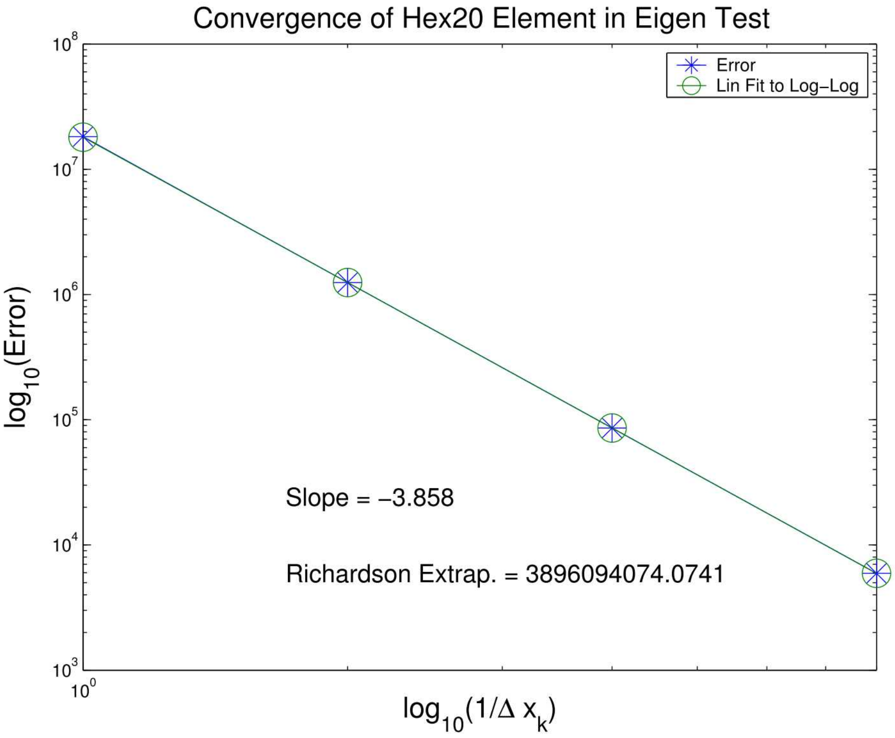


Figure 10.3: The convergence plot of the Hex20 element for the first eigen value shows a slope very close to the theoretical value of -4.

Table 10.12: 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 10.13: ThickWalled Cylinder Tests

Element Type	$\nu = .4900$	$\nu = .4990$	$\nu = .4999$
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

size and also retain the aspect ratios of the coarser mesh. Instead, it was necessary to entirely remesh the numerical beam each time a mesh of finer element size was required. Still the resulting slope of the log-log error plot (shown in Figure 10.4 is very close to (though a bit larger than) the theoretical value of -4.

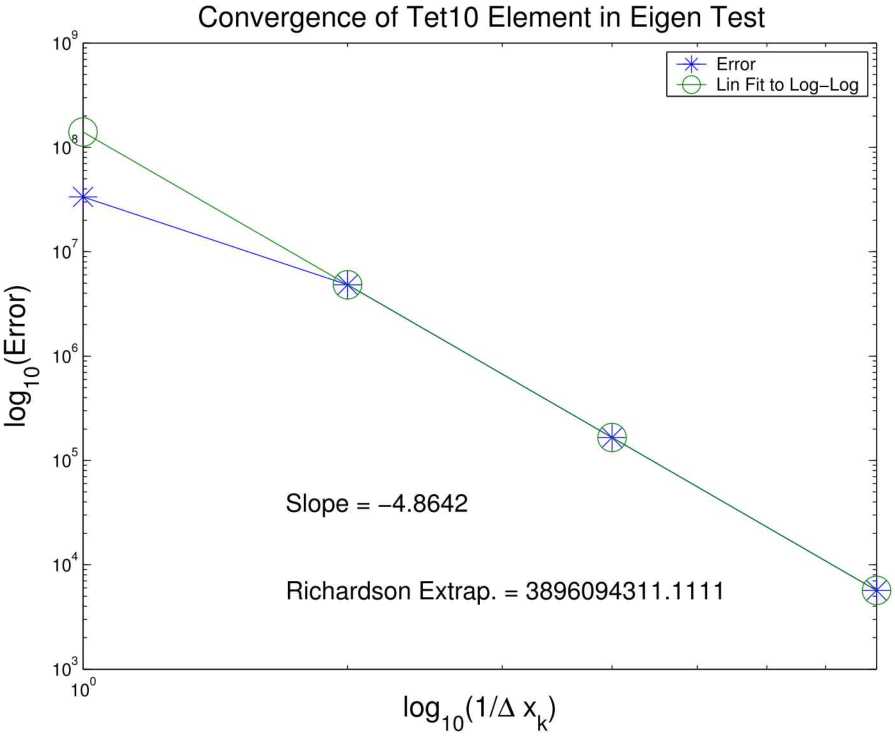


Figure 10.4: The convergence plot of the Tet10 element for the first eigen value shows a slope very close to the theoretical value of -4

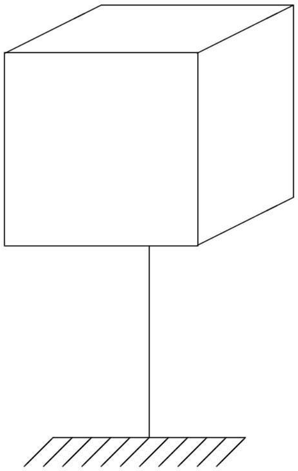


Figure 10.5: Box on a Bar test object

The convergence rates of the various elements are listed in Table 10.14.

Table 10.14: 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. Not truly higher order.
Beam2	2		
Tet4	2		
Hex20	4	4	using eigen analysis
Tet10	4	4	using eigen analysis
Tria6	2		derived from Tria3. Not truly higher order.

10.1.4 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 very 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 1. The model is named `BoxOnBarRbe3.inp`. The test is `Salinas_rtest/test_tool/fast_regression_tests/mpc/BoxOnBarRbe3.test`.

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 quite 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

10.1.5 Verification of hexshells

In this section we list the results of several verification examples for hexshell elements. These verification examples were taken from Professor Carlos Felippa, the developer of the element, (see reference 12). The goal here was to reproduce the results obtained in that report.

10.1.5.1 Example 1

This example corresponds to section 9.5 in the report 12, 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 23 and reference 24 as

$$\frac{\pi^2 - 8}{4\pi}\frac{PR^3}{EI} \tag{10.1}$$

We note that this solution is the total change in diameter for the ring.

For modeling purposes, we only model a quarter of the ring, and we apply appropriate boundary conditions on the symmetry planes. 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. Since we are only modeling a quarter ring, this result must be divided by 2. 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 24. 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 10.1 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 10.15.

Table 10.15: 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

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 results were the same as running a single ply with those same material properties, and so this provided a very weak verification of the multi-ply implementation.

10.1.5.2 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 only applied to a quarter section. The results are shown in Table 10.16.

Table 10.16: 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

10.1.5.3 Example III

The Scordelis-Lo Roof example. We note that, although in this example only a quarter of the roof is modeled, there is no need for dividing the answer by any multiple(as in previous examples) since the applied load is a gravity load rather than a point load. We note that the boundary conditions at the rigid diaphragms were incorrectly reported in Carlos's writeup. The correct ones are $u_x = u_z = 0$. With these conditions, the results as shown in Table 10.17 agree well with the expected values.

Table 10.17: 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

10.1.5.4 Example IV

This is the twisted beam model. The normalized results, compared with those of Carlos, are given in Table 10.18.

Table 10.18: 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

10.1.6 Verification of TriaShells for Composite Modeling

Laminate composites modeling in Sierra/SD is implemented by coupling Allman's triangle[25] with the DKT triangle.[26] Simply 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 27 and 28.

In the next sections we list the results of several verification examples for composite TriaShell elements.

10.1.6.1 Example 1

The first verification example is taken from Reference 27. A rectangular plate with dimensions 6" x 1" x 0.005" is modeled using 2 triangular elements (Figure 10.6). Looking at figure 10.6, 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.

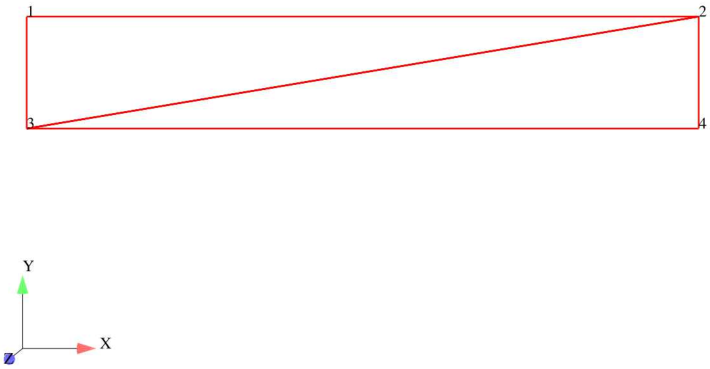


Figure 10.6: Two Element Test

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 10.7 thru 10.12 These figures show that the convergence of the Allman/DKT element is good. Both elements have similar convergence rates as the mesh is refined with the exception of the drilling degree of freedom. Figures 10.13 and 10.14 compare the x, y, z, θ_x, θ_y, and θ_z displacements at nodes 2 and 4 (see figure 10.6. Again, the Allman/DKT element compares very well with the STRI3 element as the mesh is refined. The only 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`".

10.1.6.2 Example II

The second verification example for laminate composite modeling is taken from Reference 29. A rectangular plate is subjected to a uniform pressure load of q = 0.003 psi. The plate, shown in figure 10.15 has dimensions 12 in. x 8 in. and is simply supported on each edge. The antisymmetric 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 29. 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.

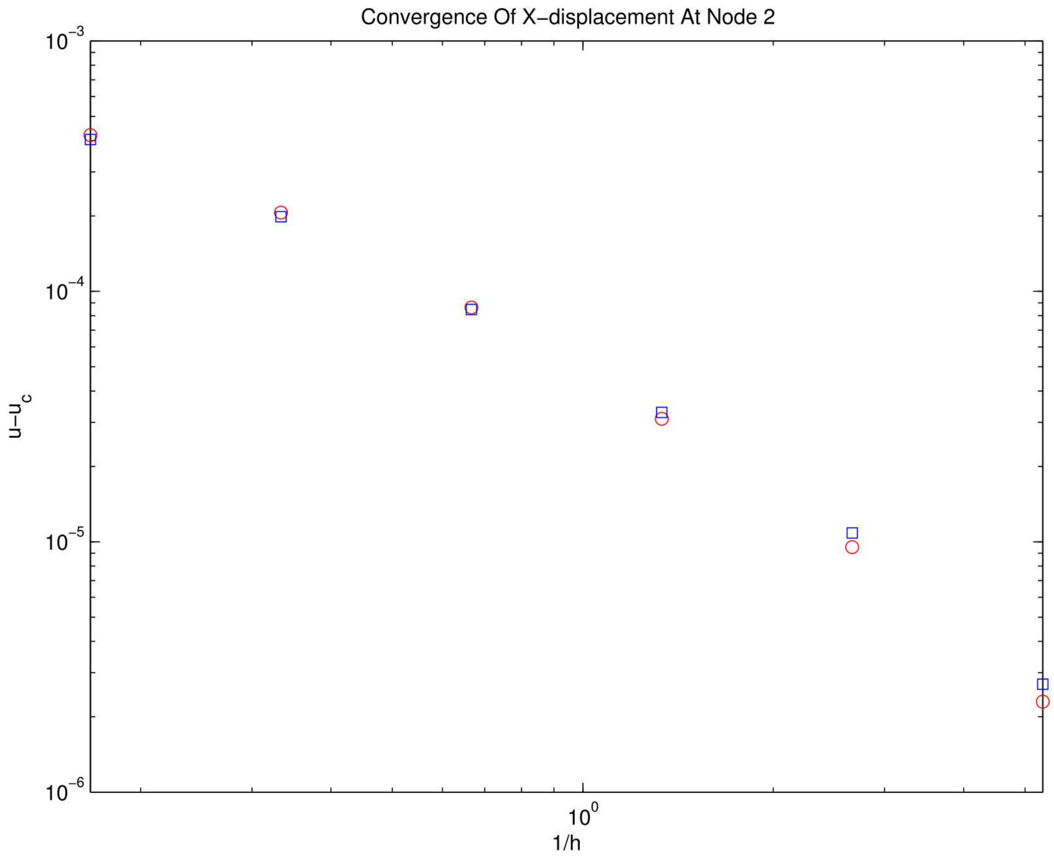


Figure 10.7: Comparison Of X-displacement Between Sierra/SD and ABAQUS

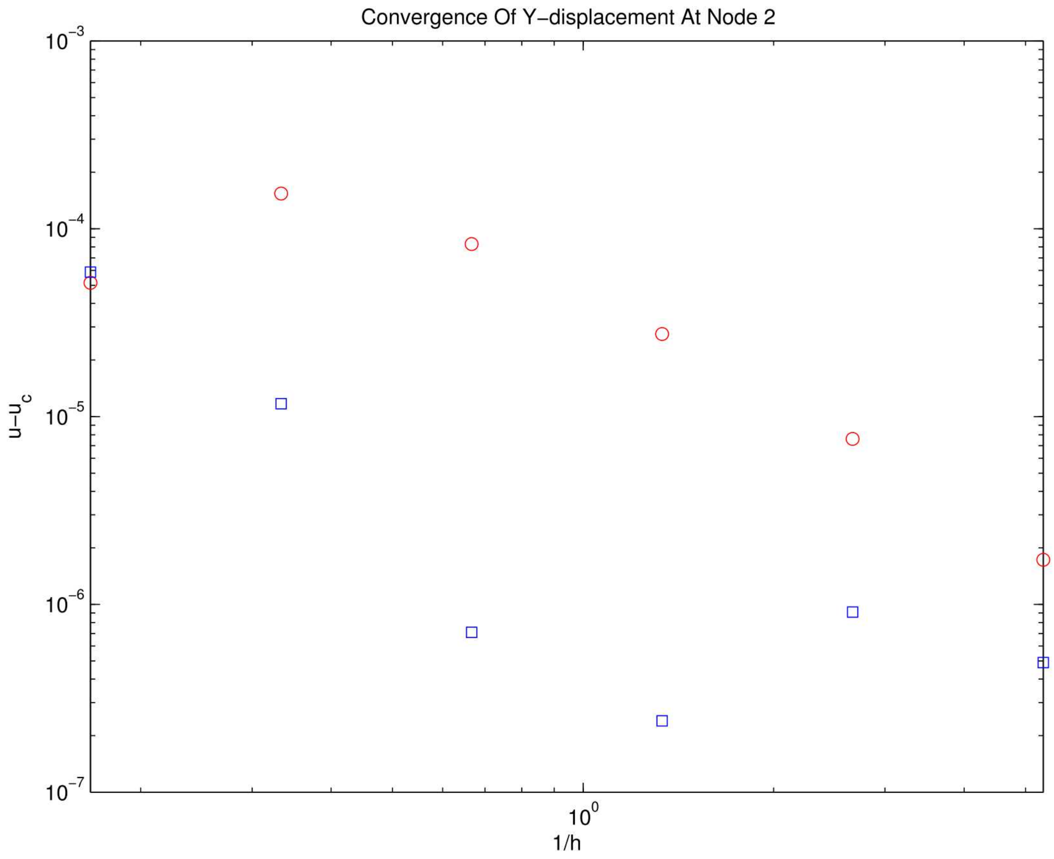


Figure 10.8: Comparison Of Y-displacement Between Sierra/SD And ABAQUS

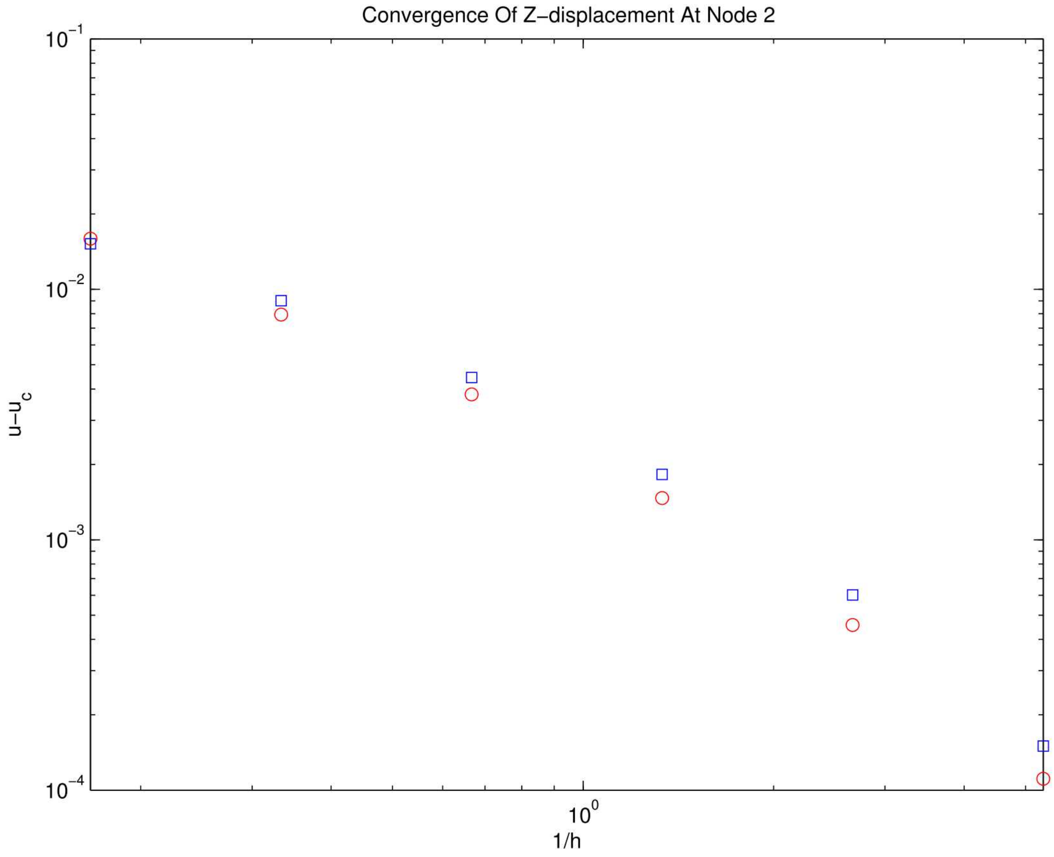


Figure 10.9: Comparison Of Z-displacement Between Sierra/SD And ABAQUS

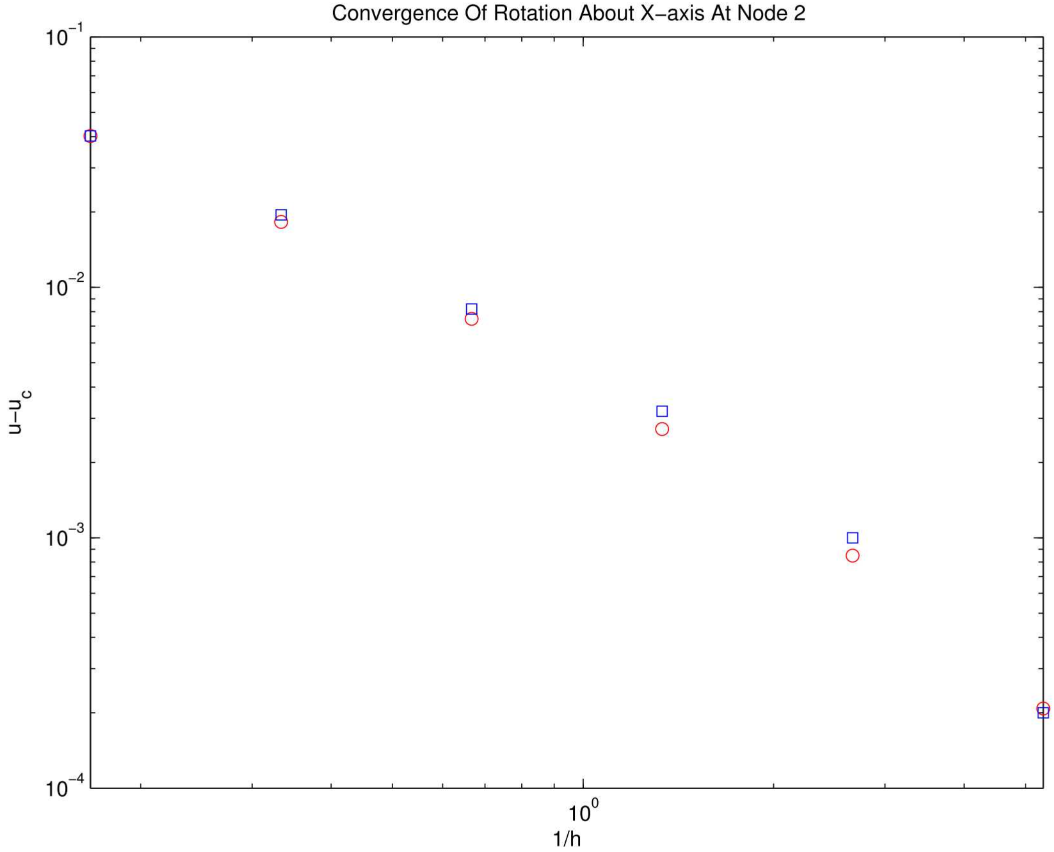


Figure 10.10: Comparison Of Rotation About X-axis Between Sierra/SD And ABAQUS

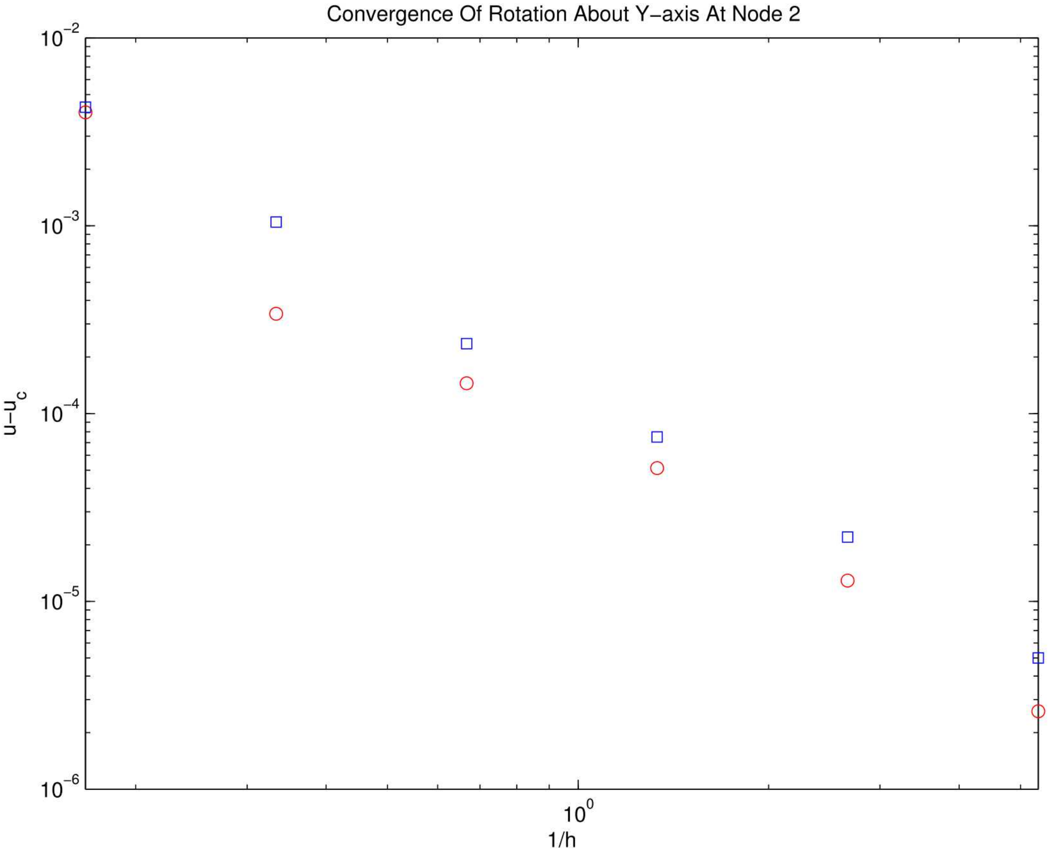


Figure 10.11: Comparison Of Rotation About Y-axis Between Sierra/SD And ABAQUS

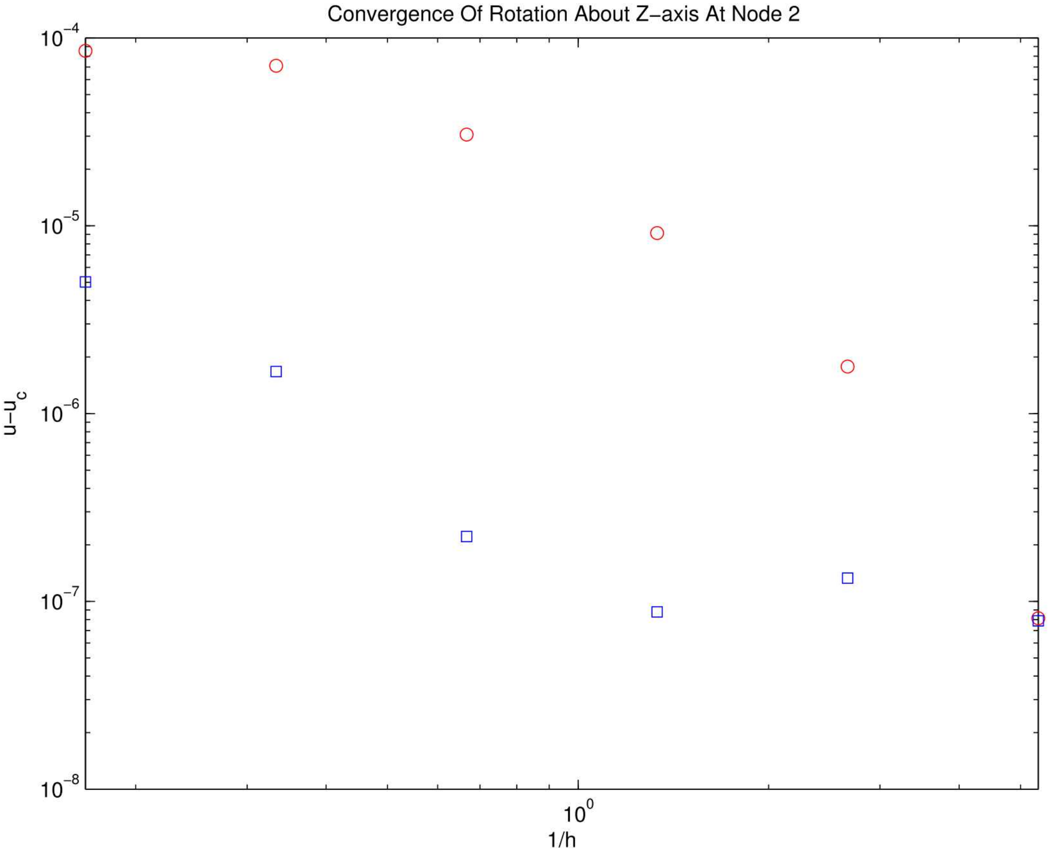


Figure 10.12: Comparison Of Rotation About Z-axis Between Sierra/SD And ABAQUS

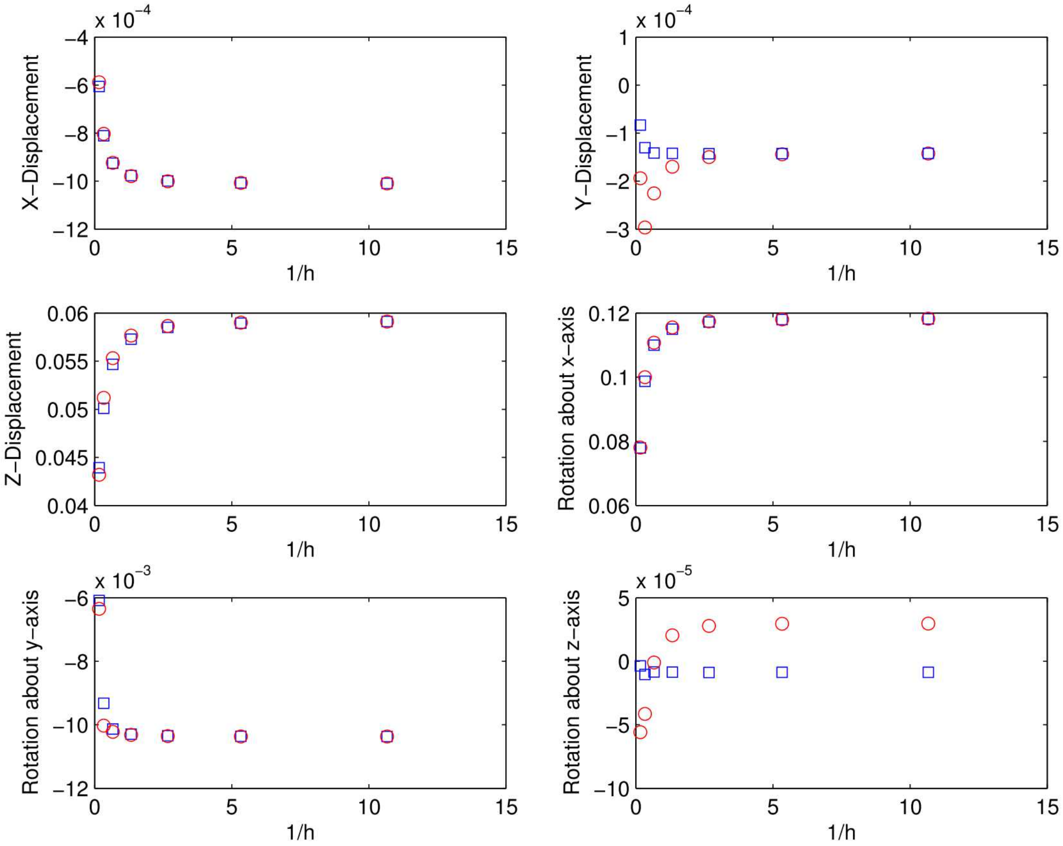


Figure 10.13: Convergence Of Displacements and Rotations At Node 2

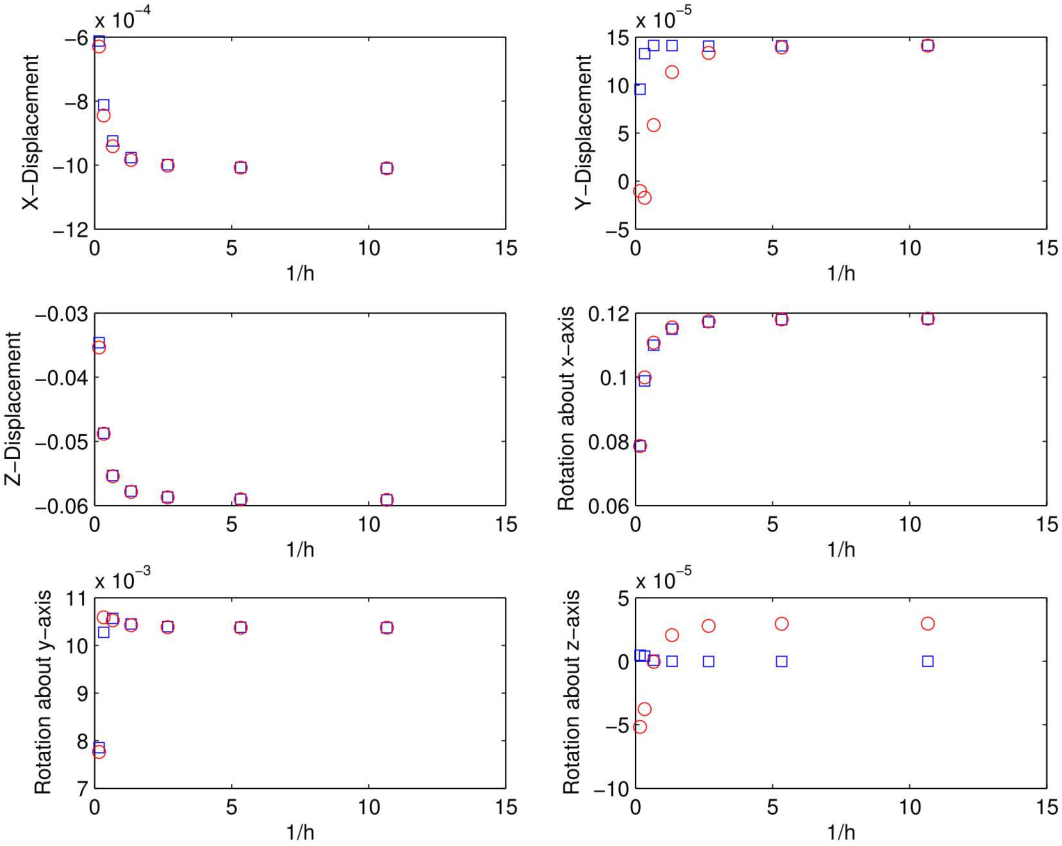


Figure 10.14: Convergence Of Displacements And Rotations At Node 4

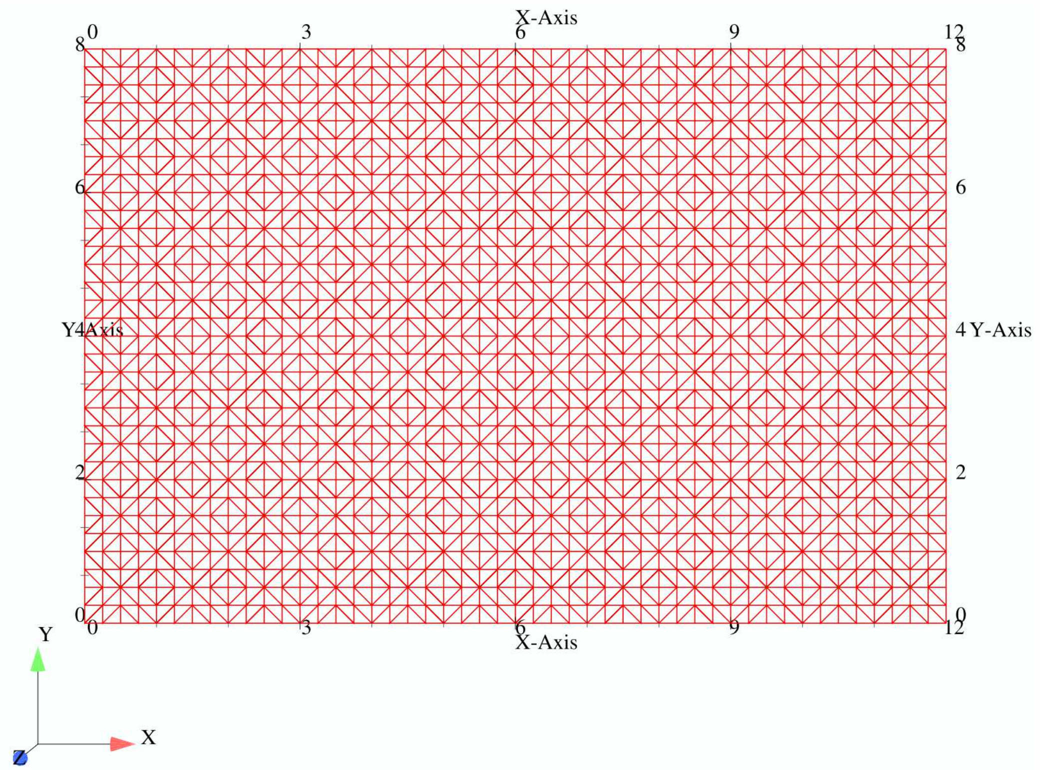


Figure 10.15: Finite Element Model Of A Flat Plate

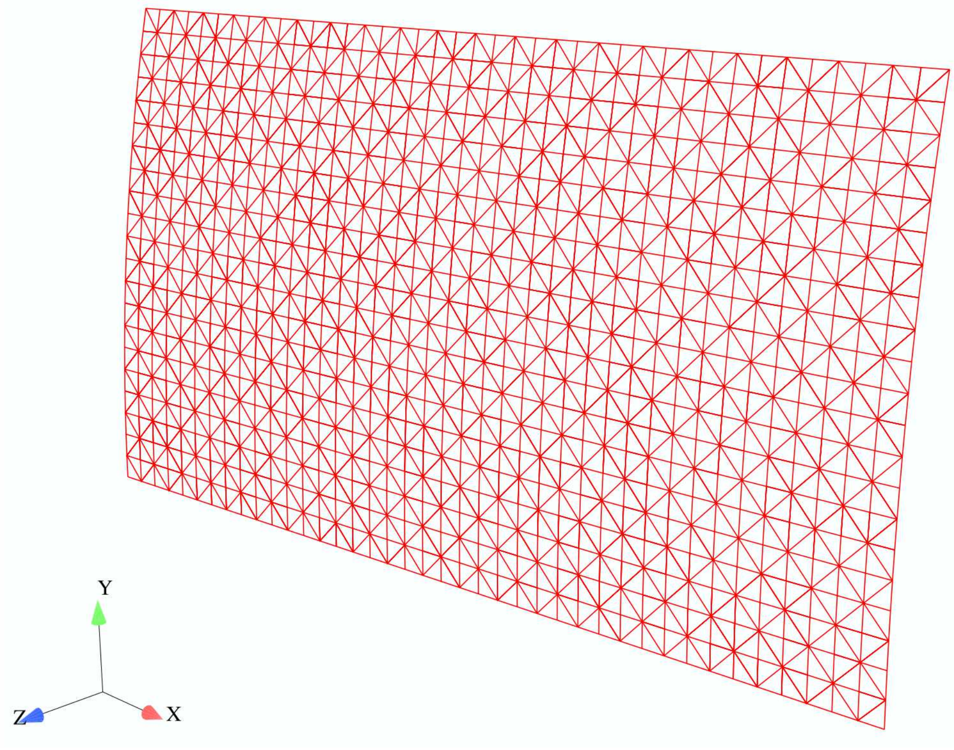


Figure 10.16: Finite Element Model Of A Cylindrical Panel

10.1.6.3 Example III

This verification example for laminate composite modeling is also taken from Reference 29. 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 10.16) 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 29. 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.

10.1.7 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. Both the Joint2g element and the Iwan constitutive model are thoroughly documented in the User's manual and Sandia 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 (presented in *SAND2002-3828*[30]) is,

$$D = r^{\chi+3} \frac{4F_s\phi_{max}(\chi+1)}{(\beta + \frac{\chi+1}{\chi+2}(\chi+2)(\chi+3)} \tag{10.2}$$

where β, χ, ϕ_{max}, and F_s are model parameters, and r satisfies

$$\frac{F_o}{F_s} = r\frac{(\beta+1) - r^{\chi+1}/(\chi+2)}{\beta + (\chi+1)/(\chi+2)}, \tag{10.3}$$

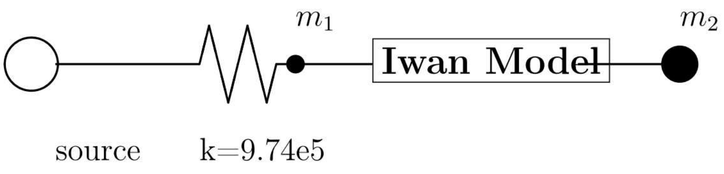
where F_o is the amplitude of the harmonic excitation. Comparison of the exact solution and **Sierra/SD** predictions is presented in Figure 10.17.

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 10.18 shows that desired accuracy is achieved with the number spring slider pairs predicted by theory.

10.1.7.1 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.

The problem looks like the following:



Here the source is a 100g wavelet base excitation, $m_1 = 0.05\,lb$, and $m_2 = 4\,lb$.

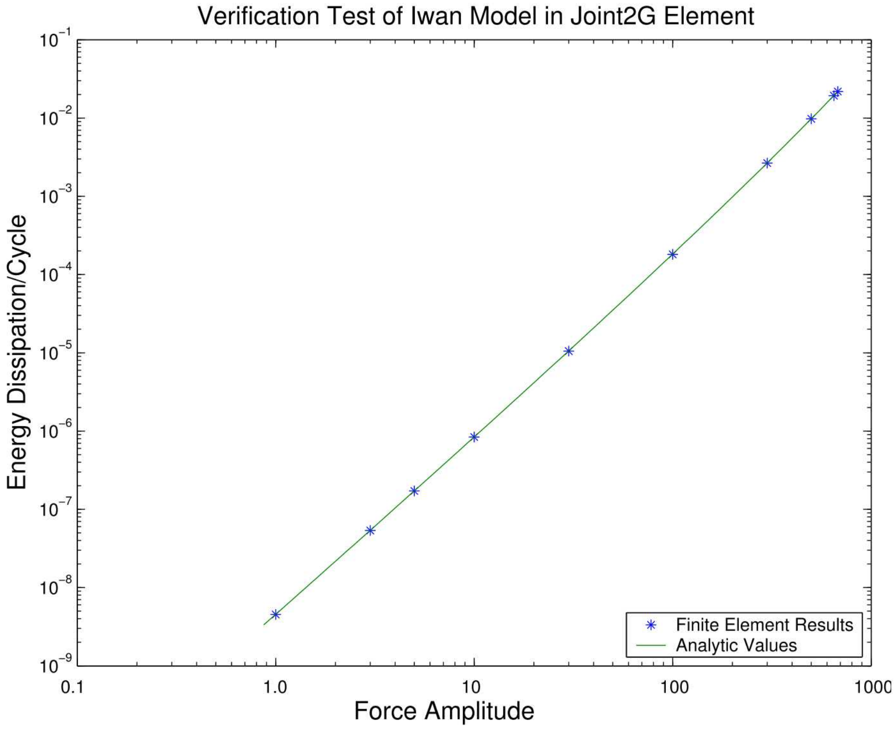


Figure 10.17: **Sierra/SD** Iwan Element: Comparison to Analytic Solution. The **Sierra/SD** predictions for unidirectional load on a simple joint agrees with the exact solutions.

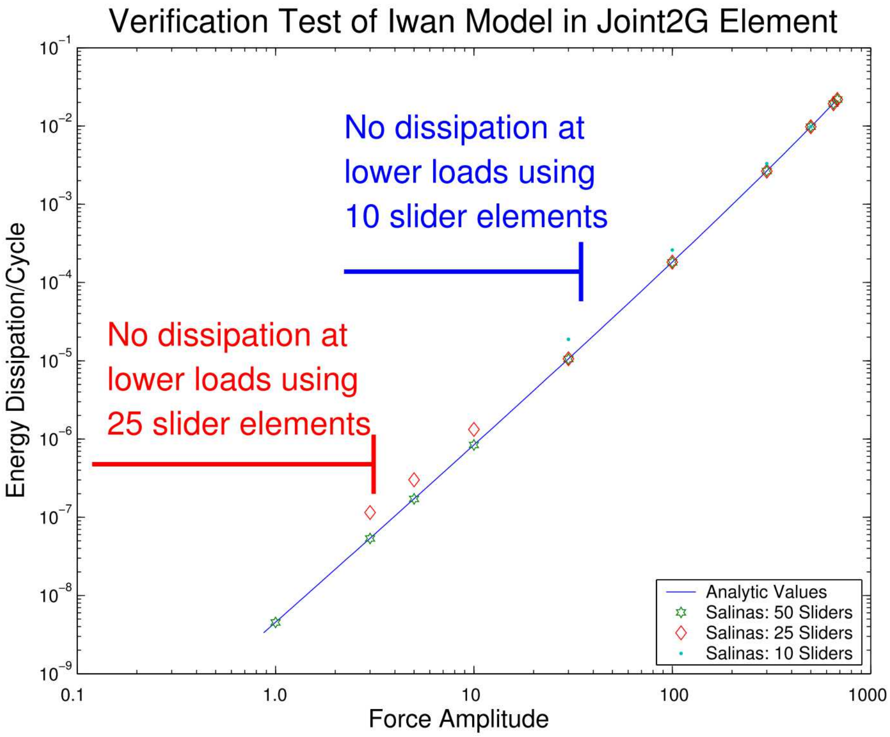


Figure 10.18: Significance of Number of Spring-Slider Pairs Used
The number of spring-slider pairs necessary to demonstrate sensitivity to given levels for load in **Sierra/SD** is that predicted by theory.

$$F_{min} \approx K_T \phi_{max} \frac{\alpha - 1}{\alpha^N - 1}$$

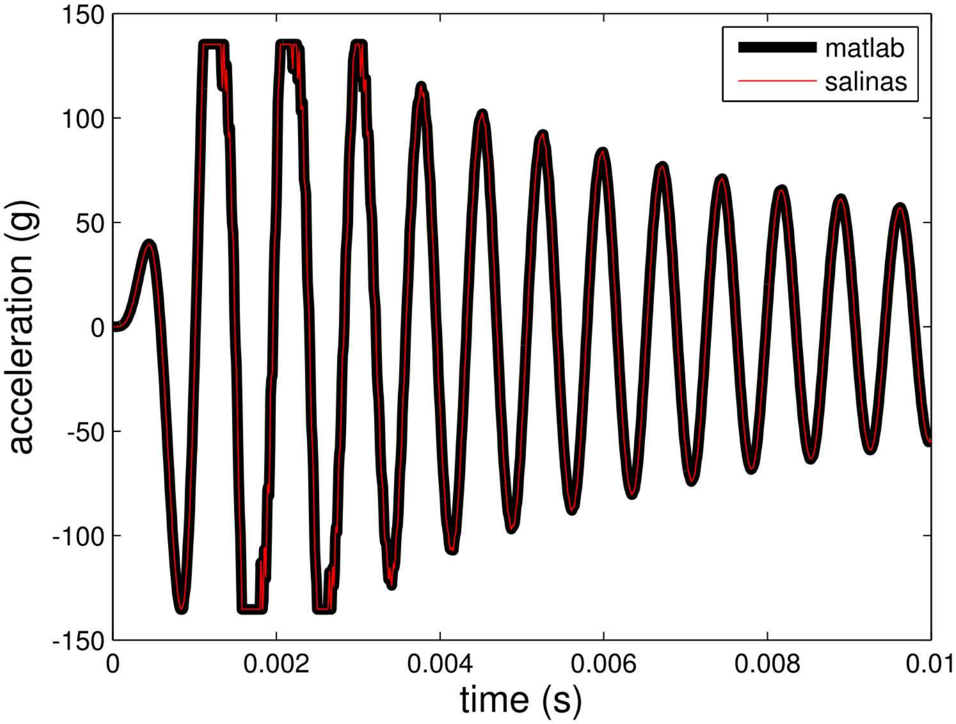


Figure 10.19: MATLAB and **Sierra/SD** calculation of M_2 acceleration.

Analysis is performed both within **Sierra/SD** and MATLAB . The acceleration of the four pound mass for each analysis method is compared in figure 10.19. We see very good agreement, though there seems to be slightly better resolution of the macro-slip in the Matlab result.

The stretch of the Iwan joint is another good indicator of agreement, and is shown in figure 10.20. 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 10.21. 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.

10.1.8 Verification of Membrane Elements

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

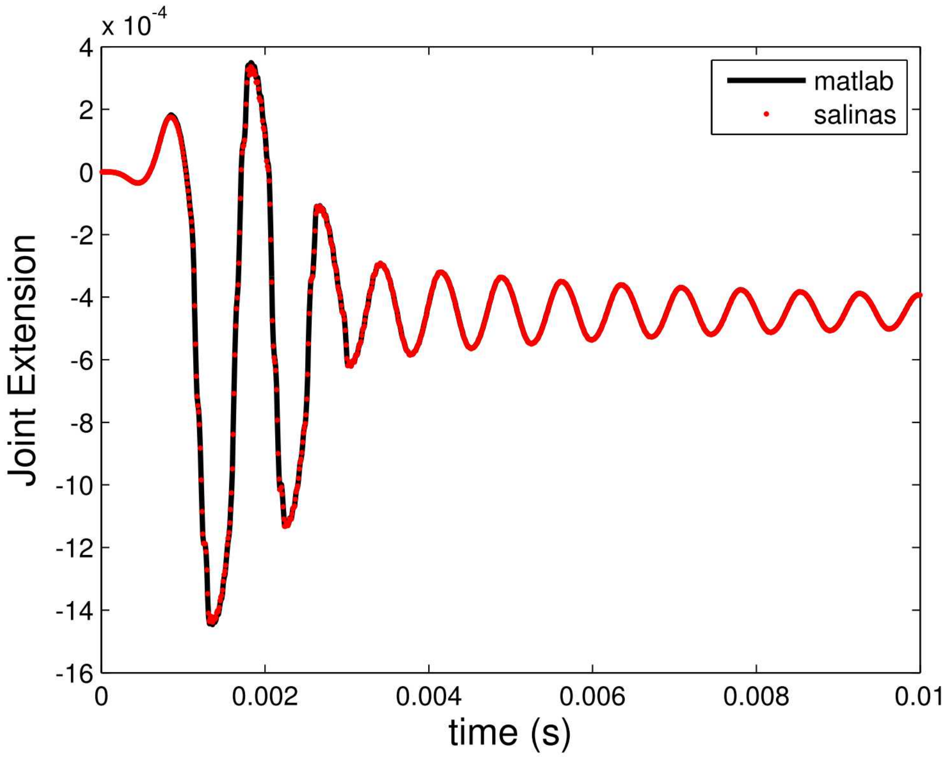


Figure 10.20: MATLAB and **Sierra/SD** calculation of joint extension.

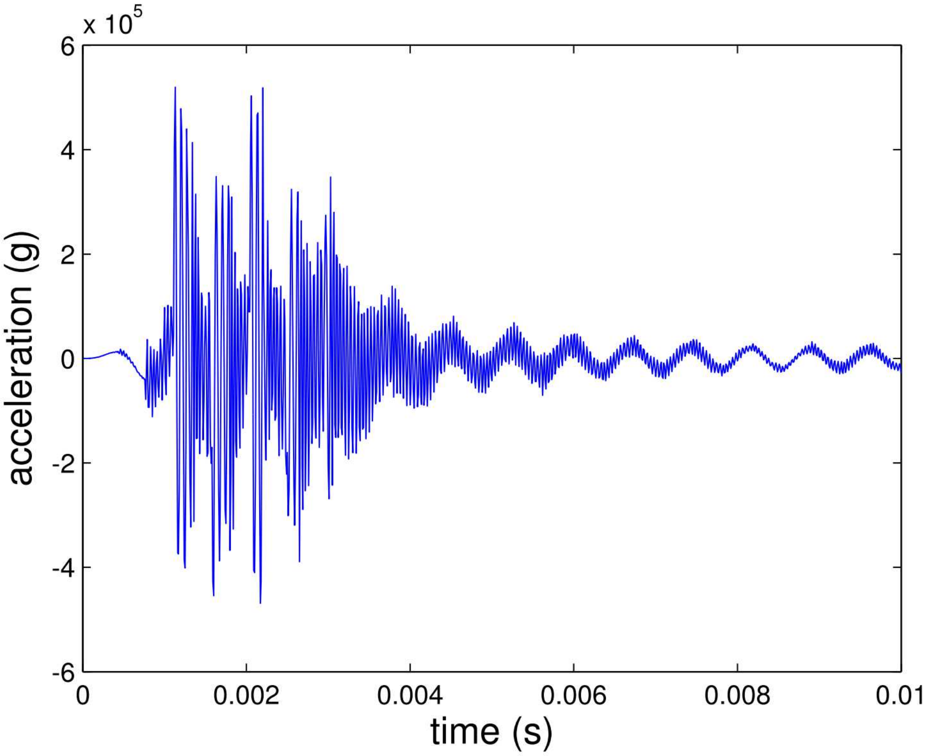


Figure 10.21: **Sierra/SD** calculation of M_1 acceleration.

exact (Hz)	computed
13.178	13.230
20.83	21.126
20.83	21.126

Table 10.19: Eigenvalue convergence for a fixed-fixed, prestressed membrane. The values given are the natural frequencies, in Hz.

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 eigenanalysis 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 eigenfrequencies for stretched square membranes are given in.[31] In the case of a membrane that is clamped along all boundaries, the frequencies are

$$f_{nm} = \frac{\omega_{nm}}{2\pi} = \frac{c}{2}\sqrt{\left(\frac{n}{L_x}\right)^2 + \left(\frac{m}{L_y}\right)^2} \tag{10.4}$$

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 $m = 1, 2, 3, ...$ and $n = 1, 2, 3, ...$ 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 10.19 shows a comparison of the first three exact and computed eigenvalues of the square clamped membrane, and Table 10.20 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.

Since they are coupled tests, they have to be located in the tempo test are under sierra. In a tempo project checked out under sierra, these tests are located in the following directories

```
tempo/tempo/rtest/tempo/membrane_free_free
tempo/tempo/rtest/tempo/membrane_free_free_par
tempo/tempo/rtest/tempo/membrane_clamped
tempo/tempo/rtest/tempo/membrane_clamped_par
```

For in-plane loading, there are 2 verification tests located at

exact (Hz)	computed
9.3169	9.3553
13.178	13.230
18.634	18.941
18.634	18.941

Table 10.20: 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 involve in-plane tension only, and simply verify that the corresponding deformation of the membrane is correct.

10.1.9 Verification of Tangent Stiffness Matrix for Sierra Transfers

In this section, we present numerical experiments to confirm the implementation of the tangent stiffness matrix following a Sierra transfer. We note that the tangent stiffness matrix is the sum of contributions from internal forces as well as external forces. In some texts, the contributions from the former is referred to as the material/geometric stiffness, while the former is referred to as the follower stiffness. In the following experiments, all components of the stiffness matrix are being exercised.

10.1.9.1 A Cantilever Beam Subjected to Large Deflection Via End Load

In this example we consider a cantilever beam that is subjected to a large deflection from a concentrated end load. We note that in this case, the follower stiffness is zero, since the load does not depend on the deformation. In this example, Adagio was used to model the deformation of the beam to the large deflection state, and then the results were passed to **Sierra/SD** for modal analysis. Table 10.21 shows the modal frequencies of the beam in the deformed state, compared with those obtained from abaqus. Excellent agreement is seen between the two codes.

10.1.9.2 A Cantilever Beam Subjected to Large Deflection Via Pressure Load

In this section, we consider the same cantilever beam as in the previous example, except in this case the beam is loaded with a distributed pressure load rather than a point load. Since the pressure will follow the beam's deformation, we expect a contribution from the follower stiffness in this case. Table 10.22 shows the comparison of **Sierra/SD** with Abaqus

Table 10.21: Comparison of **Sierra/SD** and Abaqus modal results for a cantilever beam subjected to large deflection via point load.

mode	Abaqus	**Sierra/SD**	% difference
1	56.219	56.236	0.029
2	245.720	246.106	0.154
3	274.010	274.159	0.054
4	358.280	358.316	0.010
5	400.030	399.916	0.028
6	630.540	630.113	0.058
7	649.890	650.113	0.034
8	803.580	803.389	0.024
9	933.100	933.198	0.011
10	1069.80	1070.180	0.036

for the first ten modes of the pressure-loaded beam. In this case, follower stiffness was not included in the **Sierra/SD** results. Some significant differences in the frequencies is observed.

In table 10.23, the same comparison is presented, only in this case the follower stiffness matrix is included in the **Sierra/SD** tangent stiffness matrix calculation. In this case, excellent agreement with Abaqus is obtained. We note that this example is included in the **Sierra/SD** test suite, in the following location

```
Salinas_test/verification/follower/beam_test
```

Table 10.22: Comparison of **Sierra/SD** and Abaqus modal results for a cantilever beam subjected to large deflection via pressure load, with no follower stiffness in the **Sierra/SD** tangent matrix.

mode	Abaqus	**Sierra/SD**	% difference
1	59.015	57.019	3.382
2	60.472	59.858	1.015
3	252.140	230.927	8.413
4	306.200	304.988	0.396
5	322.590	322.217	0.116
6	493.650	492.184	0.297
7	742.200	736.837	0.723
8	770.830	769.096	0.225
9	773.340	771.410	0.250
10	1230.500	1227.530	0.241

Table 10.23: Comparison of **Sierra/SD** and Abaqus modal results for a cantilever beam subjected to large deflection via pressure load, with follower stiffness in the Sierra/SD tangent matrix.

mode	Abaqus	**Sierra/SD**	% difference
1	59.015	59.053	0.064
2	60.472	60.470	0.003
3	252.140	252.194	0.021
4	306.200	306.141	0.019
5	322.590	322.651	0.018
6	493.650	493.719	0.013
7	742.200	742.064	0.019
8	770.830	771.112	0.036
9	773.340	773.366	0.003
10	1230.500	1230.25	0.020

10.1.10 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 10.22, 10.23, 10.24 shows the comparison of the X, Y, and Z displacements as a

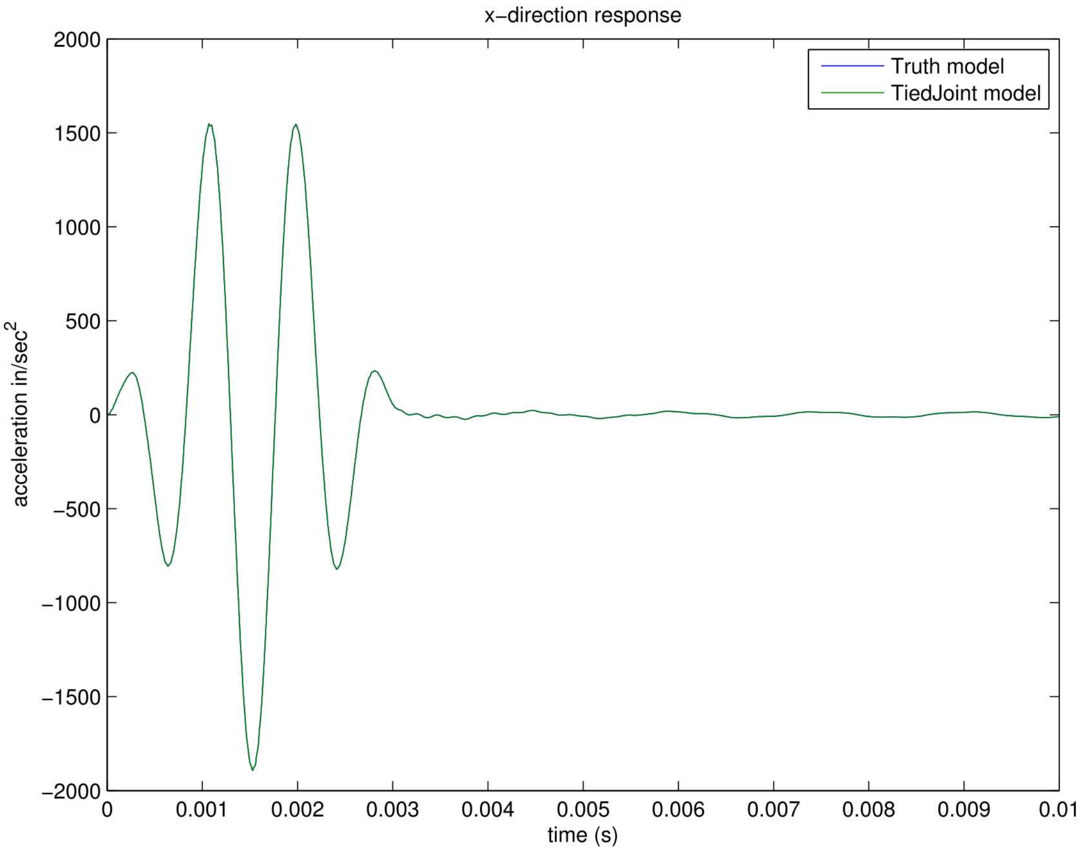


Figure 10.22: X displacement comparison for tied joint versus truth model, tied=slip, side=free

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

```
TIED JOINT
     normal definition = none
     side = rigid
        . . .
END
```

Figures 10.25, 10.26, 10.27 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
```

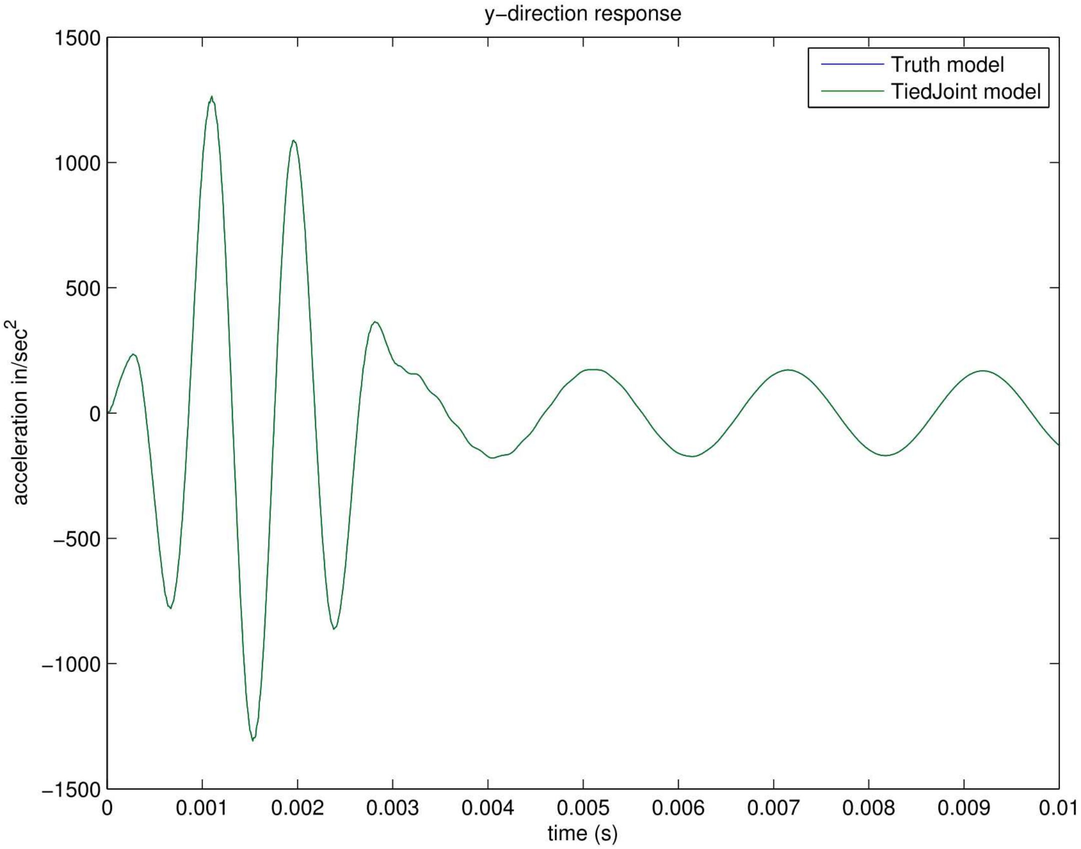


Figure 10.23: Y displacement comparison for tied joint versus truth model, tied=slip, side=free

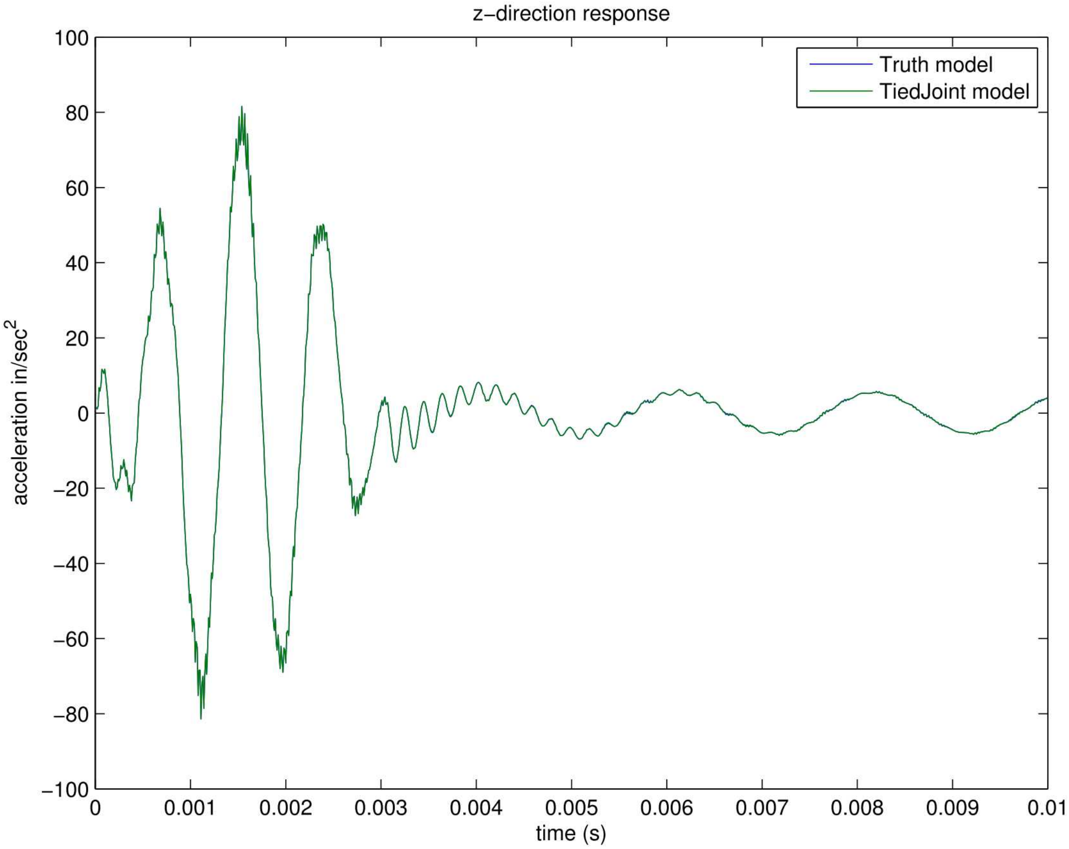


Figure 10.24: Z displacement comparison for tied joint versus truth model, tied=slip, side=free

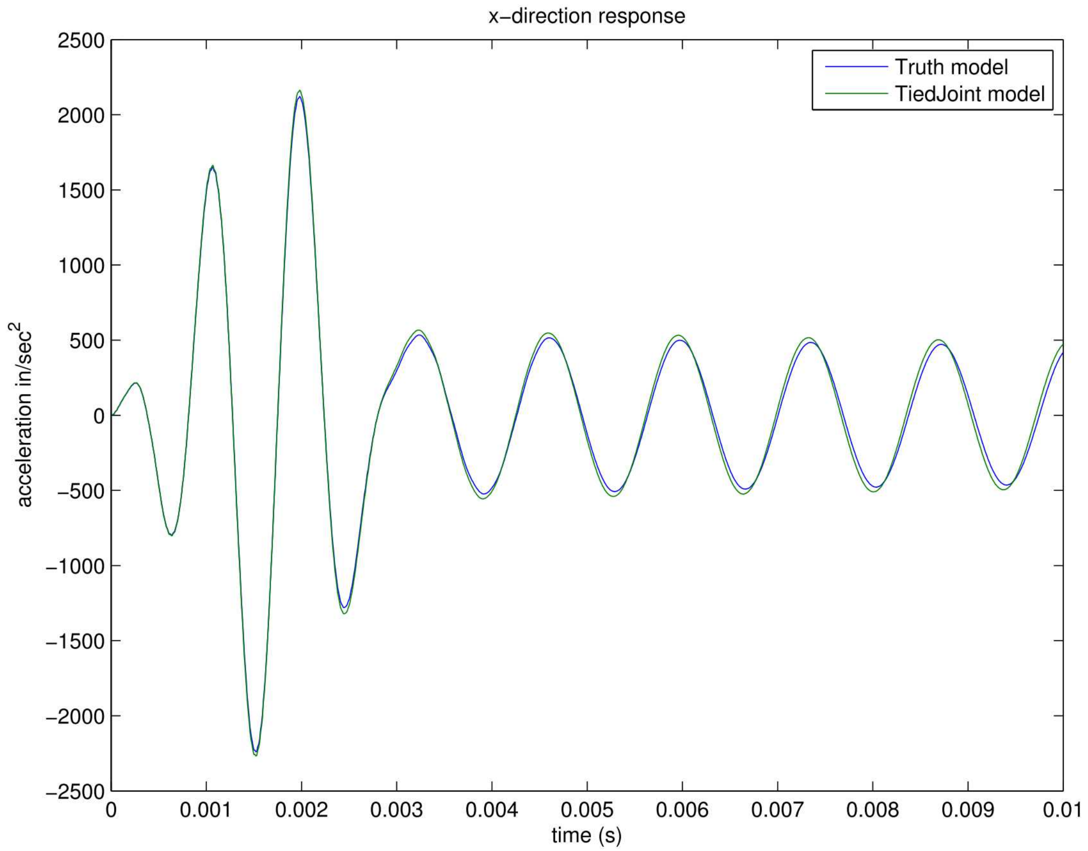


Figure 10.25: X displacement comparison for tied joint versus truth model, tied=none, side=rigid

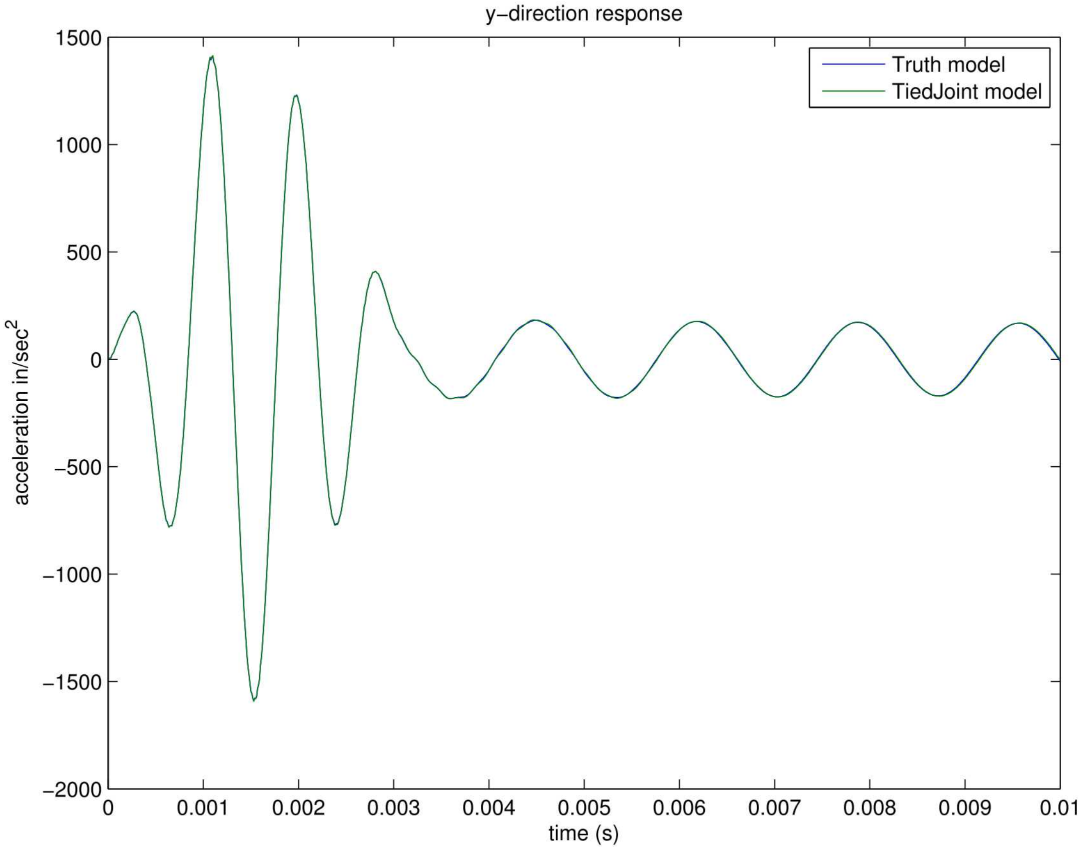


Figure 10.26: Y displacement comparison for tied joint versus truth model, tied=none, side=rigid

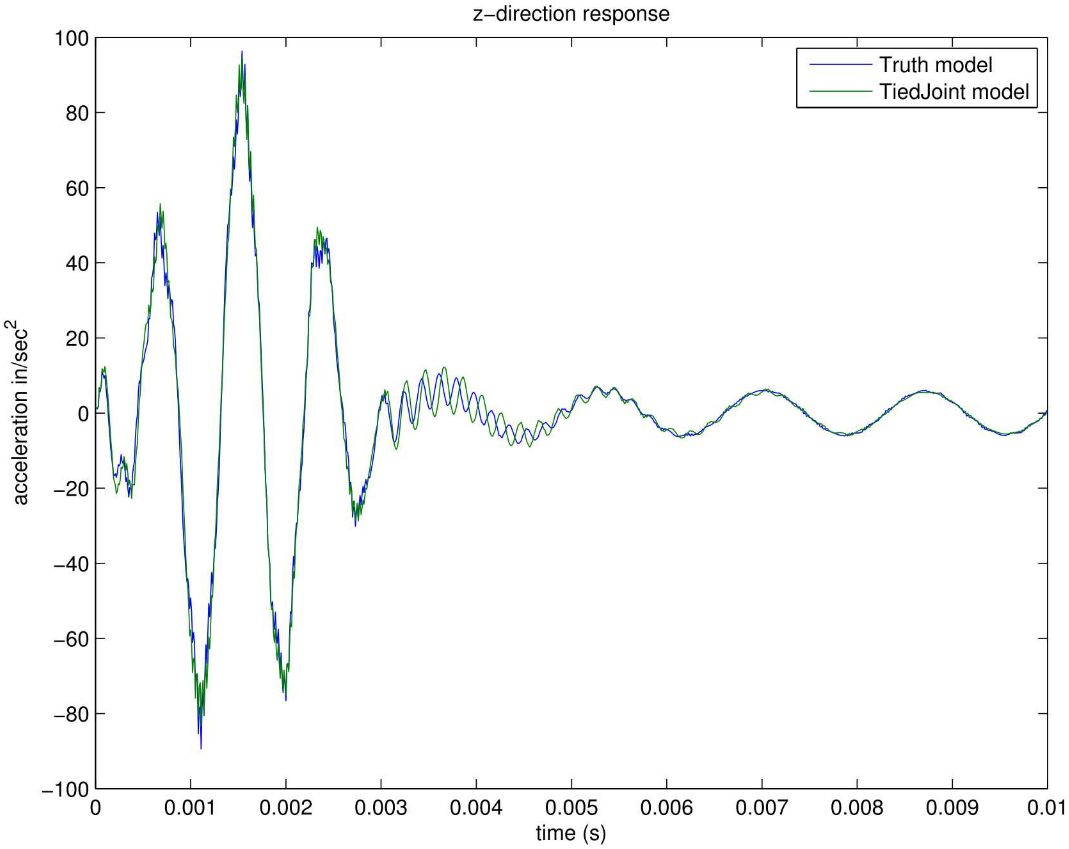


Figure 10.27: Z displacement comparison for tied joint versus truth model, tied=none, side=rigid

10.1.11 Rrodset

The rrodset mimics a kevlar type material in that in 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 kind of pivot around the center where the rrodset can be placed. It was shown that a statics solution produced the same results whether or not an rrodset was placed 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`.

10.1.12 Elements Provided by the Navy

As part of the Navy/CREATE program,[32,33] 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".

10.1.12.1 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 10.24. Table 10.25 summarizes some of the 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 10.24: Static Tests for NBeam

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

Table 10.25: 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`.

10.2 Acoustics

In the following examples computational results are compared to analytic solutions.

10.2.1 Eigen Analysis of Wave Tube

10.2.2 Eigen Analysis with Multiple Fluids

10.2.3 Eigen Analysis of Elliptic Tank

10.2.5 Direct Frequency Response

10.2.5 Transient Acoustics with Pressure Release

10.2.6 Nonconforming Acoustic-Acoustic Discretizations

10.2.7 Direct FRF of Tied Structural/Acoustics

10.2.8 Radiation from a uniformly-driven spherical shell

10.2.9 Radiation from a spherical acoustic surface

10.2.10 Scattering from a Flat Plate

10.2.11 Transient Scattering from a Flat Plate

10.2.12 Scattering a Plane Step Wave by a Spherical Shell

10.2.13 Infinite Elements on Ellipsoidal Surfaces

10.2.14 Comparison of spherical and ellipsoidal infinite elements

10.2.15 Absorbing Boundary Conditions for Infinite Elastic Spaces.

10.2.16 Impedance Boundary Conditions

10.2.17 Point Acoustic Source

10.2.18 Moving Point Source

10.2.19 Infinite Elements for Transients

10.2.20 Comparison with Absorbing Boundary Conditions

10.2.21 Acoustic-Structure Directfrf with Viscoelastic Material

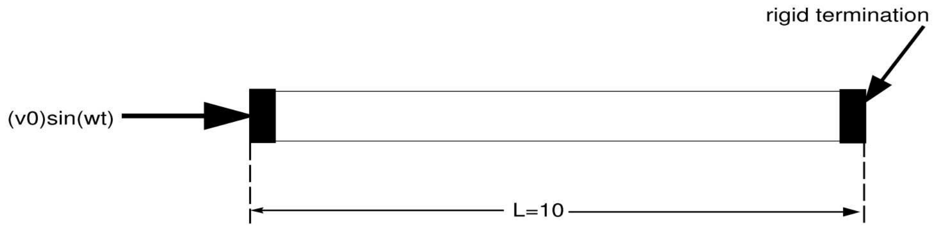


Figure 10.28: Acoustical waveguide with rigid end cap.

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 10.26: Eigenvalue convergence for a piston-driven tube with rigid cap at end. The values given are the natural frequencies, in Hz.

10.2.1 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 Fig. 10.28. The eigenvalue problem for this configuration was solved using several linear hexahedral meshes, each with half the edge length of the previous.

Table 10.26 shows the numerical results, and demonstrates that the first three natural frequencies approach the exact values. Table 10.27 demonstrates quadratic convergence for the natural frequencies, as expected for linear elements.

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 10.27: 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.

10.2.2 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

$$\frac{1}{c^2}\ddot{p} - \Delta p = 0 \tag{10.5}$$

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[34]

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) - \frac{\ddot{p}}{B} = 0 \tag{10.6}$$

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

$$\rho \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) - \frac{\ddot{p}}{c^2} = 0 \tag{10.7}$$

Next, we consider how the heterogeneous wave equation is implemented in Sierra/SD. We note that Sierra/SD actually uses the form in equation 10.7. 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 10.7 varies with position, and thus we will not be able to move the ∇ symbol over to the test function. Thus, we have

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) - \frac{\ddot{p}}{\rho c^2} = 0 \tag{10.8}$$

We actually solve for the time derivative of pressure in Sierra/SD. Thus, we substitute $p = \dot{\phi}$ into equation 10.8, and then integrate in time to obtain

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) - \frac{\ddot{\phi}}{\rho c^2} = 0 \tag{10.9}$$

Now we see that the gradient ∇ can be moved to the test function in equation 10.9. 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 10.6, 10.7, 10.8, and 10.9 will all be the same (assuming we converted the final analytic solution from equation 10.9 into pressure), since these equations are just different by a scale factor. Thus, we simply use equation 10.6 to derive the analytical solution. If we consider the eigenvalue problem, equation 10.6 becomes

$$\nabla \cdot \left(\frac{1}{\rho} \nabla p \right) + \lambda \frac{p}{B} = 0 \tag{10.10}$$

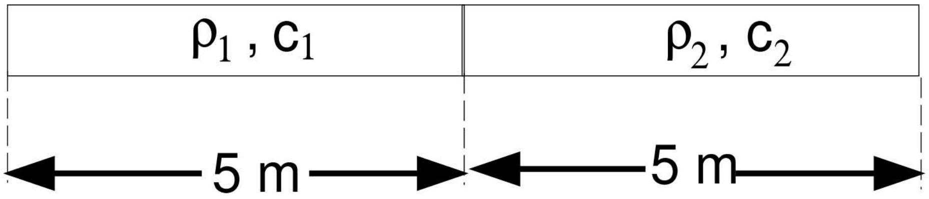


Figure 10.29: Acoustical waveguide containing two fluids.

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 10.29. An exact solution for the eigenvalues of the geometry in Figure 10.29 can be derived by considering each fluid separately and applying appropriate compatibility conditions on the fluid-fluid interface. The equations are as follows

$$\frac{d^2p_1}{dx^2} + \lambda\frac{\rho_1}{B_1}p_1 = 0 \quad 0 \leq x \leq \frac{L}{2} \tag{10.11}$$

$$\frac{d^2p_2}{dx^2} + \lambda\frac{\rho_2}{B_2}p_2 = 0 \quad \frac{L}{2} \leq x \leq L \tag{10.12}$$

$$\tag{10.13}$$

$$p_1 = p_2 \quad x = \frac{L}{2} \tag{10.14}$$

$$\frac{1}{\rho_1}\frac{dp_1}{dx} = \frac{1}{\rho_2}\frac{dp_2}{dx} \quad x = \frac{L}{2} \tag{10.15}$$

$$\tag{10.16}$$

where B_1 and B_2 are the bulk moduli of the two fluids. At the endpoints, there are two options. Either we could have rigid caps ($\frac{dp}{dx} = 0$), or we could have pressure release boundary conditions ($p = 0$). The solution will have the form

$$p_1(x) = C_1\cos\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) + C_2\sin\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \quad 0 \leq x \leq \frac{L}{2} \tag{10.17}$$

$$p_2(x) = C_3\cos\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) + C_4\sin\left((x - \frac{L}{2})\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \quad \frac{L}{2} \leq x \leq L \tag{10.18}$$

$$\tag{10.19}$$

Inserting these into equations 10.12, applying the compatibility conditions 10.15, and using the appropriate boundary conditions at the endpoints, we get two transcendental equations

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 10.28: 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
17.0965	17.1143	17.0976
35.4575	35.6039	35.4666
51.3135	51.7932	51.3435

Table 10.29: Two-fluid eigenvalue convergence with pressure release BC.

that give the exact eigenvalues. For the pressure release (Dirichlet) end cap case, we obtain

$$\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right)\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right) = -\sqrt{\frac{\rho_1 B_1}{\rho_2 B_2}}\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right)\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \tag{10.20}$$

$$\tag{10.21}$$

For the rigid (Neumann) case, we obtain

$$\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right)\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right) = -\sqrt{\frac{\rho_1 B_1}{\rho_2 B_2}}\sin\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_2}{B_2}}\right)\cos\left(\frac{L}{2}\sqrt{\frac{\lambda\rho_1}{B_1}}\right) \tag{10.22}$$

$$\tag{10.23}$$

Equations 10.20 and 10.22 can be solved to obtain the exact eigenvalues of the system shown in Figure 10.29.

First, we consider the case $\rho_1 = 1.293$, $\rho_2 = 2.5860$, $c_1 = 332.0$, $c_2 = 366.0$. Table 10.28 shows the comparison when rigid walls are placed at either end of the tube, and Table 10.29 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 10.30. 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 10.31.

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 10.30: 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 10.31: Eigenvalue convergence for an air/water system with rigid cap at ends. The values given are the natural frequencies, in Hz.

10.2.3 Eigen Analysis of Elliptic Tank

This section written by Jerry Rouse.

The acoustic eigenanalysis capability of Sierra/SD was further verified using a three dimensional elliptic cylindrical tank. The dimensions of the tank are shown in Figure 10.30. 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 endcaps 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.

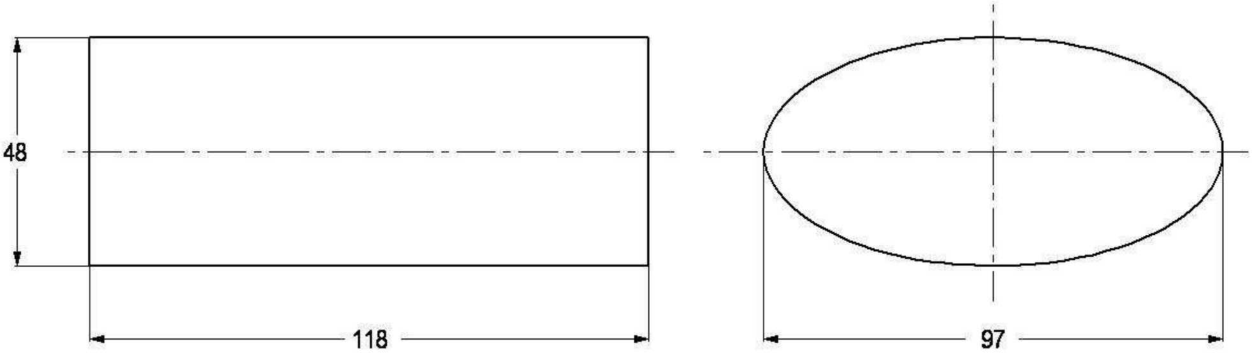


Figure 10.30: Dimensions of the elliptic cylindrical tank model. All dimensions in inches.

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 10.31. This coordinate system is not commonly encountered, and therefore the solution of the wave equation is described. The linear wave equation in terms

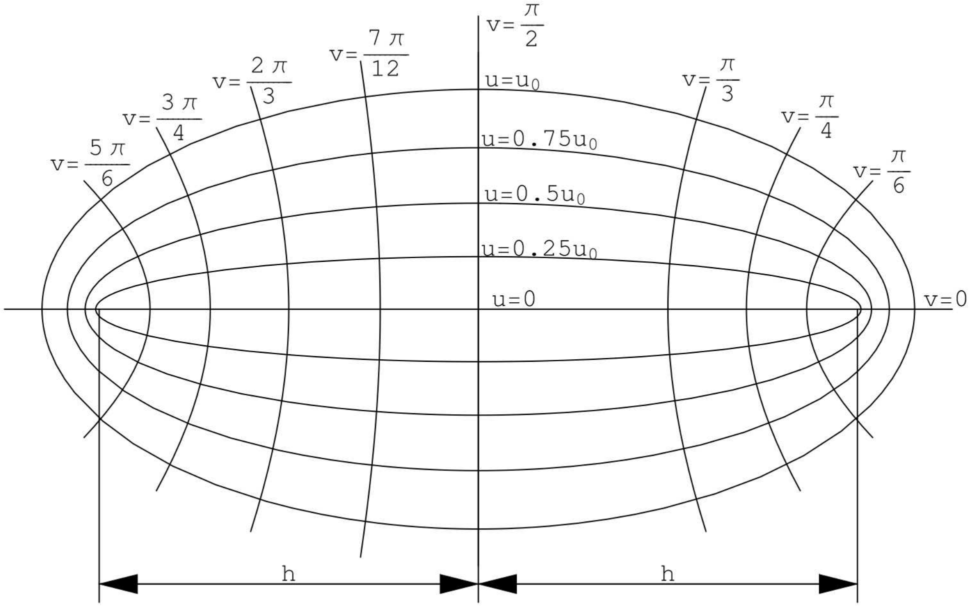


Figure 10.31: The elliptic cylindrical coordinate system.

of acoustic pressure is given by

$$\nabla^2 p + \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0. \tag{10.24}$$

In elliptic cylindrical coordinates the Laplacian has the form

$$\nabla^2 = \frac{1}{h^2(\sinh^2(u) + \sin^2(v))}\left(\frac{\partial^2 p}{\partial u^2} + \frac{\partial^2 p}{\partial v^2}\right) + \frac{\partial^2 p}{\partial z^2} \tag{10.25}$$

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 10.30 $a = \frac{97}{2}$, $b = 24$, $h = \frac{7\sqrt{145}}{2}$, 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. (10.24) produces the Helmholtz equation:

$$\nabla^2 P + k^2 P = 0, \tag{10.26}$$

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 and dividing the result by UVZ gives

$$\frac{1}{h^2(\sinh^2(u) + \sin^2(v))}\left(\frac{1}{U}\frac{d^2U}{du^2} + \frac{1}{V}\frac{d^2V}{dv^2}\right) + \frac{1}{Z}\frac{d^2Z}{dz^2} + k^2 = 0. \tag{10.27}$$

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

$$k^2 = k_z^2 + m^2. \tag{10.28}$$

The differential equation for Z,

$$\frac{d^2Z}{dz^2} + k_z^2 Z = 0, \tag{10.29}$$

has solution

$$Z(z) = A_n \cos(k_z z) + B_n \sin(k_z z). \tag{10.30}$$

Simplifying the differential equation for U and V gives:

$$\left[\frac{1}{U}\frac{d^2U}{du^2} + m^2h^2\sinh^2(u)\right] + \left[\frac{1}{V}\frac{d^2V}{dv^2} + m^2h^2\sin^2(v)\right] = 0. \tag{10.31}$$

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:

$$\frac{1}{U}\frac{d^2U}{du^2} + m^2h^2\sinh^2(u) = c \rightarrow \frac{d^2U}{du^2} - \left[c - m^2h^2\sinh^2(u)\right]U = 0 \tag{10.32}$$

$$\frac{1}{V}\frac{d^2V}{dv^2} + m^2h^2\sin^2(v) = -c \rightarrow \frac{d^2V}{dv^2} + \left[c + m^2h^2\sin^2(v)\right]V = 0 \tag{10.33}$$

The trigonometric relations

$$\sinh^2(u) = \frac{1}{2}\left(\cosh(2u) - 1\right) \tag{10.34}$$

$$\sin^2(v) = \frac{1}{2}\left(1 - \cos(2v)\right) \tag{10.35}$$

are used to simply Eq. (10.32) and Eq. (10.33). Substitution of these relations into the differential equations for U and V gives:

$$\frac{d^2U}{du^2} - \left[\left(c + \frac{m^2h^2}{2}\right) - \frac{m^2h^2}{2}\cosh(2u)\right]U = 0 \tag{10.36}$$

$$\frac{d^2V}{dv^2} + \left[\left(c + \frac{m^2h^2}{2}\right) - \frac{m^2h^2}{2}\cos(2v)\right]V = 0. \tag{10.37}$$

Letting $a \equiv c + \frac{m^2h^2}{2}$ and $q \equiv \frac{m^2h^2}{4}$ gives:

$$\frac{d^2U}{du^2} - \left[a - 2q\cosh(2u)\right]U = 0 \tag{10.38}$$

$$\frac{d^2V}{dv^2} + \left[a - 2q\cos(2v)\right]V = 0 \tag{10.39}$$

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

$$V = C_r ce_r(a, q, v) + D_r se_r(a, q, v), \tag{10.40}$$

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

$$U = E_r Ce_r(a, q, u) + F_r Se_r(a, q, u), \tag{10.41}$$

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:

$$Ce_r(a, q, z) = ce_r(a, q, iz) \tag{10.42}$$

$$Se_r(a, q, z) = -ise_r(a, q, iz), \tag{10.43}$$

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 Stegan. 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

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\vec{\nabla} p, \tag{10.44}$$

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

$$\vec{\nabla} = \frac{1}{h\sqrt{\sinh^2(u) + \sin^2(v)}} \left(\vec{e}_u \frac{\partial}{\partial u} + \vec{e}_v \frac{\partial}{\partial v} \right) + \vec{e}_z \frac{\partial}{\partial z}. \tag{10.45}$$

Subsitution of the z component of pressure in Eq. (10.30) into Eq. (10.44), and applying the $u_z = 0$ boundary condition gives

$$Z(z) = \sum_{n=0}^{\infty} A_n \cos(k_z z), \tag{10.46}$$

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 $u = u_0$. Subsitution of

Eq. (10.41) into Eq. (10.44) and applying this boundary condition gives

$$\sum_{r=1}^{\infty}\left[E_r\frac{\partial Ce_r(a,q,u)}{\partial u}\bigg|_{u=u_0}ce_r(a,q,v)+F_r\frac{\partial Se_r(a,q,u)}{\partial u}\bigg|_{u=u_0}se_r(a,q,v)\right]=0, \tag{10.47}$$

where $\frac{\partial Ce_0(a,q,v)}{\partial u}=0$. To satisfy this equation requires each term of the series equal zero, giving

$$\frac{\partial Ce_r(a_r,q_r,u)}{\partial u}\bigg|_{u=u_0}=0 \tag{10.48}$$

$$\frac{\partial Se_r(b_r,q_r,u)}{\partial u}\bigg|_{u=u_0}=0, \tag{10.49}$$

where the resonance frequencies are determined from the values of q which satisfy Eqs. (10.48) and (10.49). 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. (10.46) and $m=\frac{4\sqrt{q_r}}{h}$ obtained from Eqs. (10.48) and (10.49)

$$f=\frac{c}{2\pi}\sqrt{\left(\frac{n\pi}{L}\right)^2+\frac{4q_r}{h^2}}, \tag{10.50}$$

where $c=58724$ in/s. Table 10.32 compares the first 24 resonance frequencies between the exact determination and the Sierra/SD prediction for the case of completely rigid boundary conditions.

The boundary condition configuration having pressure release boundaries $p=0$ on the sidewall of the elliptic cylindrical tank (and rigid endcaps) requires the acoustic pressure be zero at $p(u_0,v,z)$. Applying this condition to Eq. (10.41) gives

$$\sum_{r=0}^{\infty}\left[E_rCe_r(a,q,u_0)ce_r(a,q,v)+F_rSe_r(a,q,u_0)se_r(a,q,v)\right]=0. \tag{10.51}$$

As before, to satisfy this condition each term of the series must equal zero, giving

$$Ce_r(a_r,q_r,u_0)=0 \tag{10.52}$$

$$Se_r(b_r,q_r,u_0)=0, \tag{10.53}$$

where the resonance frequencies are obtained from the values of q which satisfy Eqs. (10.52) and (10.53). The complete set of resonance frequencies for the elliptic cylindrical tanker having rigid endcaps and pressure release sidewalls is determined from Eq. (10.50) with $c=58724$ in/s. Table 10.33 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.

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 10.32: Comparison between the exact analytical resonance frequencies and Sierra/SD predictions for the elliptic cylindrical tank with completely rigid boundary boundaries.

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 10.33: Comparison between the exact analytical resonance frequencies and Sierra/SD predictions for the elliptic cylindrical tank having rigid endcaps and pressure release boundary conditions on the sidewall.

10.2.4 Direct Frequency Response

The second example involves the previous configuration, only that a direct frequency response is computed rather than an eigen analysis. Also, in this case two types of boundary conditions are considered, the rigidly capped configuration of the previous example (a Neumann boundary condition), and a pressure release condition (a Dirichlet condition). For the two types of boundary conditions on the right end,[34] gives the exact resonance frequencies. When the tube is rigidly capped, they are

$$f_n = \frac{nc}{2L} \quad n = 0, 1, 2, 3, ... \tag{10.54}$$

and when the tube is open (pressure release) they are

$$f_n = \frac{(n + \frac{1}{2})c}{2L} \quad n = 0, 1, 2, ... \tag{10.55}$$

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

$$f_n = 0.0, 16.6, 33.2, 49.8, ... \tag{10.56}$$

and

$$f_n = 8.3, 24.9, 41.5, ... \tag{10.57}$$

Figures 10.32 and 10.33 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[31] as

$$p = -j\rho c V_0 cot(kL) \tag{10.58}$$

In Figure 10.34, 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 very inaccurate, and thus some difference there is expected.

10.2.5 Transient Acoustics with Pressure Release

This example was very similar to the previous case, except that the far end of the tube was assigned a pressure release boundary condition, rather than rigid. Also, in this case the velocity of the piston was assigned as

$$v(0, t) = v_p(t) = \sin(\omega t) \tag{10.59}$$

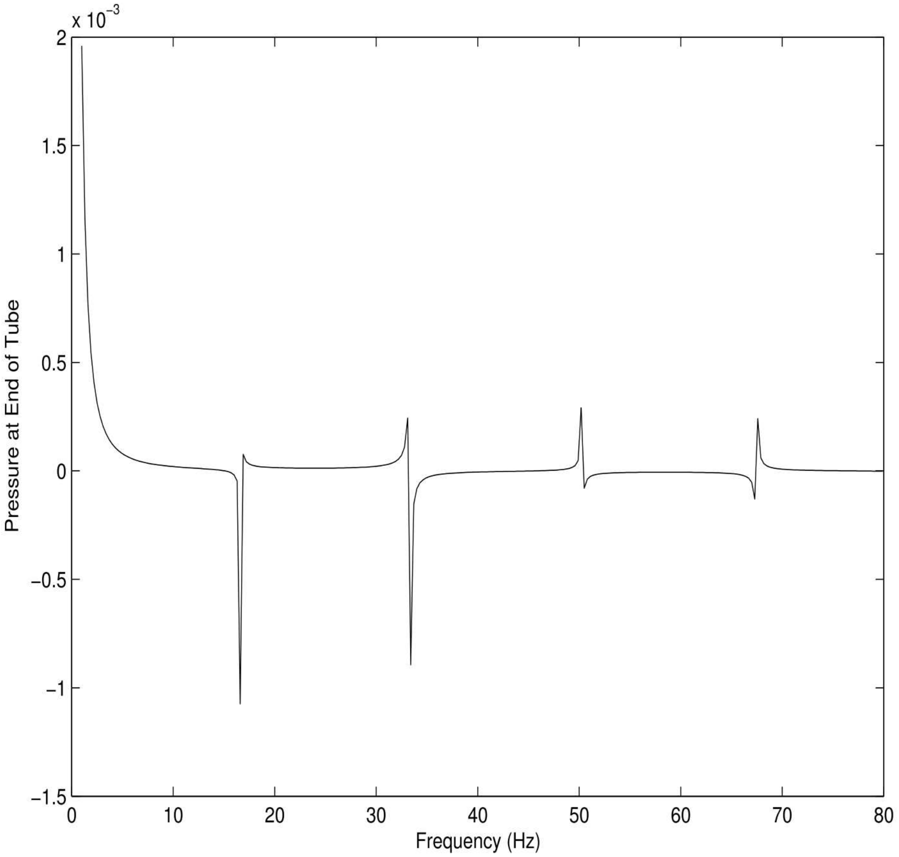


Figure 10.32: Direct frequency response of an acoustical waveguide with rigid end cap.

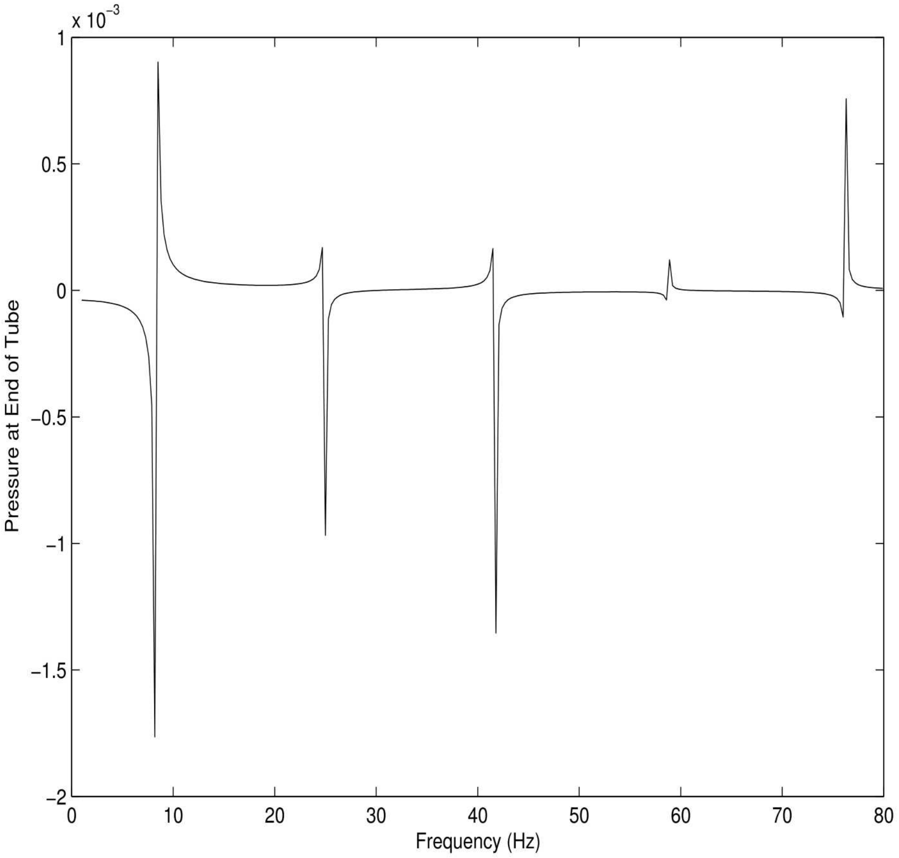


Figure 10.33: Direct frequency response of an acoustical waveguide with pressure release end.

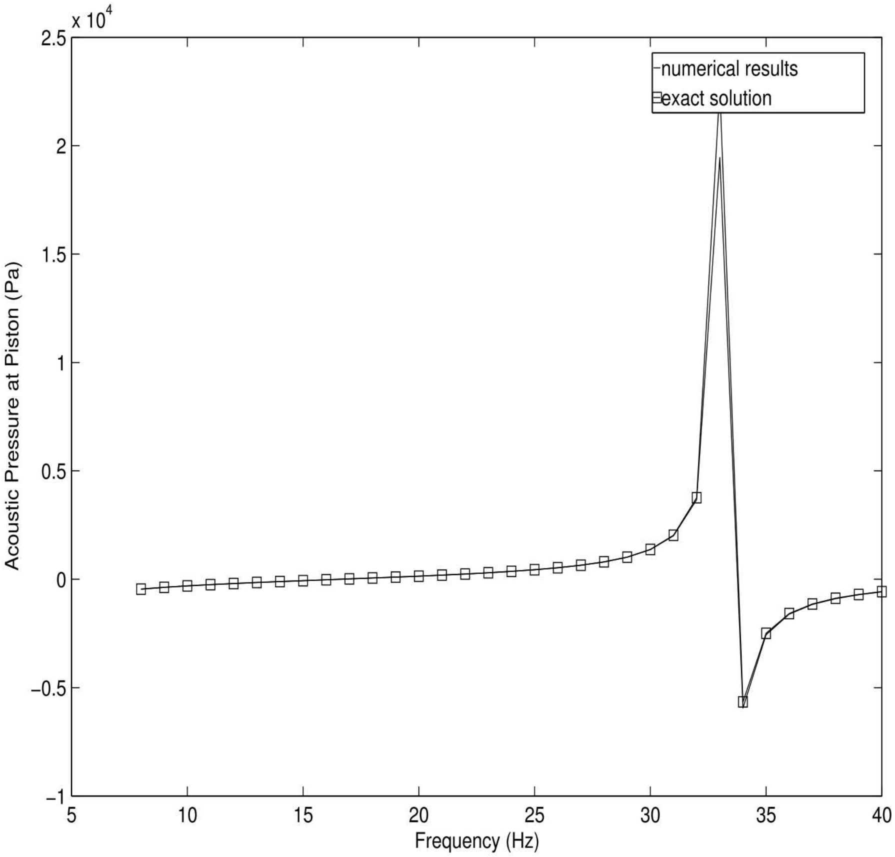


Figure 10.34: Direct frequency response of an acoustical waveguide with rigid end cap. A comparison of computed and exact acoustic pressure at the piston.

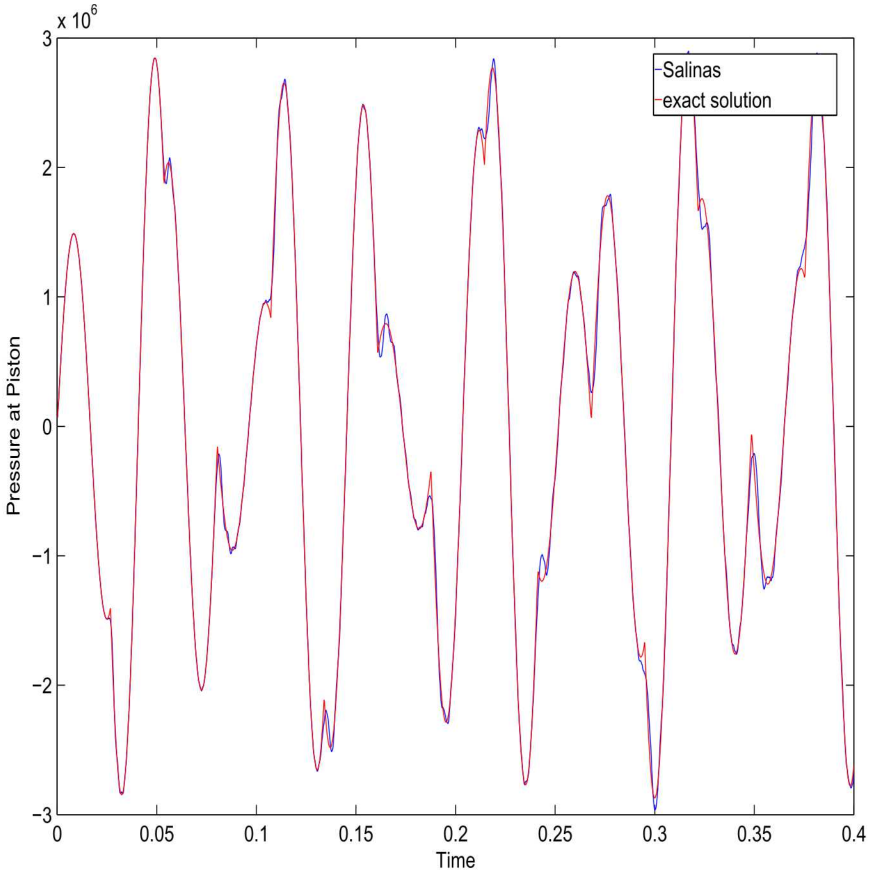


Figure 10.35: Transient simulation of an acoustical waveguide with pressure release end condition.

where $\omega = 60\pi$. The exact solution is given in[34] as

$$p(0,t) = \rho c \left[v_p(t) + 2 \sum_{n=1}^{\infty} (-1)^n v_p(t - \frac{2nL}{c}) \right] \tag{10.60}$$

where again the terms in the summation only become nonzero when their arguments are positive. This behavior was implemented in matlab using Heaviside functions, and the results were compared with Sierra/SD. Figure 10.35 shows the results. Excellent agreement between exact and computed solutions is observed.

10.2.6 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. (10.36). 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. (10.36), 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. We only consider the case of rigid walls, which for acoustics corresponds to homogeneous Neumann boundary conditions. Equations 3.29 from[35] were solved with zero forcing on the right hand side, thus corresponding to the acoustic eigenvalue problem with mismatched meshes on subdomains.

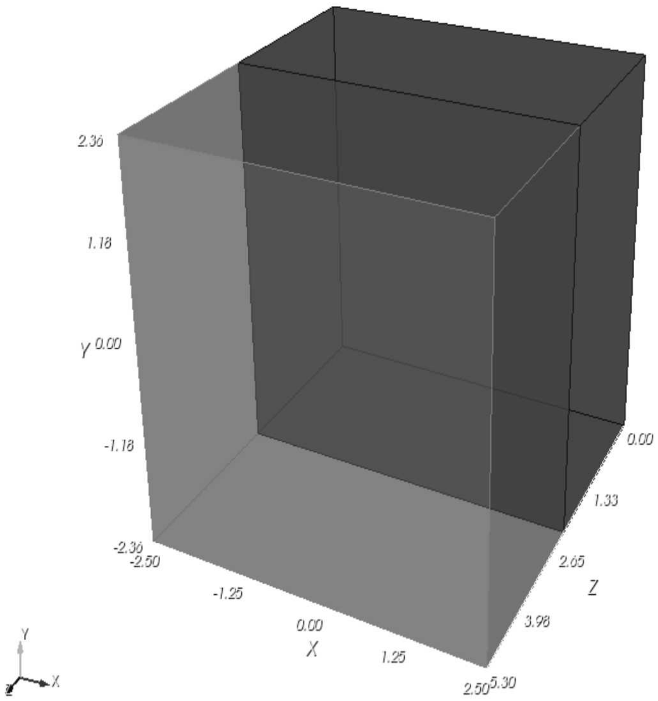


Figure 10.36: Three-dimensional model

Four master/slave 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 master surface is more finely discretized than the slave, see for example.[36,37] In all cases, the master 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 hexahedral elements were considered. The eigenvalues of the first thirty modes in the model were compared to the theoretical eigenvalues given by

$$f = \frac{c}{2}\sqrt{\frac{N_x^2}{L_x^2} + \frac{N_y^2}{L_y^2} + \frac{N_z^2}{L_z^2}}, \tag{10.61}$$

where N_x, N_y, and N_z are non-negative integers. For comparison of the convergence rates, the eigenvalues of a fully conforming model were also obtained. In Figs. (10.37)-(10.39) 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 master side elements. The eigenvalue error is given by $100(\lambda^h - \lambda)/\lambda$. Figures (10.37), (10.38), and (10.39) 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 of 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.

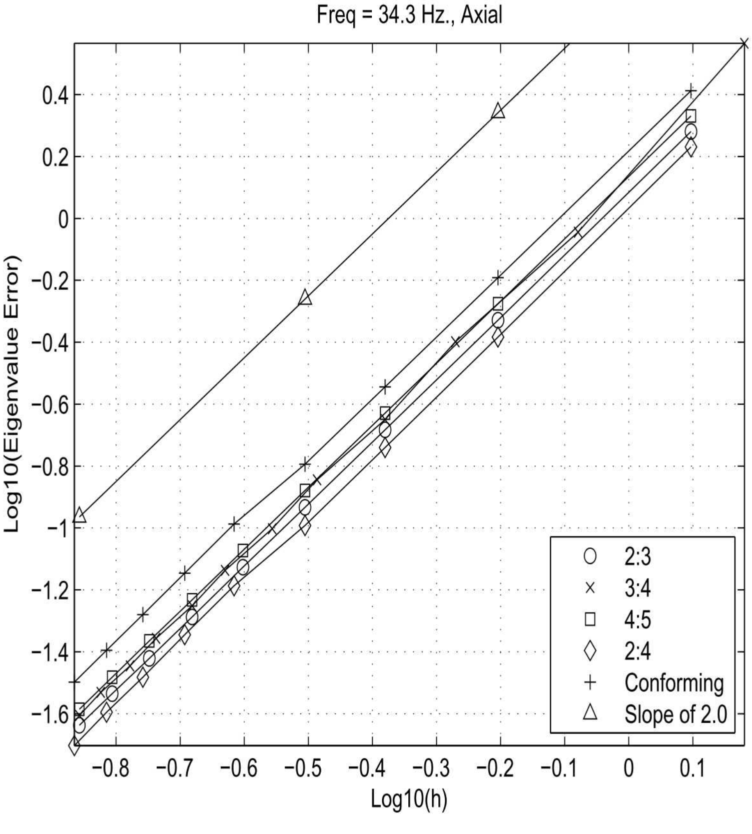


Figure 10.37: Convergence plot for an axial mode ($N_x = 1, N_y = N_z = 0$)

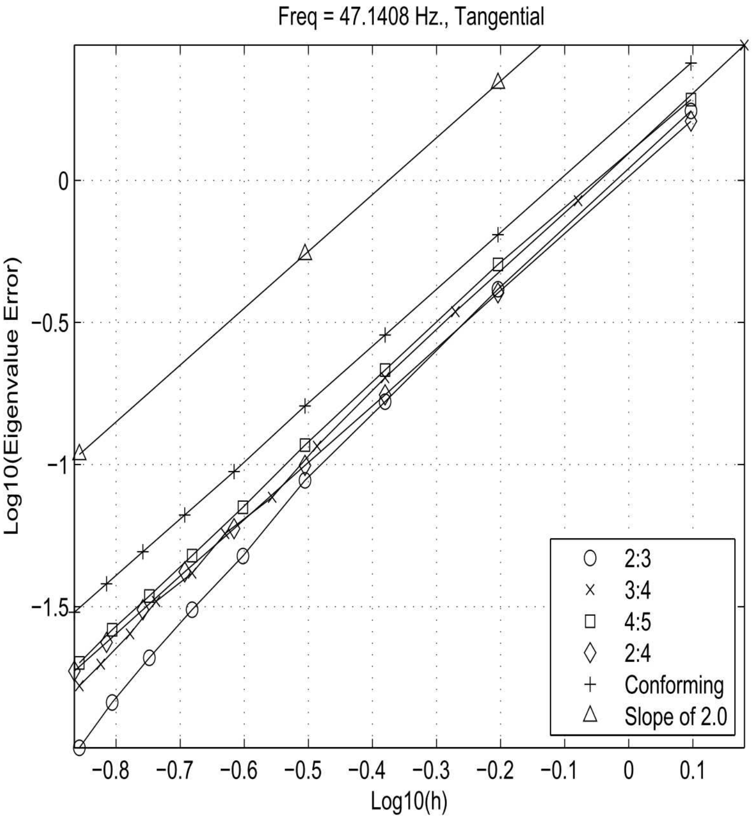


Figure 10.38: Convergence plot for a tangential mode ($N_x = 1, N_y = 0, N_z = 1$)

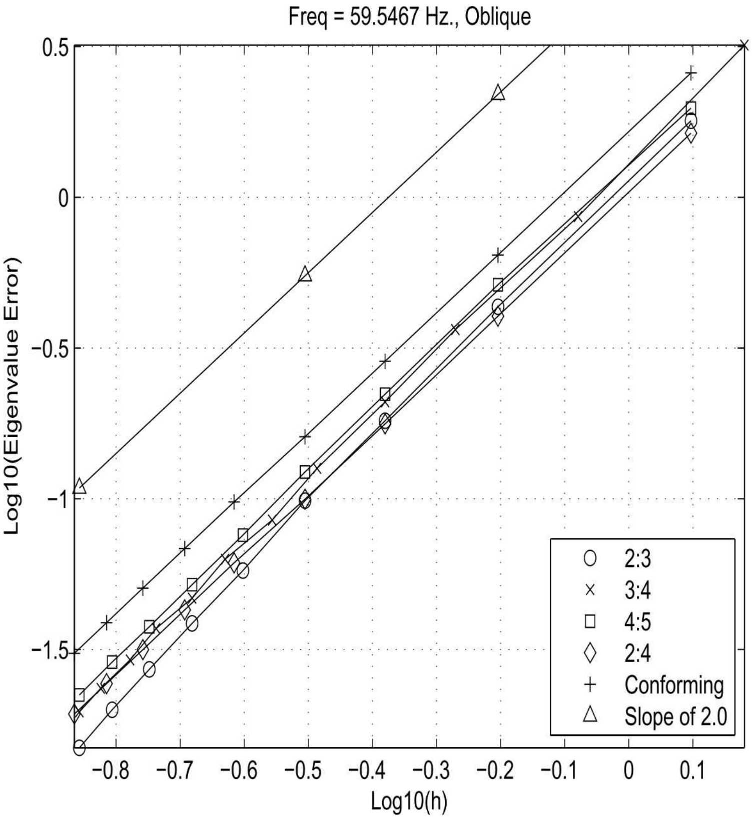


Figure 10.39: Convergence plot for an oblique mode ($N_x = N_y = N_z = 1$)

10.2.7 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 master surface. The model was investigated using the direct frequency response solution in Siera/SD running in serial. The Siera/SD prediction was verified at the tied interface between the fluid and solid regions.

The theoretical reponse of the system was obtained by solving 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

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c_s^2}\frac{\partial^2 u}{\partial t^2} = 0, \tag{10.62}$$

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. The coordinate system for the solid was aligned such that the x_s-axis was directed down 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. (10.62), expressed in terms of left and right traveling waves, gives

$$u = A\sin(k_s x_s)e^{i\omega t}, \tag{10.63}$$

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

$$\frac{\partial^2 p}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0, \tag{10.64}$$

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 directed down 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

$$V_0 = \frac{i}{\omega \rho_0} \frac{\partial p}{\partial x}\bigg|_{x_f=0}. \tag{10.65}$$

Application of this boundary condition to the general solution of Eq. (10.64) gives

$$p = \left[\frac{V_0\,\omega\,\rho_0}{k_f} e^{-ik_f x_f} + B\cos(k_f\,x_f)\right] e^{i\omega t}, \tag{10.66}$$

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

$$E\frac{\partial u}{\partial x}\bigg|_{x_s=-L_s} = -p\bigg|_{x_f=L_f}, \tag{10.67}$$

where tensile stress in the solid is considered positive, and the continuity condition on veloctiy is given by

$$\frac{\partial u}{\partial t}\bigg|_{x_s=-L_s} = \frac{i}{\omega\rho_0}\frac{\partial p}{\partial x}\bigg|_{x_f=L_f}. \tag{10.68}$$

Substitution of Eqs. (10.63) and (10.66) into Eqs. (10.67) and (10.68), and solving for the frequency dependent coefficients A and B finds

$$A = \frac{iV_0\omega\rho_0}{\omega^2\rho_0\sin(k_sL_s)\cos(k_fL_f) + Ek_sk_f\cos(k_sL_s)\sin(k_fL_f)}, \tag{10.69}$$

and

$$B = \frac{-V_0c\rho_0\sin(k_fL_f)e^{-ik_fL_f}\left[\omega^2\rho_0 + iEk_sk_f\cot(k_sL_s)\right]}{\omega^2\rho_0\cot(k_fL_f) + Ek_sk_f\cot(k_sL_s)}. \tag{10.70}$$

With these coefficients now determined, the structural particle displacement is given by

$$u = \frac{iV_0\omega\rho_0\sin(k_sx_s)e^{i\omega t}}{\omega^2\rho_0\sin(k_sL_s)\cos(k_fL_f) + Ek_sk_f\cos(k_sL_s)\sin(k_fL_f)}, \tag{10.71}$$

and the acoustic pressure given by

$$p = \frac{iV_0c\rho_0\sin(k_fL_f)e^{i\omega t}\left[\omega^2\rho_0\sin(k_f(L_f - x_f)) - Ek_sk_f\cot(k_sL_s)\cos(k_f(L_f - x_f))\right]}{\omega^2\rho_0\cot(k_fL_f) + Ek_sk_f\cot(k_sL_s)}. \tag{10.72}$$

The Siera/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 10.40 the Siera/SD prediction of structural particle displacement at the solid-fluid interface is compared to the theoretical result given by Eq. (10.71) evaluated at $x_s = -L_s$. The Siera/SD prediction was obtained over the frequency range 1 to 60 Hz. using a frequency step of 1 Hz. In Figure 10.41 the Siera/SD prediction of acoustic pressure at the solid-fluid interface is compated to the theoretical result given by Eq. (10.72) evaluated at $x_f = L_f$. In both figures the Siera/SD prediction shows excellent agreement with the theoretical result.

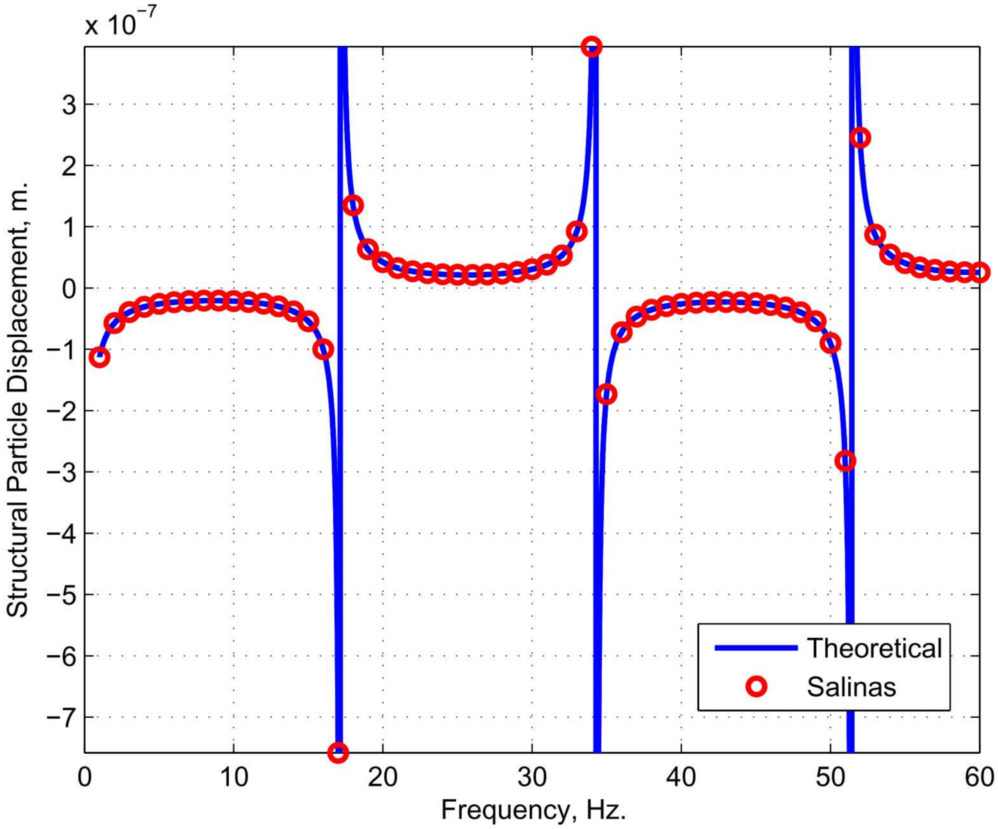


Figure 10.40: Comparison of the Siera/SD prediction of structural particle displacement at the solid-fluid interface with the theoretical result.

10.2.8 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 fully 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

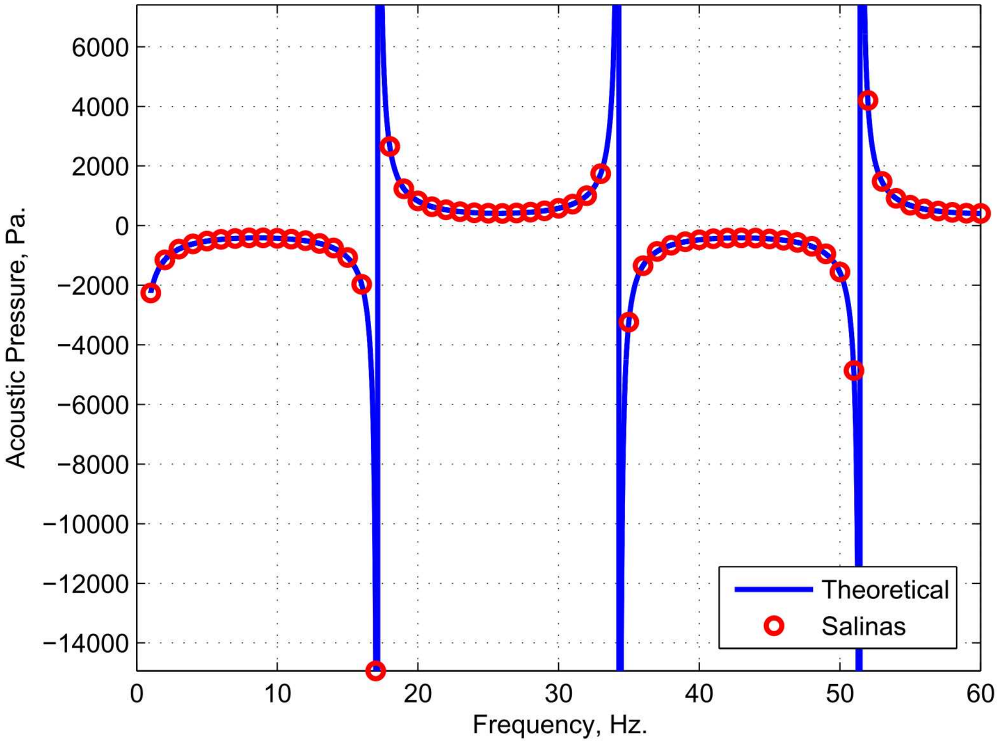


Figure 10.41: Comparison of the Siera/SD prediction of acoustic pressure at the solid-fluid interface with the theoretical result.

domain. The analytic solution to this problem was derived in.[38] First we define some physical quantities. The impedance of the shell structure is given as

$$Z_s = \frac{i}{\omega}(\omega^2 m_s - k_s) \tag{10.73}$$

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 seem by the spherical surface that defines the shell) is

$$Z_f = \frac{i\omega\rho 4\pi a^3}{1 + ika} \tag{10.74}$$

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

$$d = \frac{4\pi a^2 p_0}{i\omega(Z_s + Z_f})\tag{10.75}$$

Figure 10.42 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.

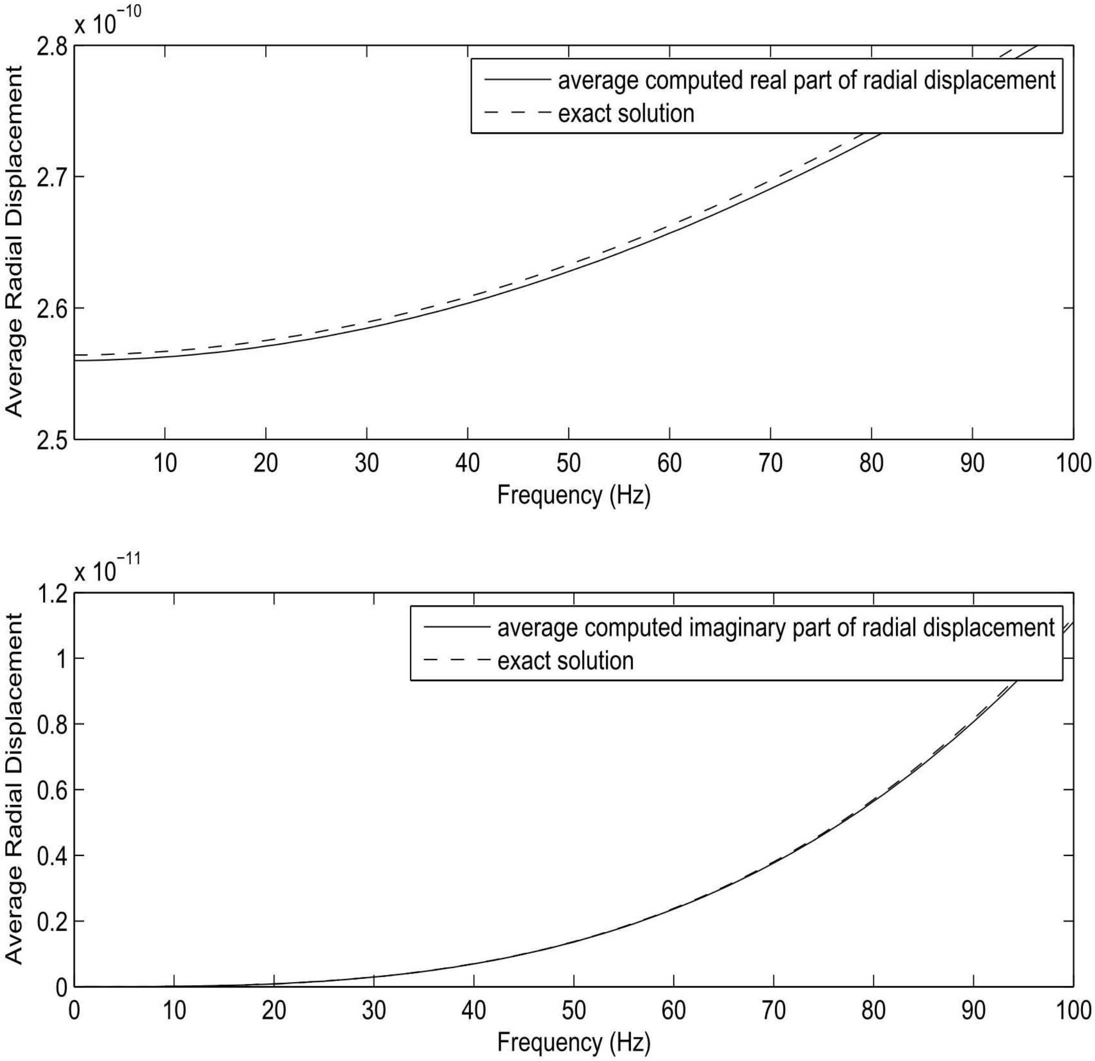


Figure 10.42: Direct frequency response of a spherical shell immersed in an infinite fluid. The real and imaginary parts of the analytical solution are compared against Sierra/SD. The results show good agreement.

10.2.9 Radiation from a uniformly driven spherical acoustic surface

This example is very 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[34]

$$P = \frac{iv_0 \omega \rho a^2}{r(1 + ika)} e^{ik(r-a)} \tag{10.76}$$

where v_0 is the amplitude of the imposed particle velocity on the driven surface.

Figure 10.43 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.

10.2.10 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 looks like 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 10.44 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 very light fluid that has a high speed of sound ($\rho = 1.0$, $c = 1500.0$). 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 so 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 10.45 shows the displacement of a corner point on the plate as a function frequency.

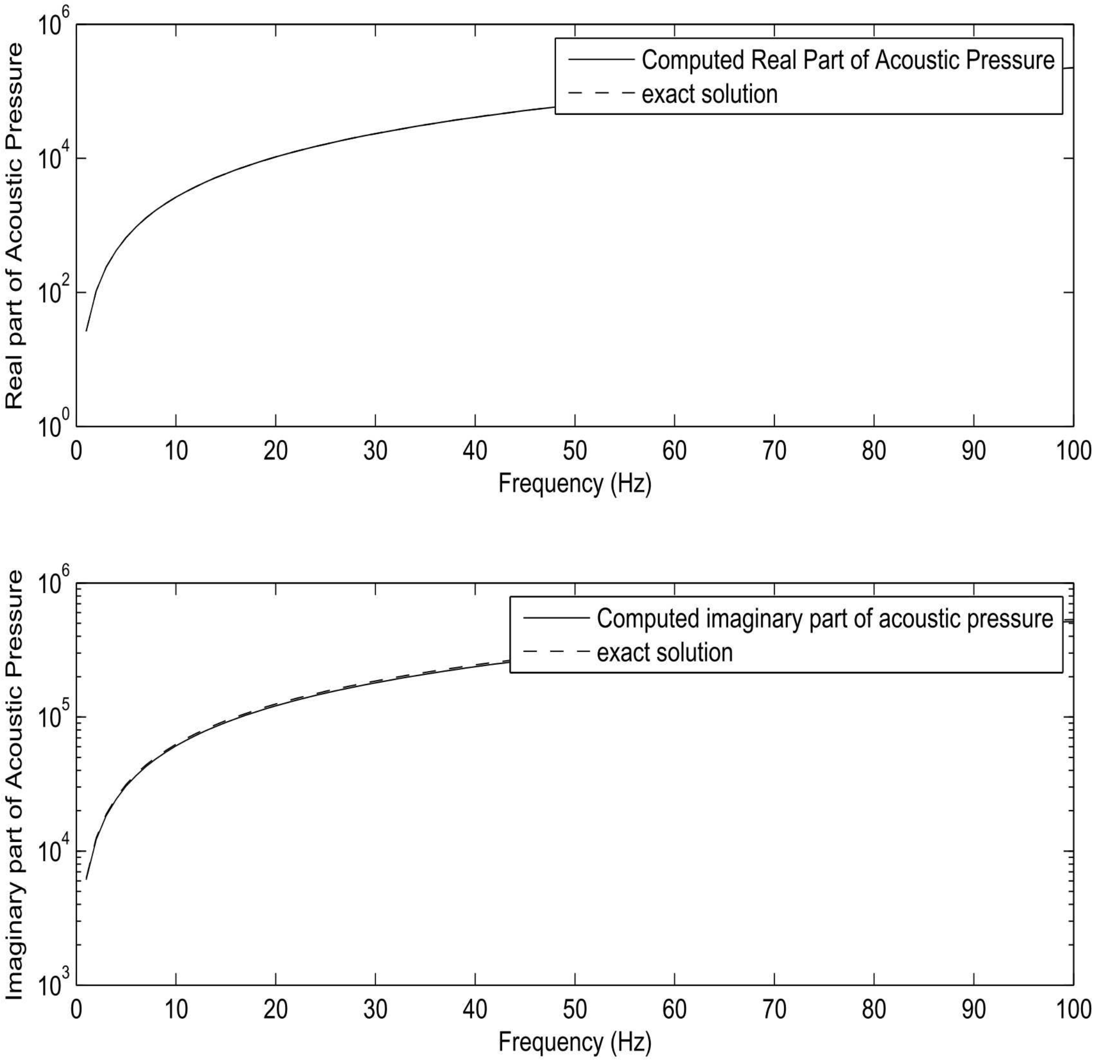


Figure 10.43: Direct frequency response of a spherical shell immersed in an infinite fluid. The real and imaginary parts of the analytical solution are compared against Sierra/SD. The results show good agreement.

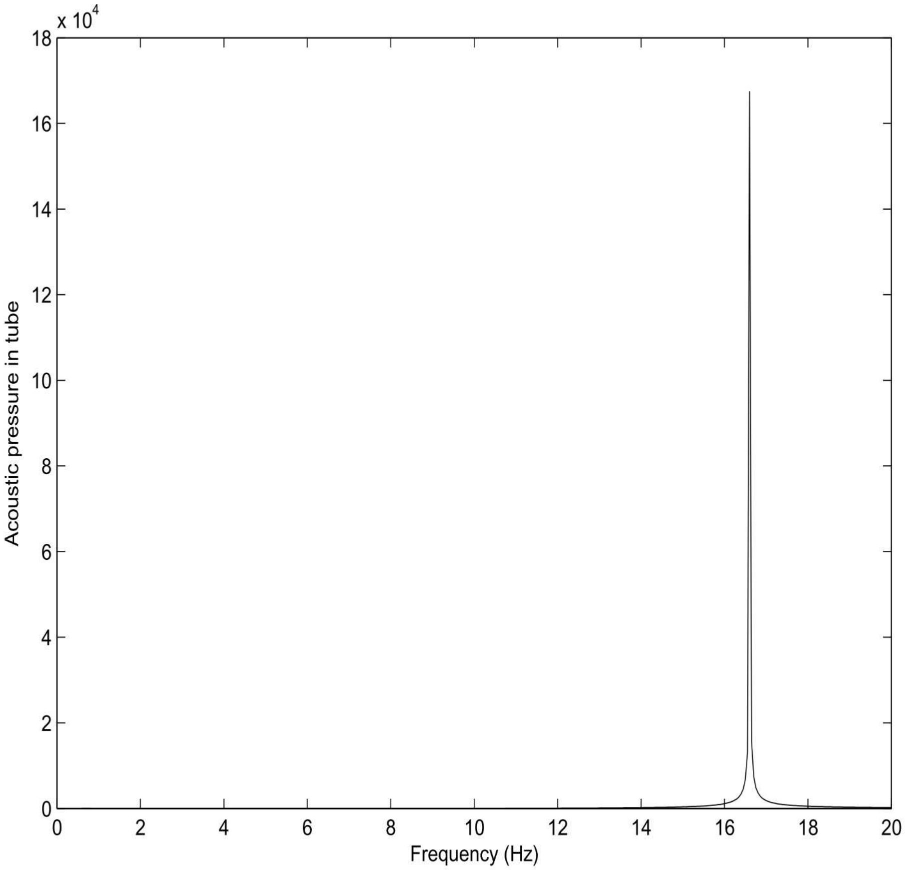


Figure 10.44: Acoustic scattering from a plate. In the case when the plate is very rigid compared with the fluid, the first resonance of the fluid tube, 16.6Hz, is reproduced quite well.

The numerical results correctly predict the first two resonances of the plate.

10.2.11 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 × 1 × 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

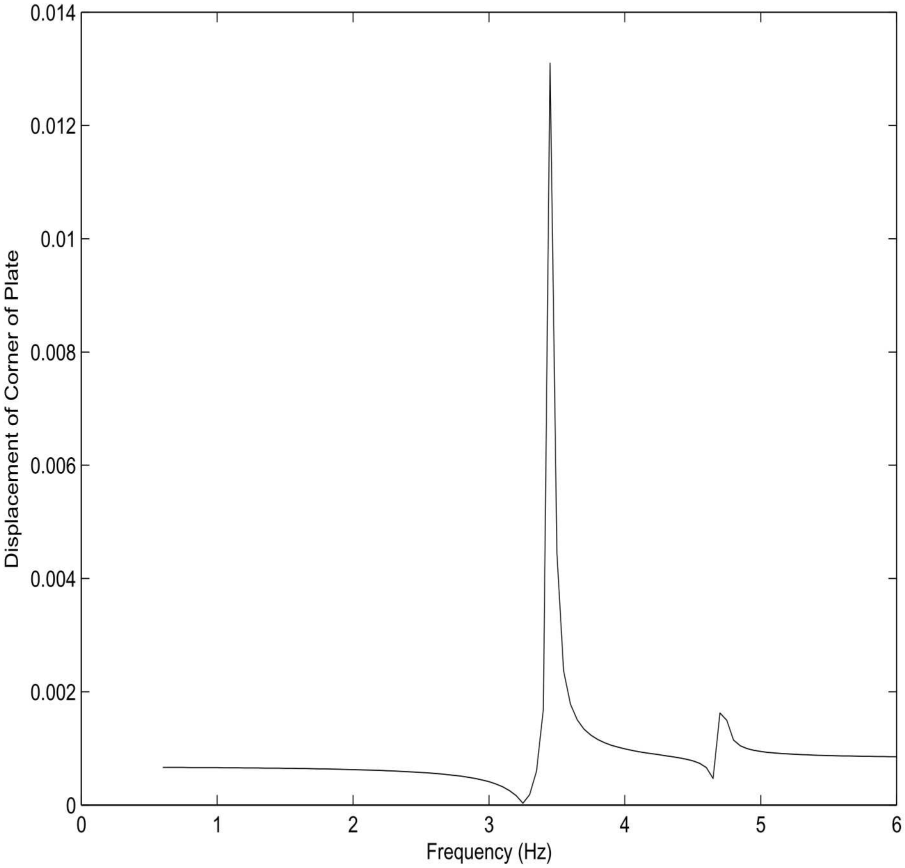


Figure 10.45: Acoustic scattering from a plate. In the case when the fluid is given a low density and high speed of sound, the first resonance of the plate appears before the acoustic tube resonances. In that case, the first two resonances of the plate, 3.5Hz and 4.7Hz, are reproduced well.

surface is given by

$$p(t) = \cos(\omega t) \tag{10.77}$$

The corresponding velocity input on the acoustic domain is given by

$$v(t) = \frac{1}{\rho c}\cos(\omega t) \tag{10.78}$$

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 10.46 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

$$a = \frac{F}{m} = \frac{2\cos(\omega t)}{770} \tag{10.79}$$

Figure 10.47 shows the comparison of the analytical and computed acceleration of the plate.

This test case can be found at

```
Salinas_rtest/verification/acoustic/hexplane.xml
```

10.2.12 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 relatively 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 greatly reduces the cost of the computation.

The theory manual details the formulation. Here we address verification of a simple sphere in an infinite medium. The example is taken from the USA LS-Dyna verification manual found in 39. 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 10.34.

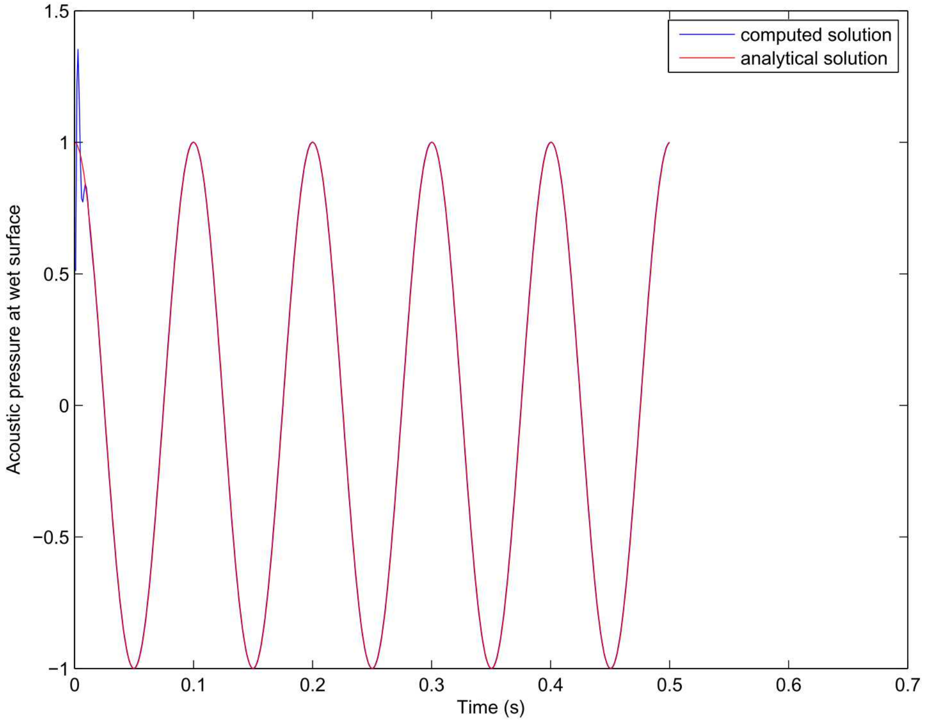


Figure 10.46: Comparison of Sierra/SD result with analytical solution of the scattered acoustic pressure for a simple 1D problem.

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 10.34: Parameters from Verification Model of Spherical Shell Subjected to Plane Step Wave

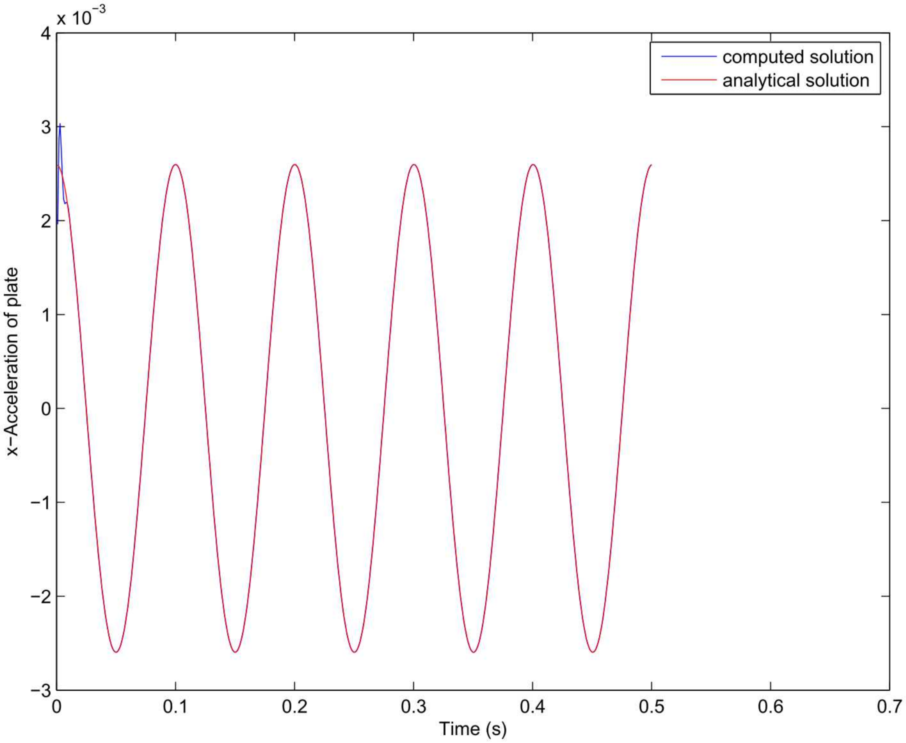


Figure 10.47: Comparison of Sierra/SD result with analytical solution of the acceleration for a simple 1D scattering problem.

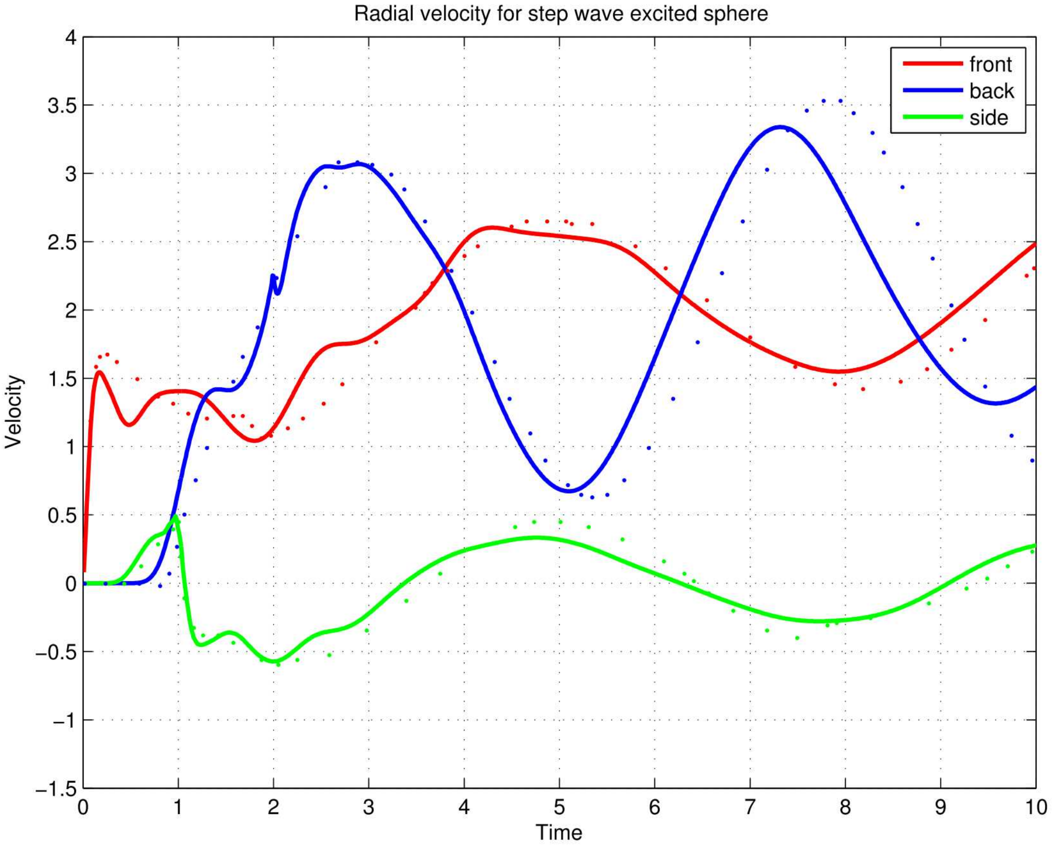


Figure 10.48: Sphere Impacted by Step Wave. The incoming step wave arrives from the $-Z$ direction. Dashed lines are the analytic solution.

The solution is shown in Figure 10.48. Clearly there are discrepancies. The FEM solution excites higher order modes not seen in the analytic solution. It is also quite likely that there are reflections from the boundaries of the fluid mesh. The verification example (found in verification/acoustic/scattering), is "quarter-sphere".

We note that the quarter-sphere model just described utilized the standard absorbing boundary condition for the exterior surface of the acoustic mesh. Identical results are obtained using infinite elements, and so we do not duplicate the plot here, but we mention that this additional test can be found in the location

```
Salinas_rtest/verification/acoustic/scattering/quarter_sphereIE.inp
```

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 10.35: Mesh Parameters of Infinite Elements on Ellipsoidal Surfaces

10.2.13 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 can be huge, while an ellipsoidal mesh greatly reduces the acoustic mesh size. To verify the behavior of the infinite elements on this boundary, we use the spherical structure of section 10.2.12 and compare with the closed form solutions obtained by Huang,[40] and referenced in the USA verification manual.[39]

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 shortcoming, the basis functions for an ellipsoidal can be defined using a variable source location, such that each element (or rather 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 10.35. All meshes are quarter symmetry models. A representative mesh is shown in Figure 10.49. Results from the analyses are shown in Figures 10.50 through 10.52.

10.2.13.0.1 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 of Huang). It contains only the first few structural modes in the solution, and thus effectively 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.

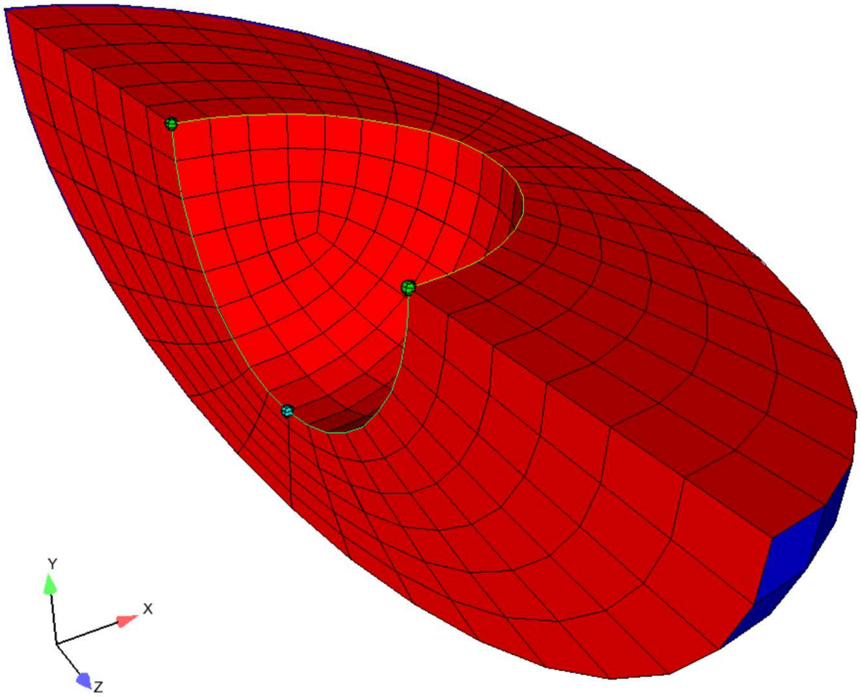


Figure 10.49: Representative Mesh of Quarter Symmetry Sphere in Ellipse

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$^{\text{TM}}$, "filtfilt" function on a Butterworth lowpass filter of order 6. The cutoff frequency is 10 kHz.

The radial response of an unfiltered and filtered responses is shown in Figure 10.53. Clearly, even with increasing mesh density, high frequency oscillations continue to dominate the response.

10.2.13.0.2 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 10.54 and 10.55 show the response for various decay factors as observed on the leading and trailing edges of the sphere. The analytical solutions for this

[1] The pressure can be written as,

$$P = H(t - \tau)\exp\left(-\beta[t - \tau]\right)$$

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.

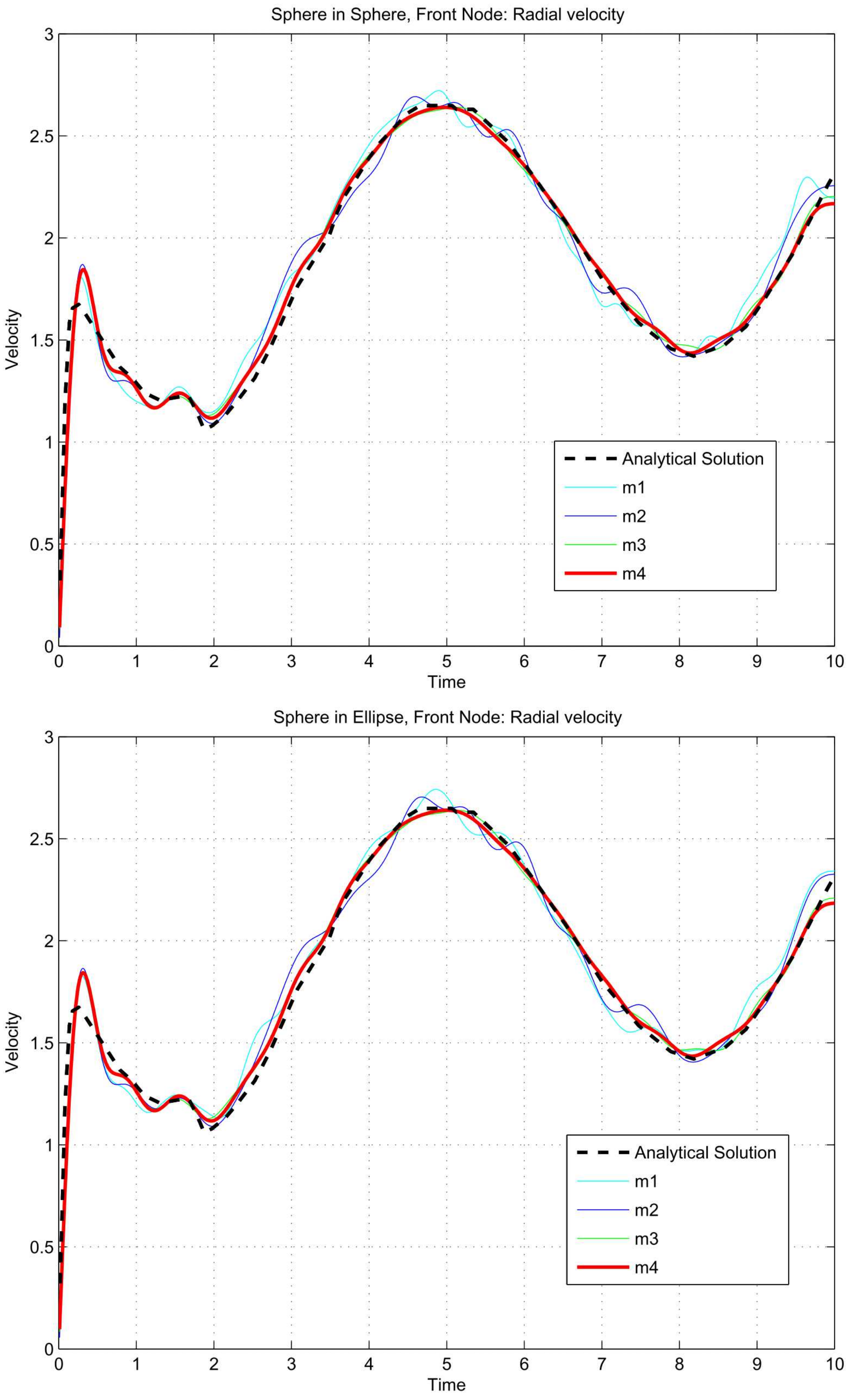


Figure 10.50: Filtered Front Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

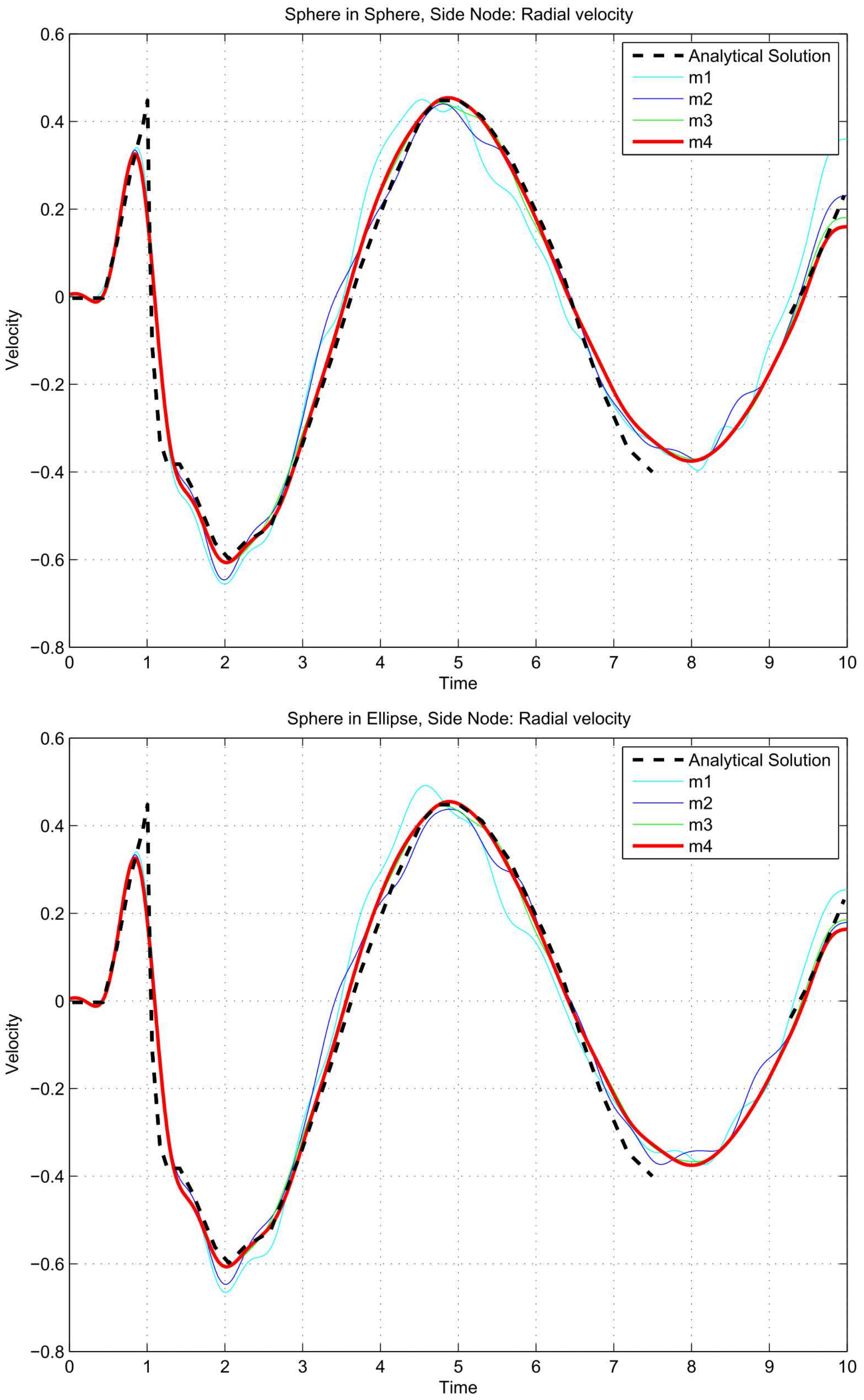


Figure 10.51: Filtered Side Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

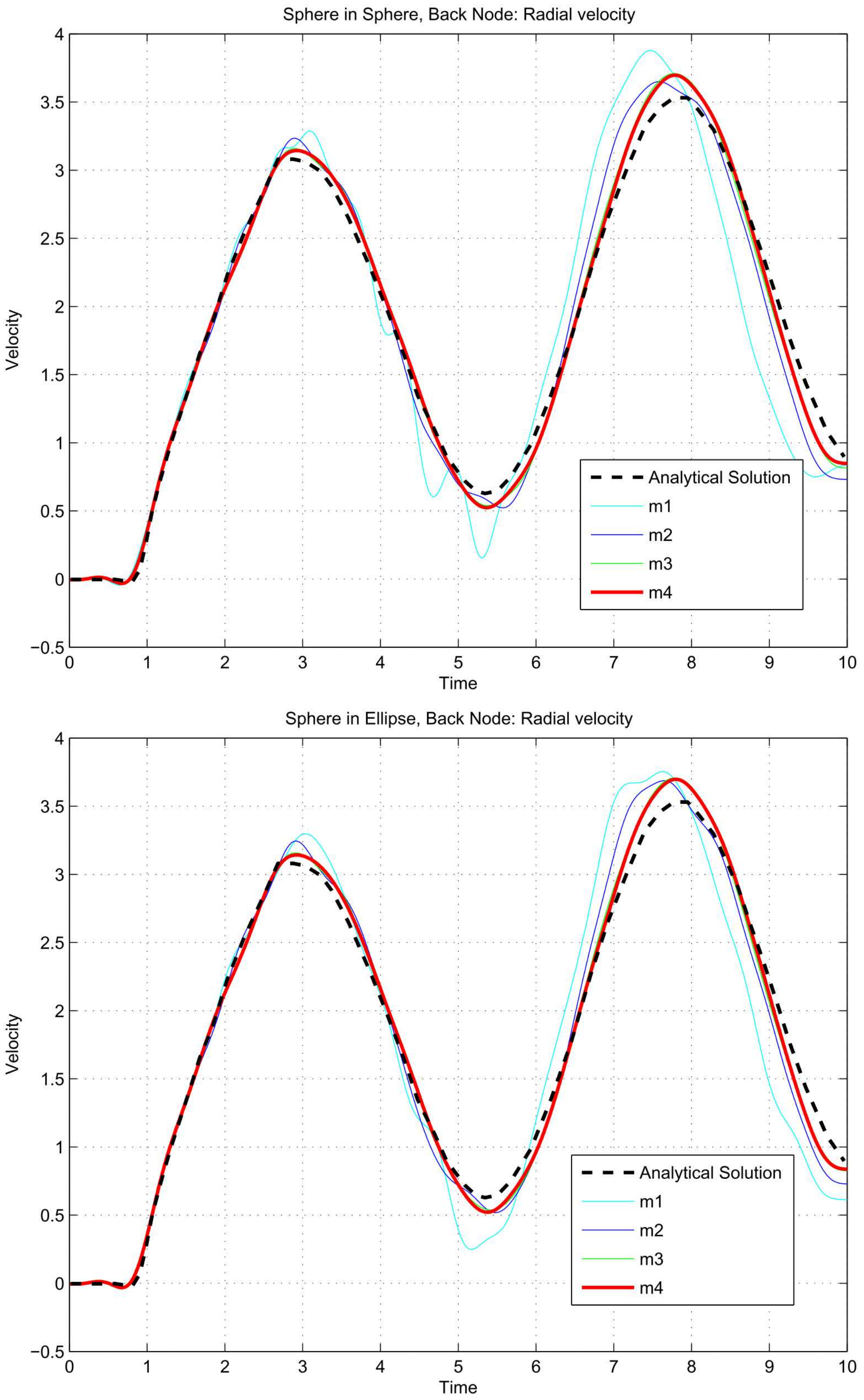


Figure 10.52: Filtered Back Node Response and mesh convergence for both a spherical and ellipsoidal acoustic region.

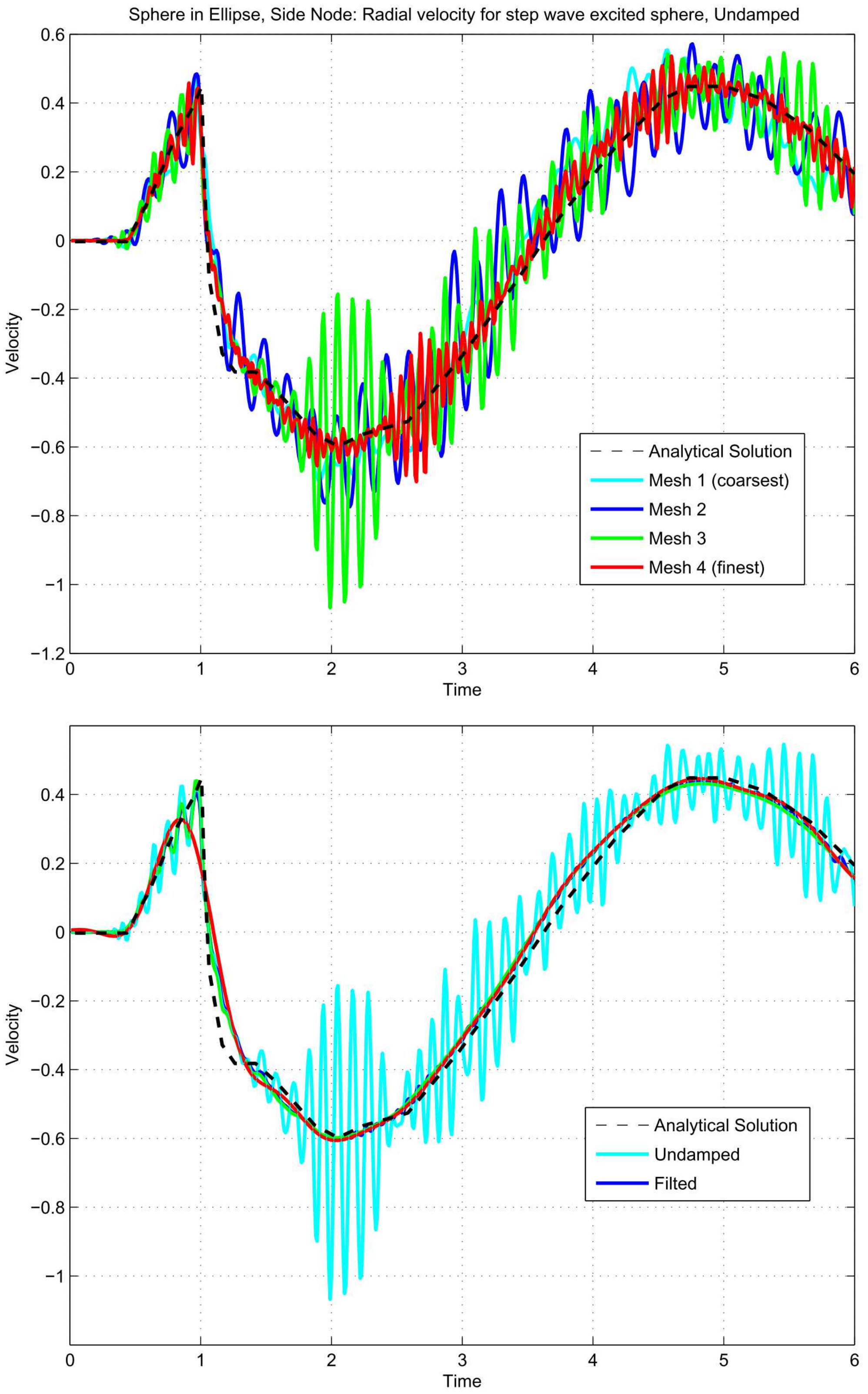


Figure 10.53: Sphere in Ellipsoid. Unfiltered response at 90^o location.

case were taken from Sprague and Geers.[41]

The purposes of these plots is to determine the dependence of the solution on the decay parameter "beta". While this dependence seems well represented in a general way, there are significant discrepancies in the plots. Notably, there is an apparent phase shift between the analytical and numerical solutions. Figure 10.56 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 completely identified at this time. The two analytical solutions should be identical, but clearly differ. We can guess that perhaps a different number of terms were retained in the series expansion. Unfortunately, the USA solution is only available for $\beta = 0$.

10.2.14 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 10.57 and 10.58 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 10.58 only shows the aspect ratio of 10 : 1. An acoustic mesh is defined on a spherical (Figure 10.57) and ellipsoidal (Figure 10.58) 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 10.59.

Figure 10.60 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 required a zero mass matrix even when the mass matrix was non-zero, as in this case. Figure 10.61 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 10.62 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,

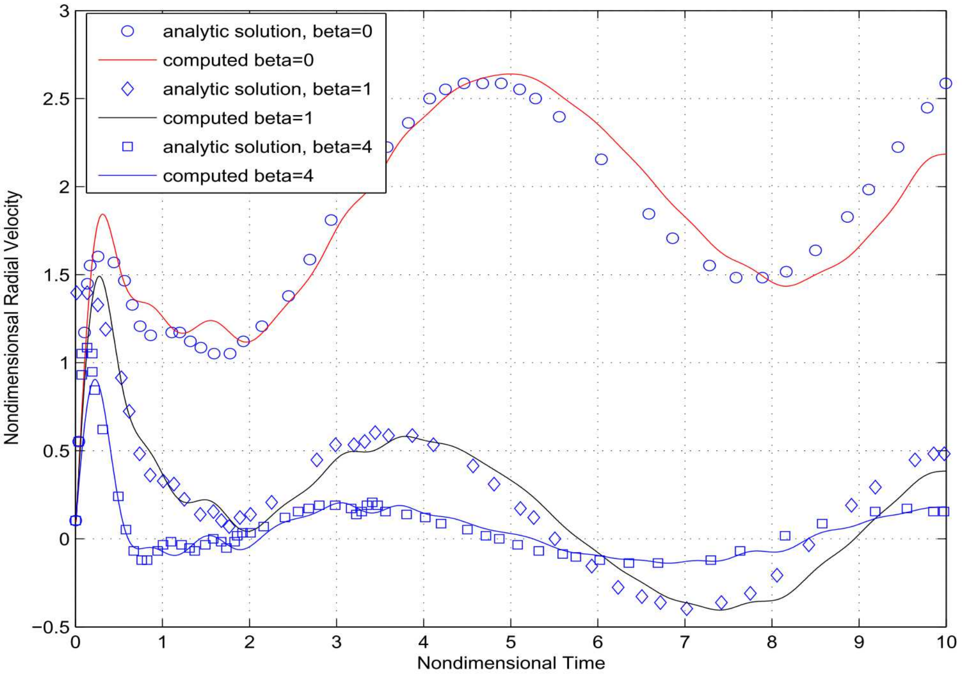


Figure 10.54: Comparison of Sierra/SD result with analytical solution of the scattered acoustic pressure on the leading surface of a sphere. Mesh=m4.

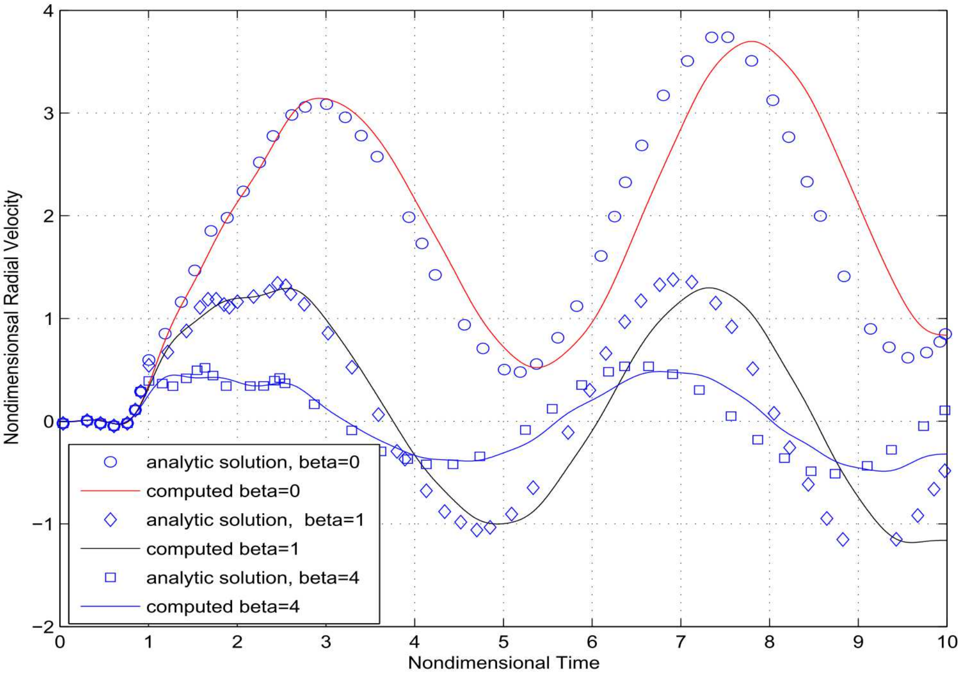


Figure 10.55: Comparison of Sierra/SD result with analytical solution of the scattered acoustic pressure on the back surface of a sphere. Mesh=m4.

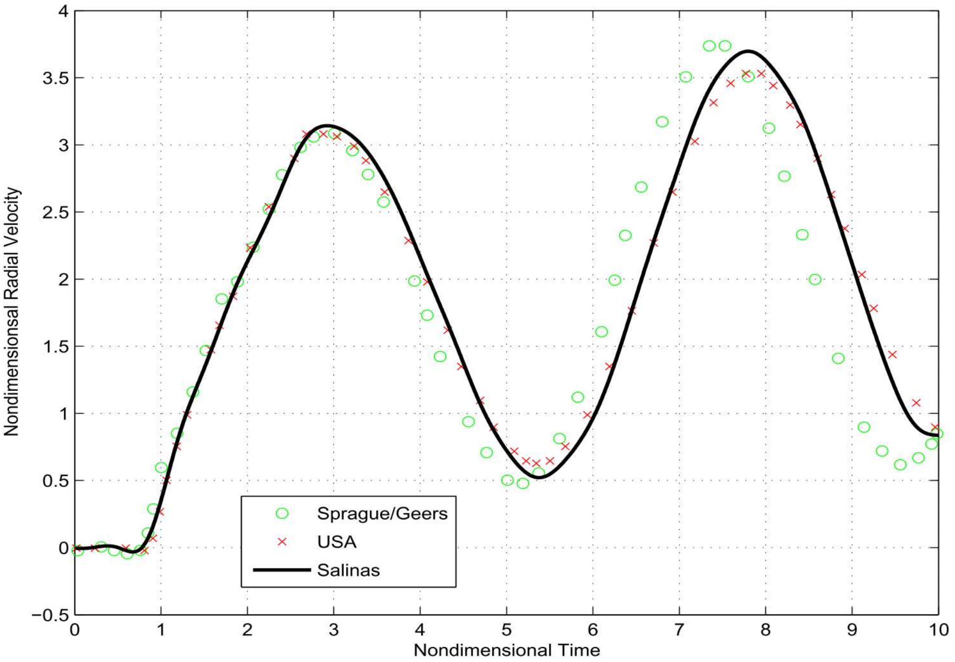


Figure 10.56: Comparison of Sierra/SD numerical result with two differing analytic solutions. Mesh=m4. Prediction on the back surface only.

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 10.63 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, and thus it is likely that one or more additional mesh refinements of the acoustic mesh are necessary to bring these results into agreement.

10.2.15 Absorbing Boundary Conditions for Infinite Elastic Spaces.

In this example we consider a perfect cube, of dimensions 1 × 1 × 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

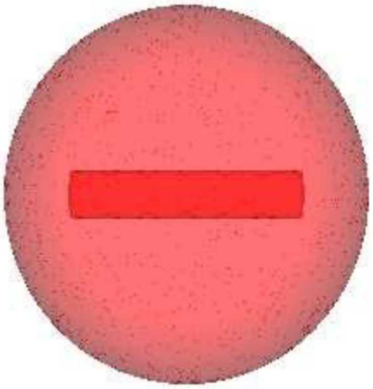


Figure 10.57: Spherical acoustic mesh for cylindrical cutout problem.

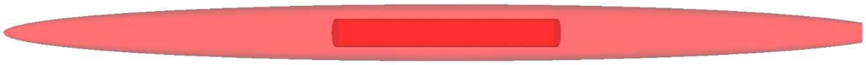


Figure 10.58: Ellipsoidal mesh with aspect ratio 10:1 for cylindrical cutout problem.

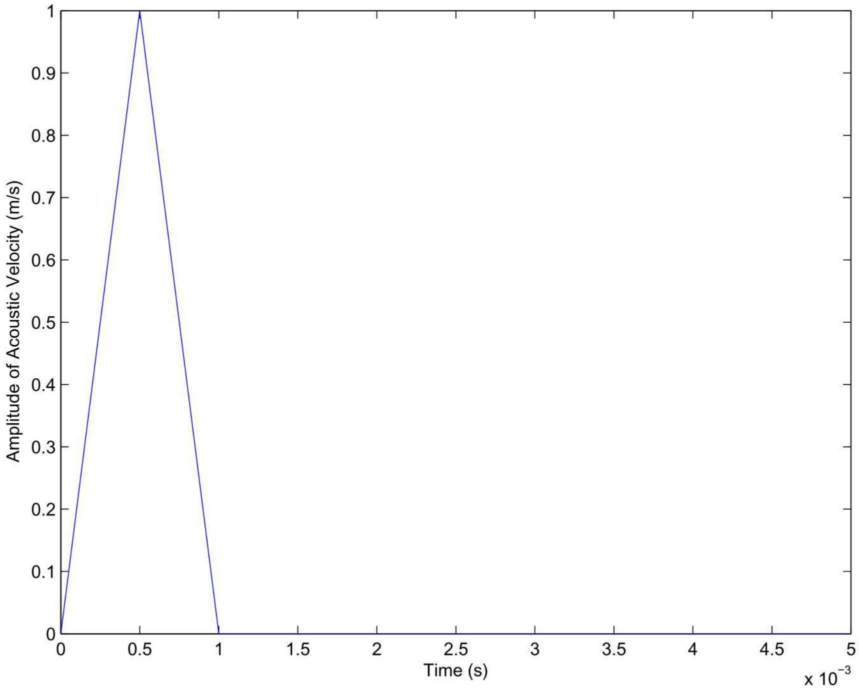


Figure 10.59: Amplitude function used to scale input acoustic velocity for cylindrical cutout problem.

```
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

$$u = \frac{P}{\omega \rho c} \tag{10.80}$$

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. Note that this solution is only valid for the infinite space, and hence will test the performance of the absorbing boundary condition for pressure waves. Figure 10.64 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

$$u = \frac{T}{\omega \rho c_s} \tag{10.81}$$

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. Note that this solution is only valid for the infinite space, and hence will test the performance of the absorbing boundary condition for shear waves. Figure 10.65 shows the comparison of this exact solution with the displacements obtained by Sierra/SD. The results are indistinguishable.

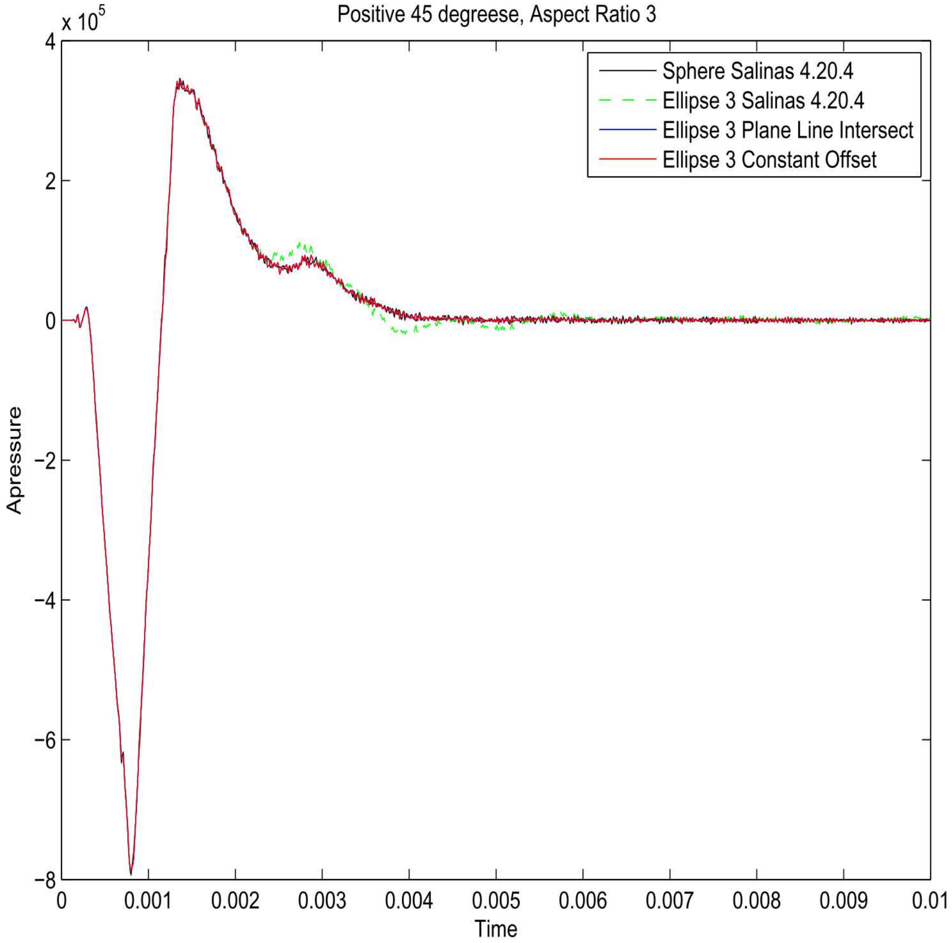


Figure 10.60: A comparison of results along a 45^o angle from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 3:1

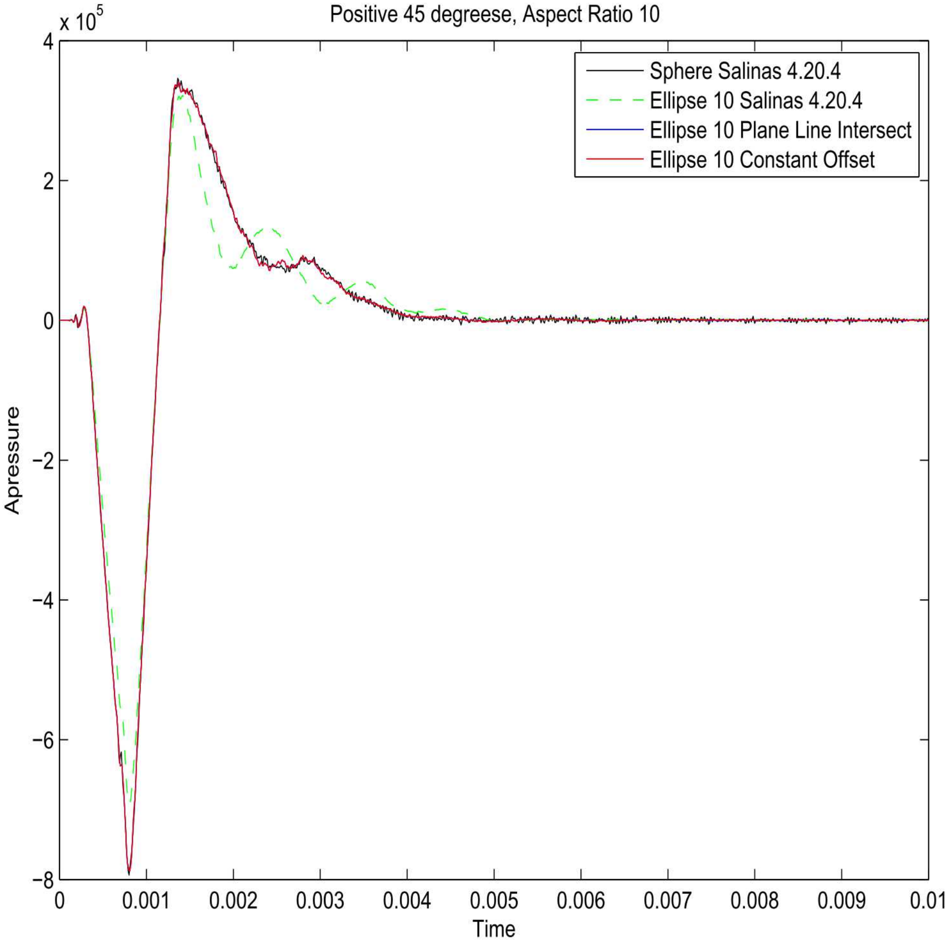


Figure 10.61: A comparison of results along a 45^o angle from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 10:1

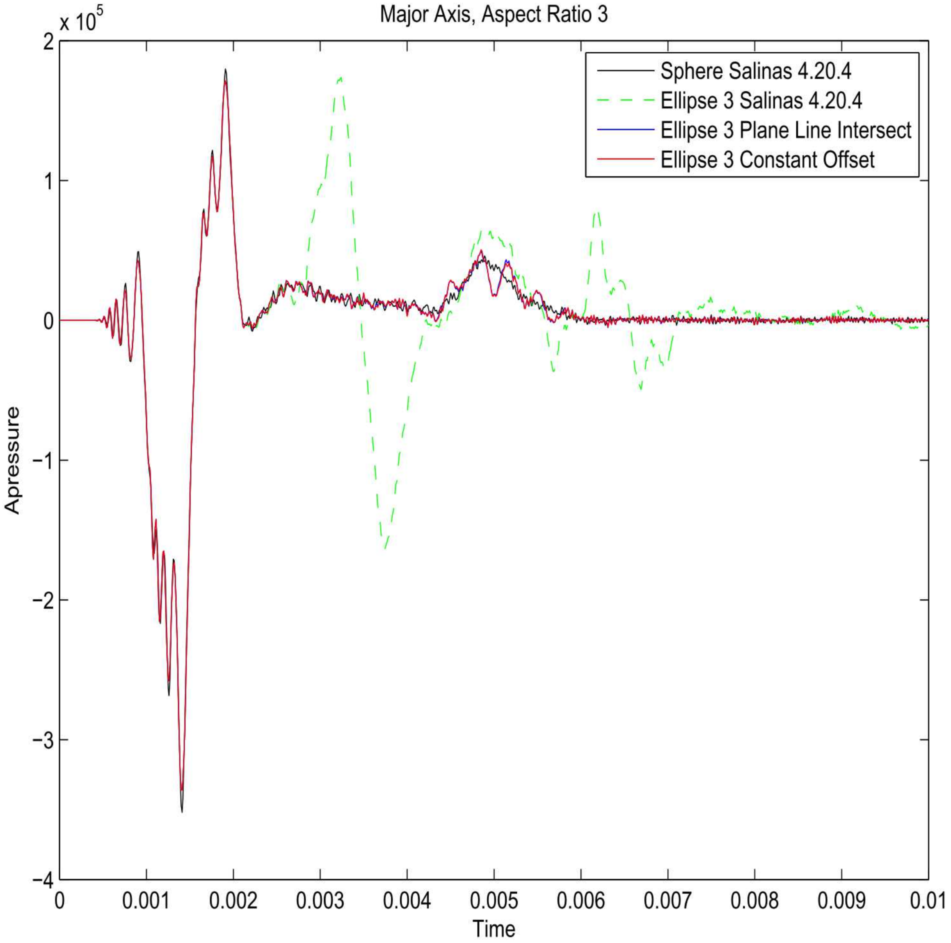


Figure 10.62: A comparison of results along the major axis from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 3:1

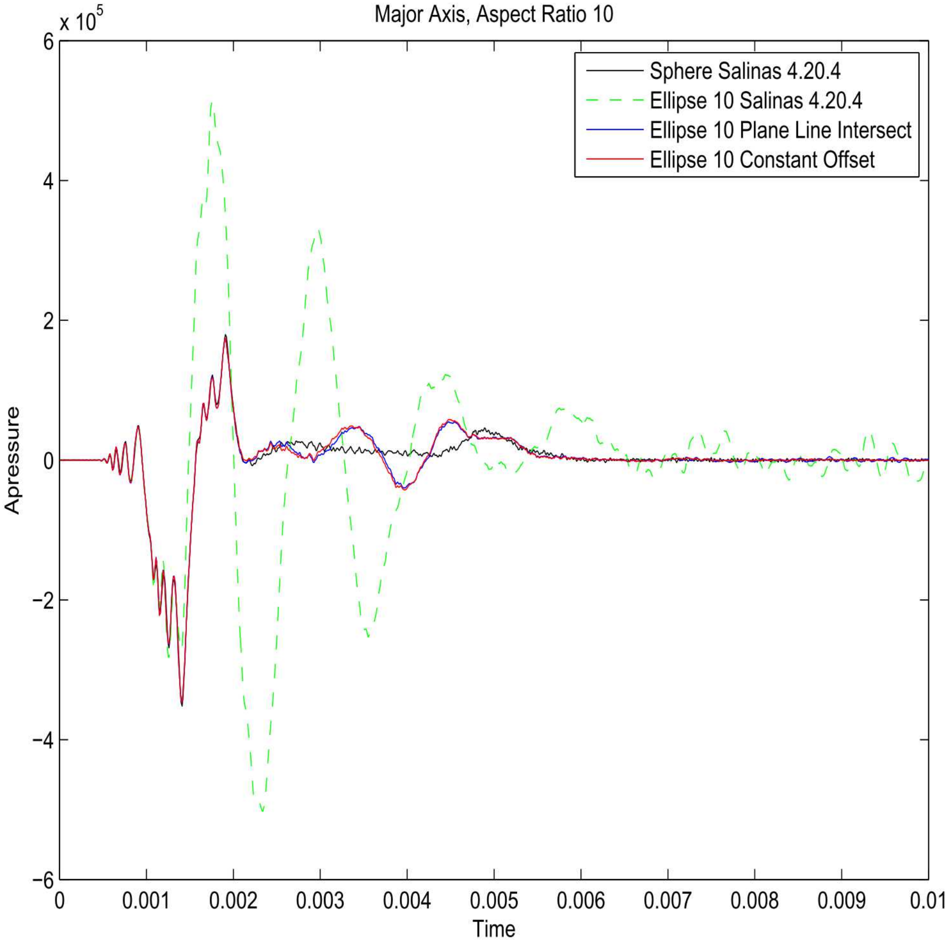


Figure 10.63: A comparison of results along the major axis from cylindrical cutout problem on spherical and ellipsoidal meshes of aspect ratios 10:1

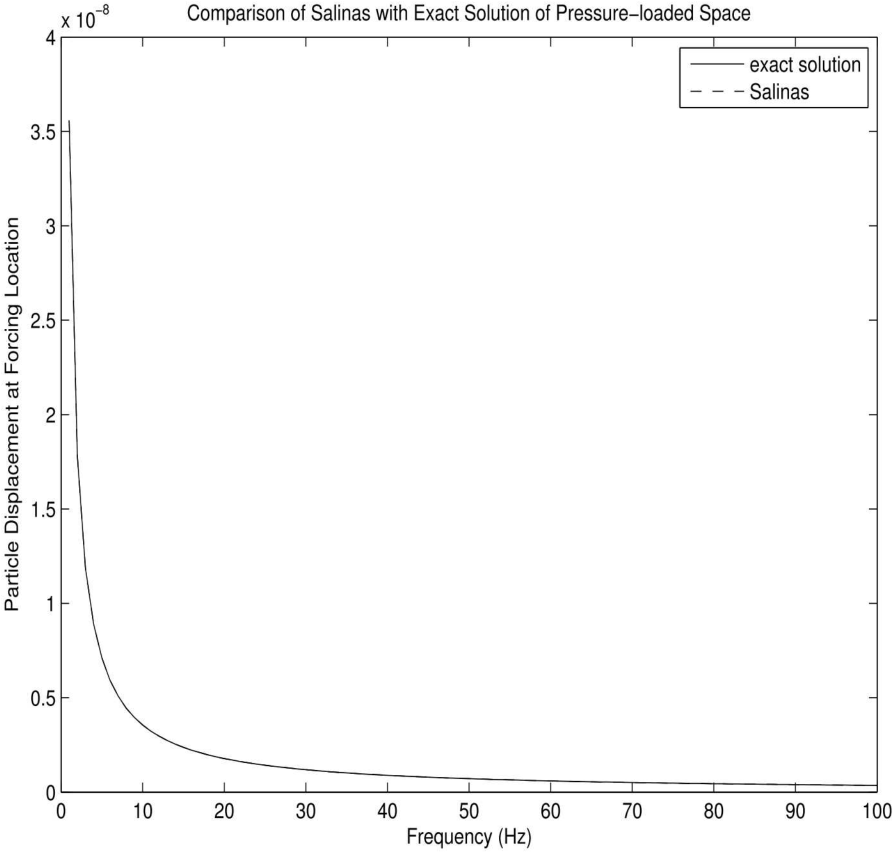


Figure 10.64: This plot shows the comparison of Sierra/SD prediction with the analytical solution of particle displacement at the forcing boundary, for a perfect cube subjected to a pressure load at one end and an absorbing boundary condition at the opposite end.

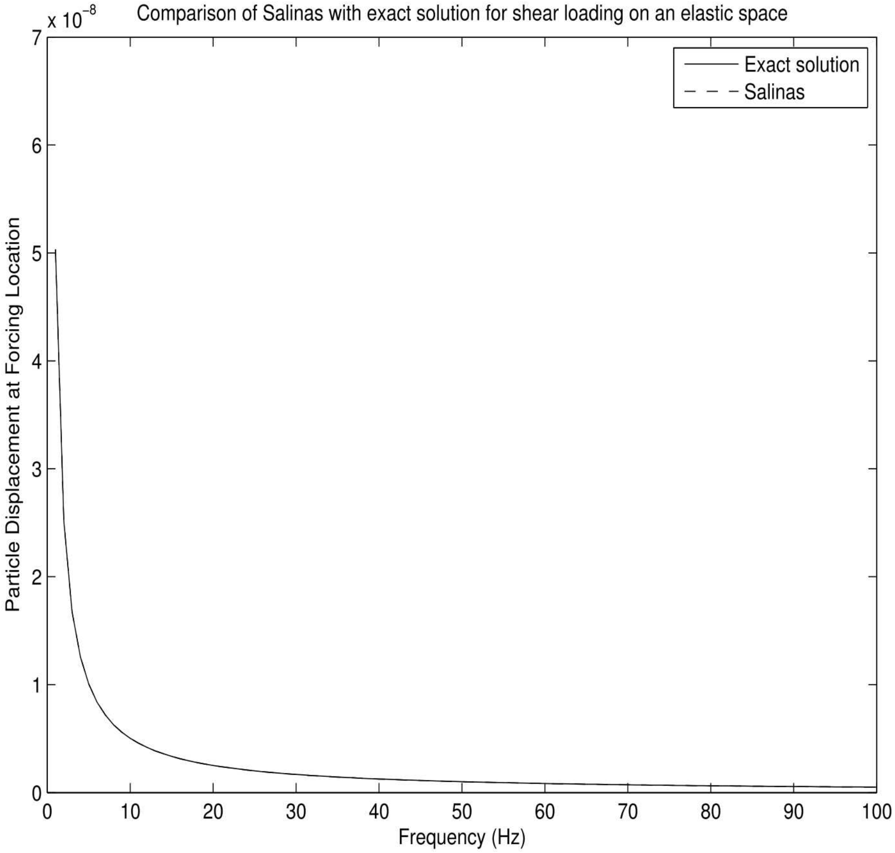


Figure 10.65: This plot shows the comparison of Sierra/SD prediction with the analytical solution of particle displacement at the forcing boundary, for a perfect cube subjected to a shear load at one end and an absorbing boundary condition at the opposite end.

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

$$P = \frac{A}{r} e^{-ikr} \tag{10.82}$$

If we prescribe the value $P = P_a$ at some value of a, as in the time-domain example described above, then we have

$$P_a = \frac{A}{a} e^{-ika} \tag{10.83}$$

This implies that $A = P_a a e^{ika}$, and thus

$$P = P_a \frac{a}{r} e^{-ik(r-a)} \tag{10.84}$$

Equation 10.84 was used to compute the far-field solution to the frequency-domain version of the

10.2.16 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[31] as

$$p = V\rho c * \frac{\frac{Z}{\rho c} + jtan(kL)}{1 + j\frac{Z}{\rho c}tan(kL)} \tag{10.85}$$

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 10.66 the Sierra/SD results against the analytic solution in equation 10.85. An excellent agreement is observed.

This example is located in the test suite at

```
Salina_rtest/verification/acoustic/waveguide_impedance.inp
```

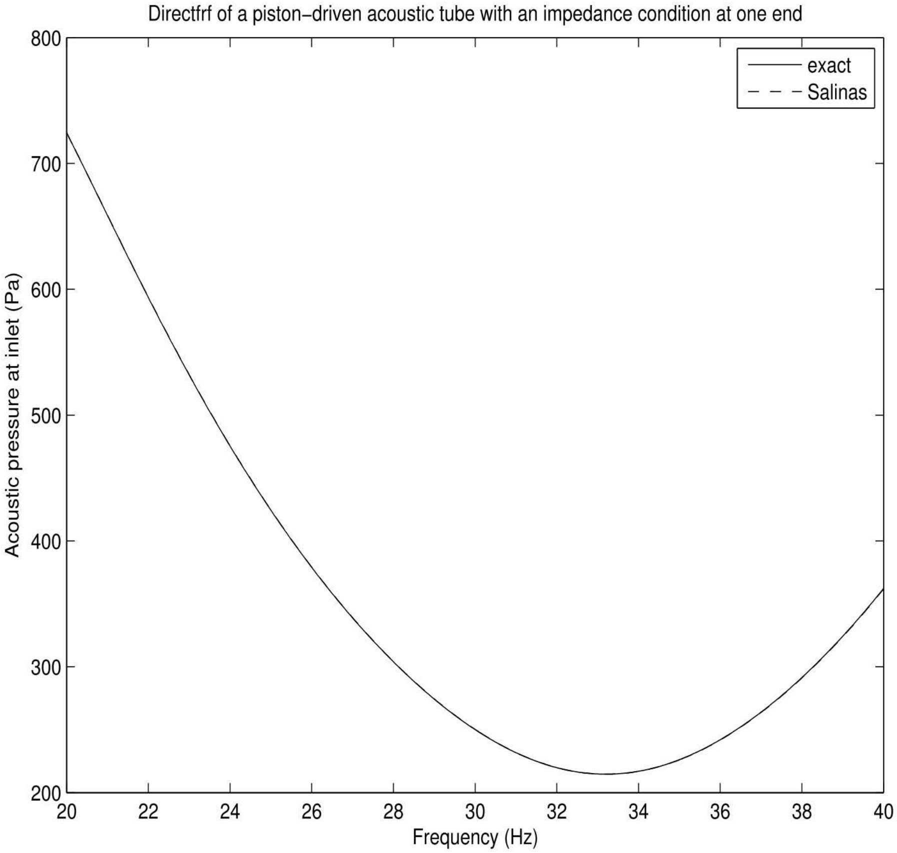


Figure 10.66: This plot shows the comparison of Sierra/SD prediction with the analytical solution of acoustic pressure, for a piston-driven acoustic wave tube with an impedance boundary condition at the opposite end.

10.2.17 Point Acoustic Source

In order to verify the acoustic point source 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, so as 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,[34] as follows

$$p(R,t) = \frac{\rho}{4\pi R}\dot{Q}(t-\frac{R}{c})H(t-\frac{R}{c}) \tag{10.86}$$

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, i.e.

$$Q = \frac{dV}{dt} \tag{10.87}$$

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 10.86 gives

$$p(R,t) = \frac{50\rho}{8}\cos(50\pi(t-\frac{2}{343})H(t-\frac{R}{c}) \tag{10.88}$$

Figure 10.67 shows a comparison of the Sierra/SD results for this problem compared against equation 10.88. 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 10.88. We note that this test can be found in

```
Salinas_rtest/verification/acoustics/point_source.inp
```

10.2.18 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[42]

$$p(R,t) = \frac{\rho}{4\pi}\frac{\dot{Q}(t-\frac{R}{c})}{R(1-M\cos\theta)^2}H(t-\frac{R}{c}) + \frac{\rho Q(t-\frac{R}{c})}{4\pi}\frac{(\cos\theta - M)V}{R^2(1-M\cos\theta)^2}H(t-\frac{R}{c}) \tag{10.89}$$

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.

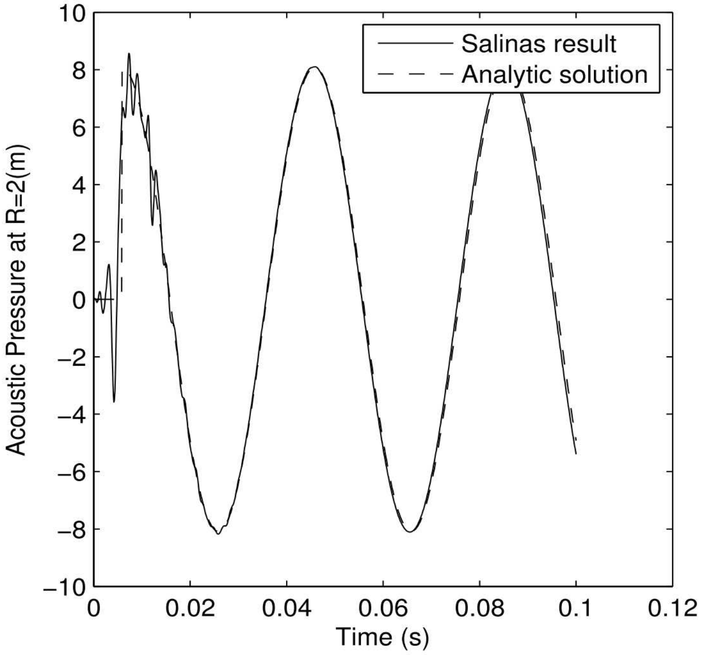


Figure 10.67: Comparison of computed and exact solution for a point source in an infinite medium.

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 10.89 is zero and equation 10.89 reduces to equation 10.86. Also, we note that equation 10.89 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,[42] but we will not consider it here.

Figure 10.68 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 centerline 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

$$Q(t) = \frac{8}{3\sqrt{3}} \left(r_0 + \Delta \sin(\omega t)\right)^3 \tag{10.90}$$

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 10.89 can be used to compute the exact solution.

Figure 10.69 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

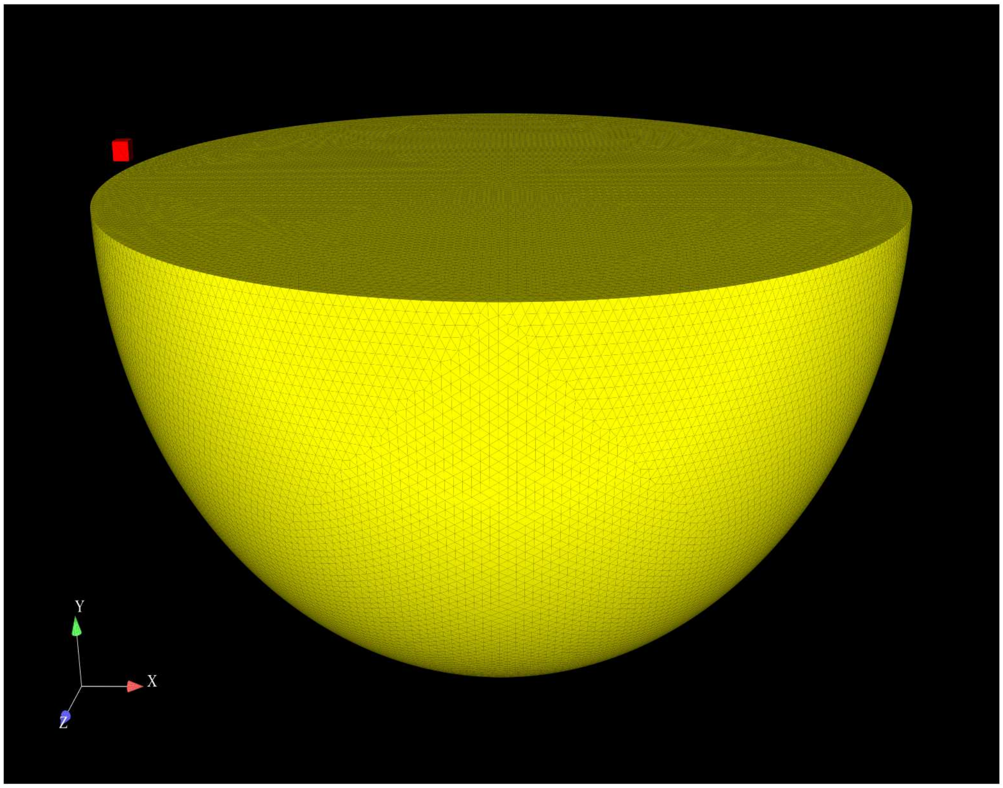


Figure 10.68: Geometry for verification example of moving point acoustic source in an infinite medium.

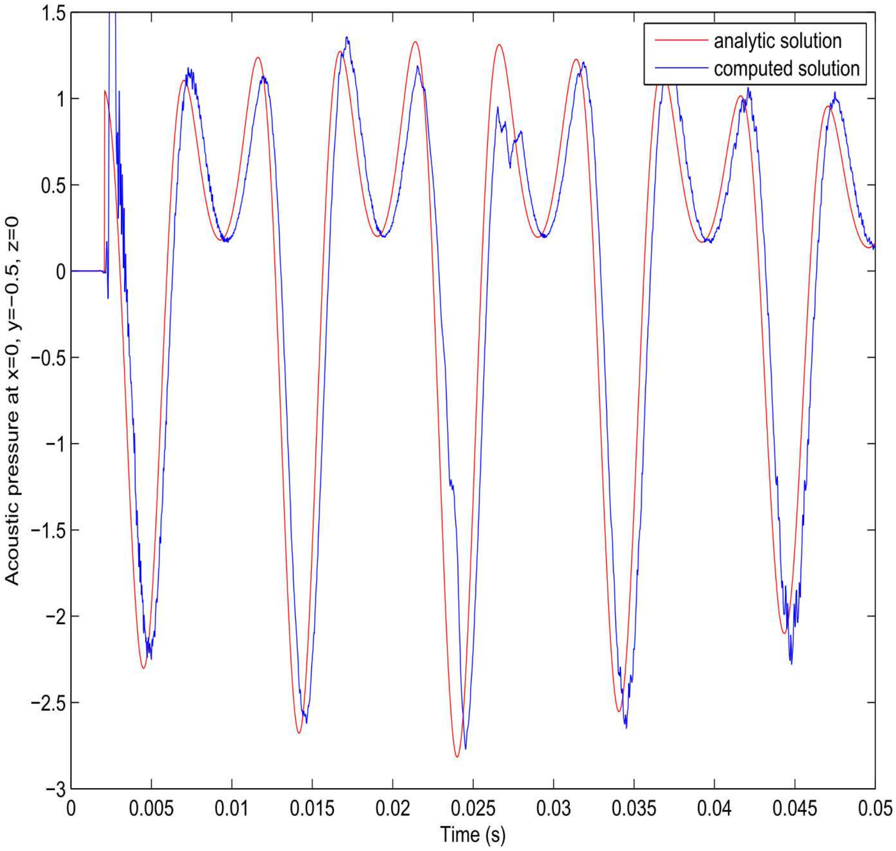


Figure 10.69: Comparison of computed and analytic solutions for verification example of moving point acoustic source in an infinite medium.

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
```

10.2.19 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[34] as

$$\phi(t) = a \int_{-\infty}^{t} e^{-(c/a)(t-\tau)} v_S(\tau) d\tau \tag{10.91}$$

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

$$p(r,t) = \frac{\rho c \dot{\phi}}{r} \tag{10.92}$$

If we define an input surface acceleration as

$$a_S(t) = \sin(2\pi t) \tag{10.93}$$

Then we have an implied input velocity of

$$v_S(t) = \frac{-1}{2\pi}\cos(2\pi t) + \frac{1}{2\pi} \tag{10.94}$$

Substituting this into equation 10.91, we obtain

$$\phi = \frac{-a}{2\pi} \int_{-\infty}^{t} e^{-(c/a)(t-\tau)} \left[\frac{-1}{2\pi}\cos(2\pi t) + \frac{1}{2\pi}\right] d\tau \tag{10.95}$$

Simplifying, and using the identity

$$\int e^{c_1 x}\cos(c_2 x) = \frac{e^{c_1 x}}{c_1^2 + c_2^2}\left(c_1\cos(c_1 x) + c_2\sin(c_2 x)\right) \tag{10.96}$$

we obtain

$$\phi(t) = \frac{-a}{2\pi}\frac{1}{(\frac{c}{a})^2 + (2\pi)^2}\left[\frac{c}{a}\cos(2\pi t) + (2\pi)^2\sin(2\pi t)\right] + \frac{2\pi a^2}{c\left(\frac{c}{a}\right)^2 + c(2\pi)^2} e^{\frac{-ct}{R}} \tag{10.97}$$

Inserting this expression into equation 10.92, we obtain the exact solution on the surface of the sphere (R=a)

$$p(r,t) = \frac{\rho c}{\left((\frac{c}{a})^2 + (2\pi)^2\right)}\left[2\pi e^{\frac{-ct}{a}} + \frac{c}{a}\sin(2\pi t) - 2\pi\cos(2\pi t)\right] \tag{10.98}$$

We note that there is both a transient and a steady-state component to the solution in equation 10.98. The transient term dies out after sufficient time, and then the steady terms persist.

Figure 10.70 shows the comparison of the exact solution of equation 10.98 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 10.71 shows a schematic of the geometry. A 3D hemispherical domain of radius $0.5(m)$ was constructed and meshed with tetrahedral finite elements. A normal

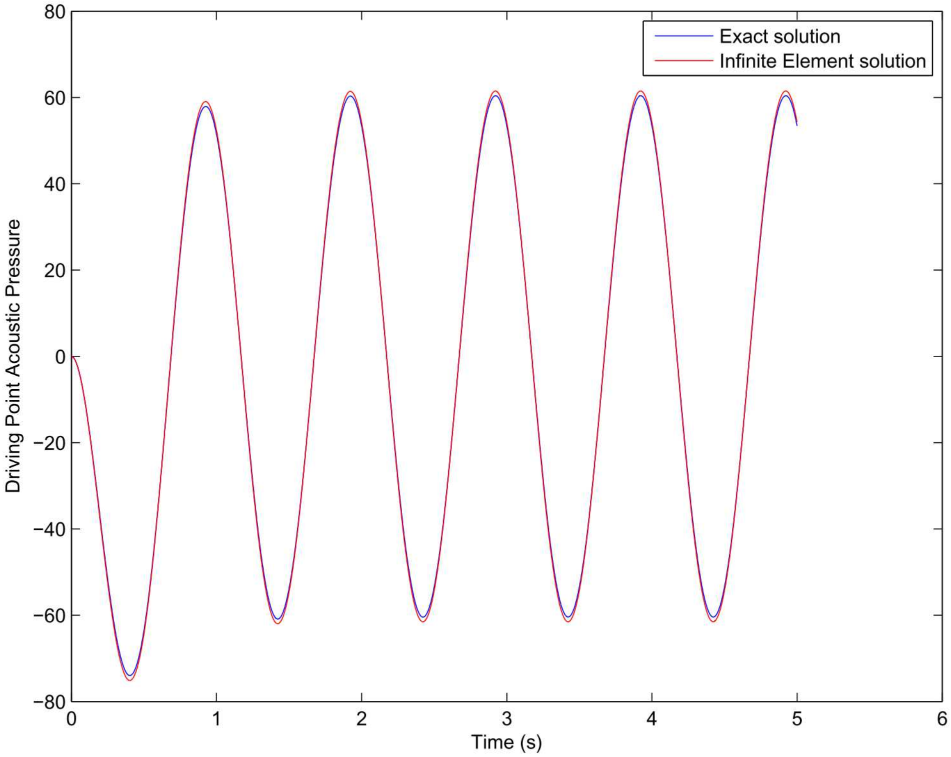


Figure 10.70: A comparison of an exact solution for spherical wave radiation and the Sierra/SD computation using transient infinite elements.

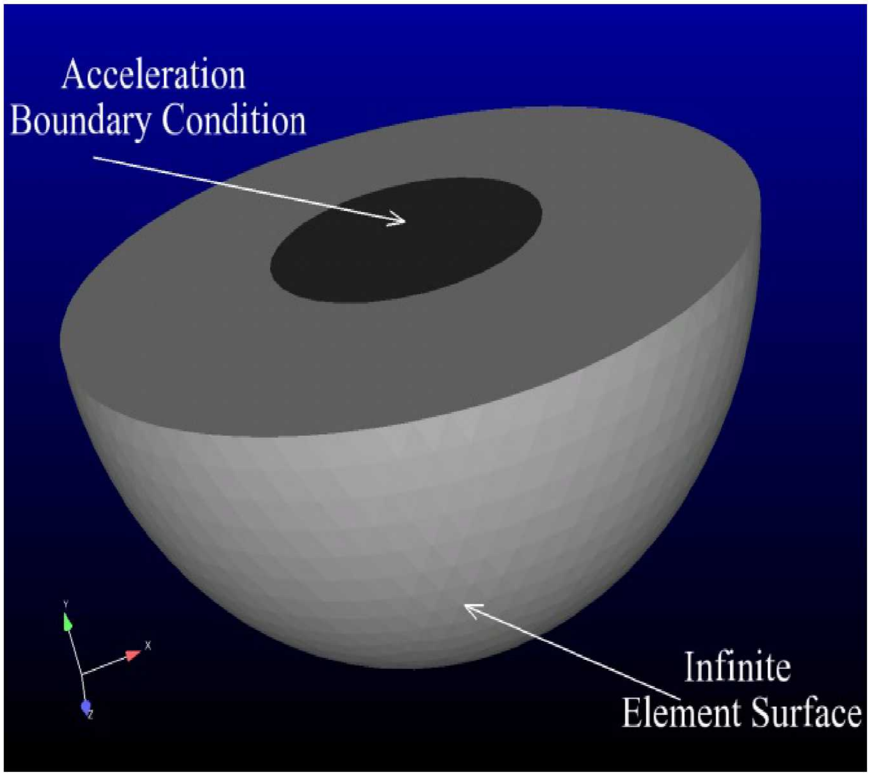


Figure 10.71: A schematic of the geometry of a piston mounted on an infinite baffle for verification of transient infinite elements.

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 10.71. 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[34]

$$p(x,t) = \frac{\rho}{2\pi} \int_S \frac{a_n(x_s, y_x, t - R/c)}{R} dS \tag{10.99}$$

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, $R = \sqrt{[(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2]}$ 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 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 10.99 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 10.72 and 10.73 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.

10.2.20 Variable Order Infinite Element Implementation

Before making comparisons of the infinite element and Kirchoff 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 10.74 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

$$p(\boldsymbol{x},t) = \frac{\rho}{2\pi} \int_S \frac{a_P(\boldsymbol{x}_{\boldsymbol{S}}, t - \frac{R}{c})}{R} dS \tag{10.100}$$

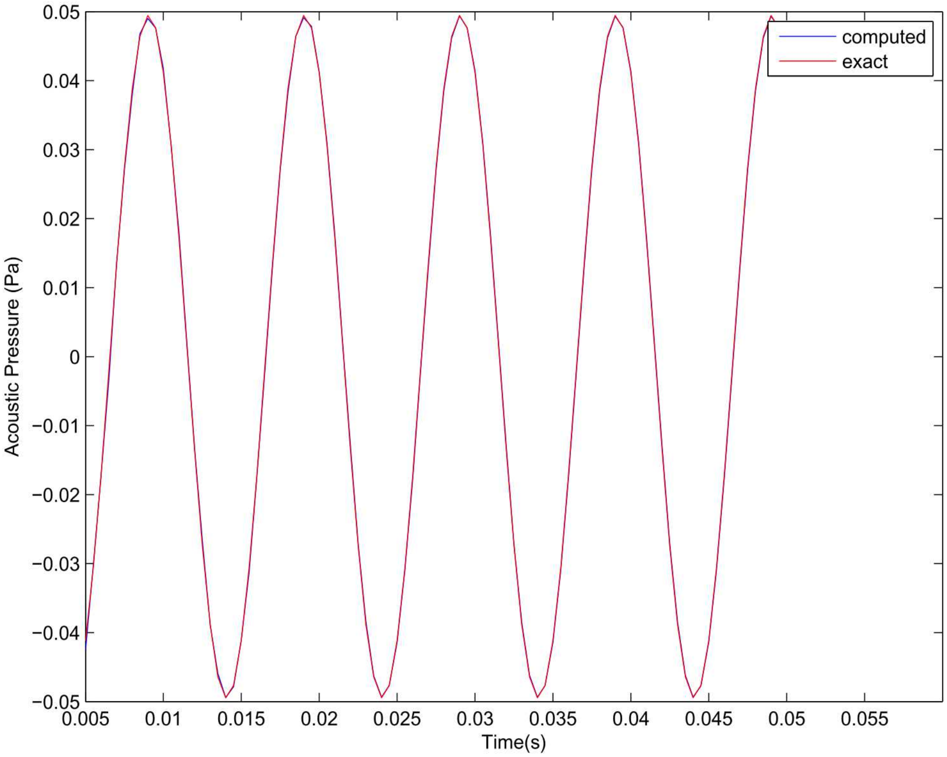


Figure 10.72: A comparison of computed vs. analytic solution for a piston mounted on an infinite baffle. Field point is at $x = 0$, $y = -0.5$, $z = 0$.

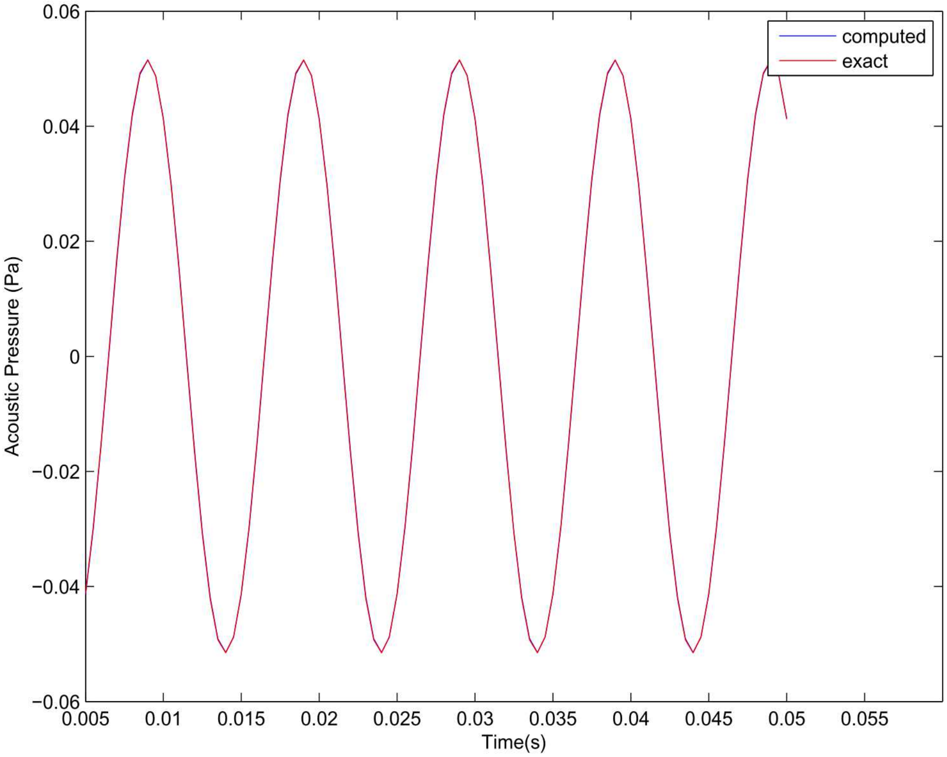


Figure 10.73: A comparison of computed vs. analytic solution for a piston mounted on an infinite baffle. Field point is at $x = 0.5$, $y = 0.0$, $z = 0$.

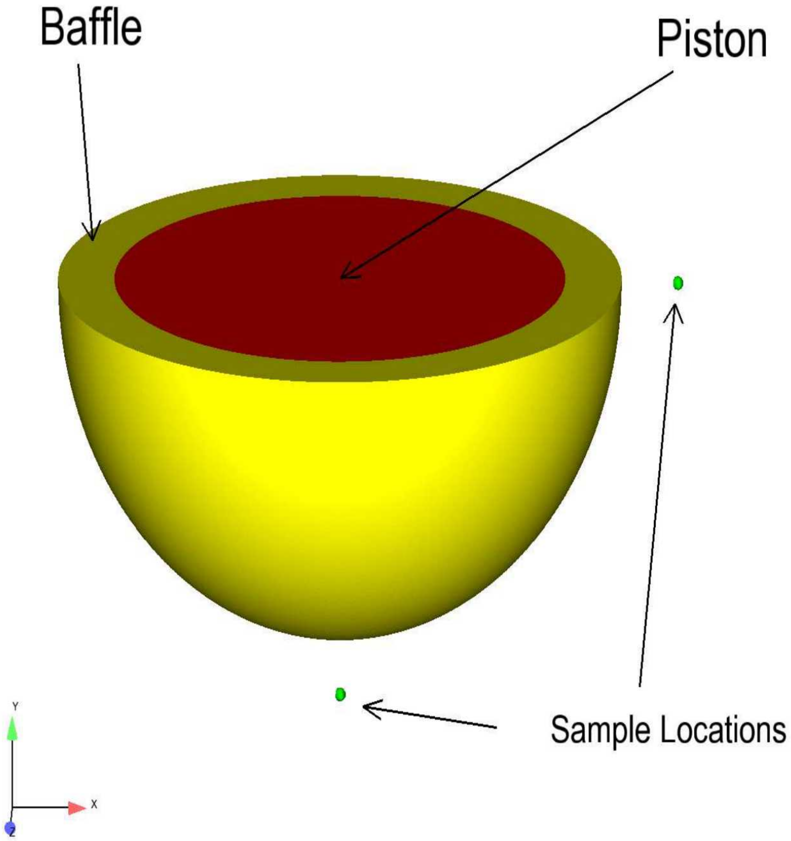


Figure 10.74: The geometry and mesh of the baffled piston problem.

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 10.100 was carried out numerically, and this provided the exact solution for comparison with the computations.

In all of 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 5.0x10^{-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 10.75 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 $a_P(t) = \sin(2\pi ft)H(t)$, $f = 2000$(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.

10.2.21 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 10.76. The pressure contours for both the Sierra/SD and ABAQUS outputs are shown in Figures 10.78 and 10.77, respectively, while a comparison of peak values are shown in Table 10.36.

	Peak Pressure (Pa)	Peak Uy (m)
ABAQUS	-10811.5	1.031e-6
Sierra/SD	-10818.16	-1.030e-6

Table 10.36: . Peak pressure and displacement for coupled acoustic-structure interaction problem with viscoelastic material.

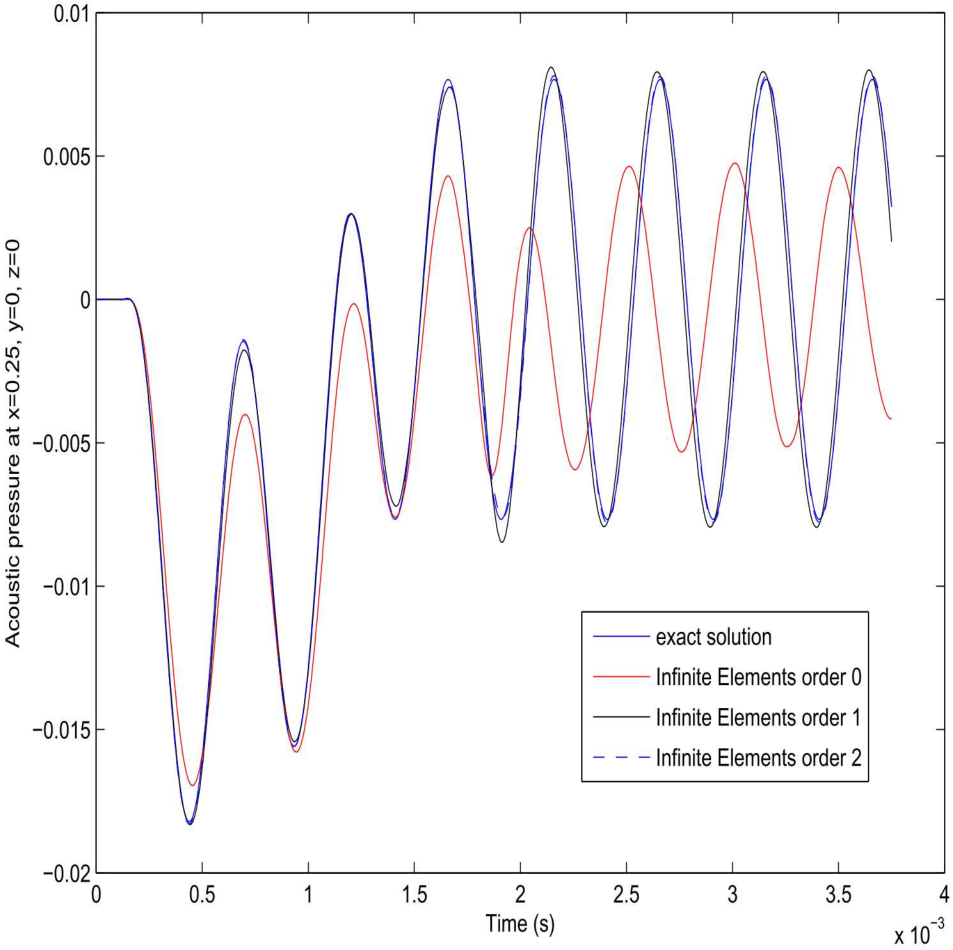


Figure 10.75: A convergence study for infinite element order, demonstrated on the baffled piston problem

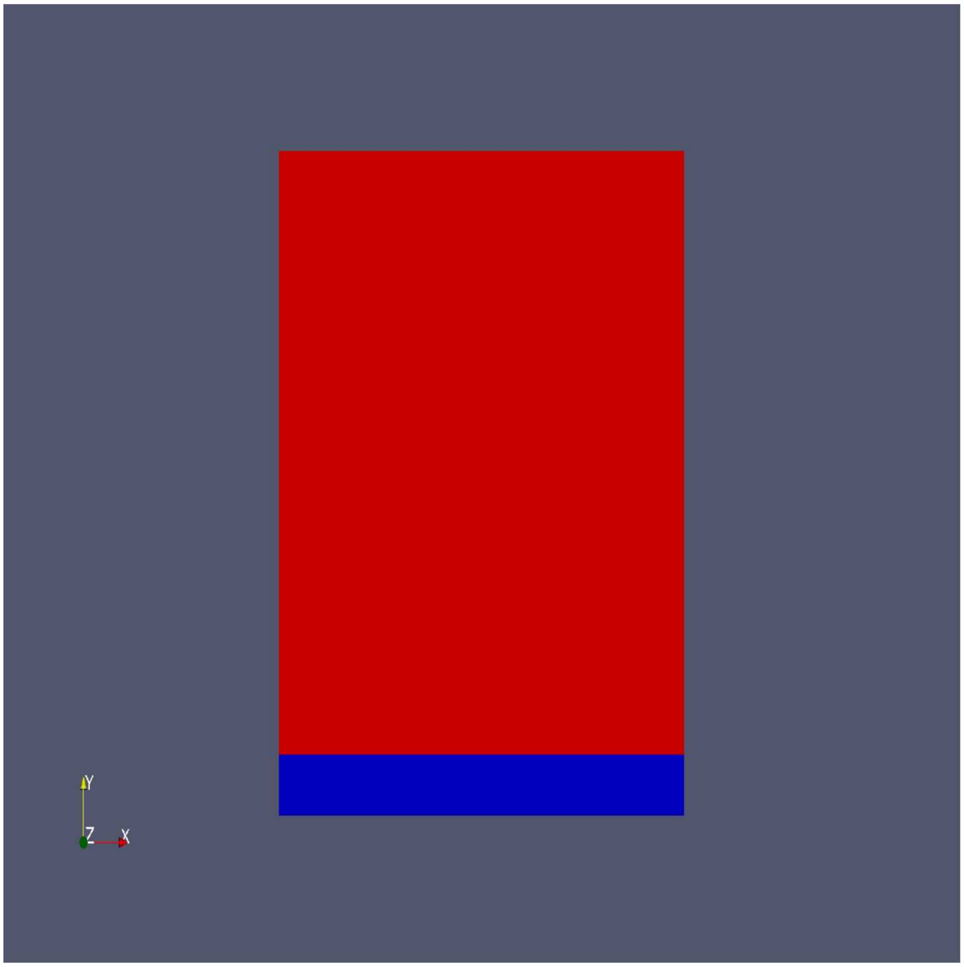


Figure 10.76: Problem sketch. The bottom part is the solid, the top part is the fluid

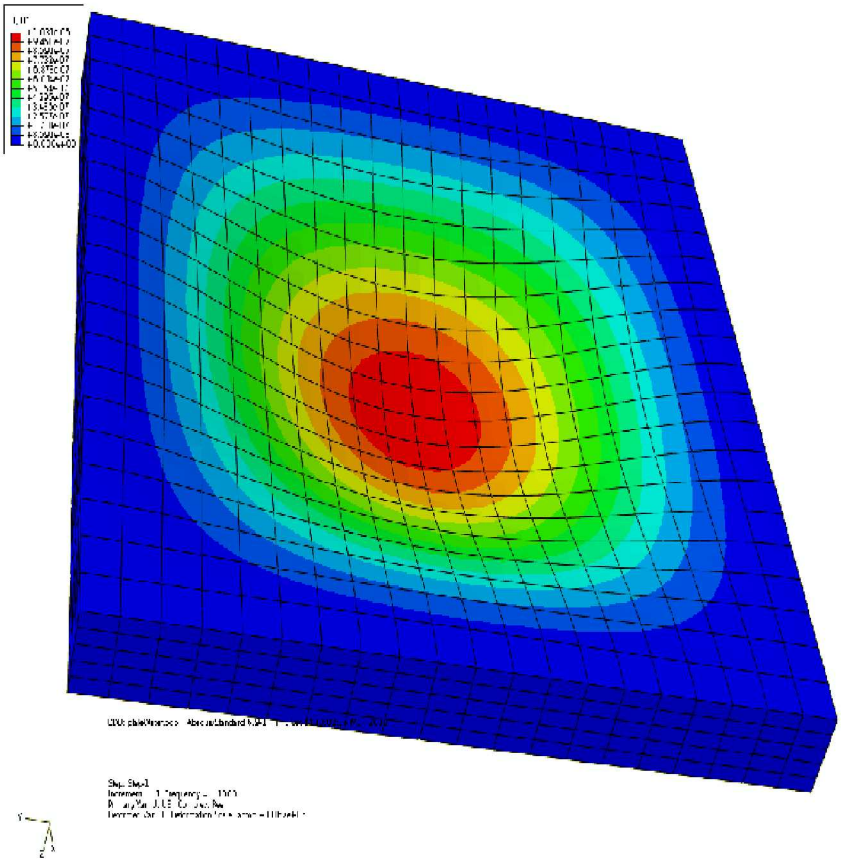


Figure 10.77: Vertical displacement distribution from ABAQUS.

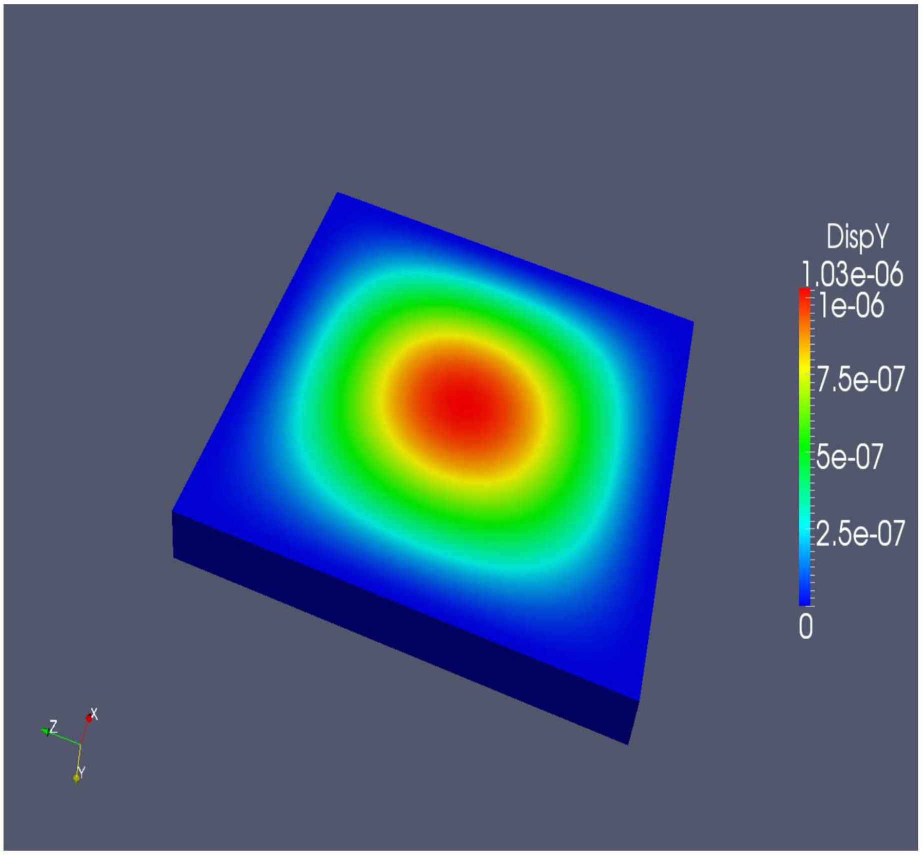


Figure 10.78: Vertical displacement distribution from Sierra/SD.

10.3 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[43,44] involving a piston-radiation problem. This example is shown in Figure 10.79. 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

$$p(0,t) = p_0 \sin(\omega t) \tag{10.101}$$

$$v(0,t) = v_0 \sin(\omega t) \tag{10.102}$$

In order 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[34]) in the pre-shock regime and by the Fay solution in the post-shock regime.

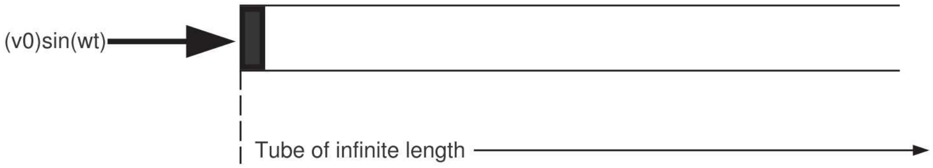


Figure 10.79: A wave tube example for verification

In the case of a plane wave, the distance to shock formation is given as

$$\sigma = \frac{c}{\left(1 + \frac{B/A}{2}\right) v_0 k} \tag{10.103}$$

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[45,46] is given by

$$p(x,t) = p_0 \sum_{n=1}^{\infty} \frac{2}{n\bar{x}} J_n(n\bar{x}) \sin(n\omega\tau) \tag{10.104}$$

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

$$p(x,t) = p_0 \frac{2}{\Gamma} \sum_{n=1}^{\infty} \frac{\sin(n\omega\tau)}{\sinh\left[n(1+\sigma)\Gamma)\right]} \tag{10.105}$$

where Γ is the ratio of the absorption length to the shock formation distance (see[46]). The Fubini solution assumes a lossless media, and thus it is only 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[47] that provide exact solutions for $\sigma < x < 3.5\sigma$, but we do not consider those here.

For all of the results presented next, the fluid is air at ambient conditions, with $c = 332.0\frac{m}{s}$, $\rho = 1.293\frac{Kg}{m^3}$. Also, if we only account for viscosity and thermal conductivity as loss mechanisms, the absorption parameter can be calculated from the following equation[34]

$$\begin{aligned}
\frac{b}{c^2} &= \frac{1}{\rho c^2}\left[\frac{4}{3}\eta + (\gamma - 1)\frac{k}{C_p}\right] && (10.106)\\
&= \frac{1}{1.293x332^2}\left[\frac{4}{3}1.846e^{-5} + (0.4)\frac{2.624e^{-2}}{1000}\right] && (10.107)\\
&= 7.017e^{-6}x\left[2.461e^{-5} + 1.0496e^{-5}\right] = 2.46e^{-10} && (10.108)\\
& && (10.109)
\end{aligned}$$

For air, $\frac{b}{c^2}$ is so small that it was found to have virtually no effect on the numerical results. Note that this estimate neglects additional loss mechanisms such as molecular relaxation, and wall losses.

Figures 10.80, 10.81, and 10.82 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 10.80, 10.81, and 10.82, respectively.

In order to demonstrate the significant difference between linear and nonlinear solutions, in Figure 10.83 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.

$$\frac{|Res_f|}{|F_{ext}|} \leq 10^{-6} \qquad (10.110)$$

Also, we mention that the starting point for the Newton iterations is the value of velocity potential from the previous time step. Figure 10.84 shows the number of Newton iterations required to satisfy the inequality 10.110, 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

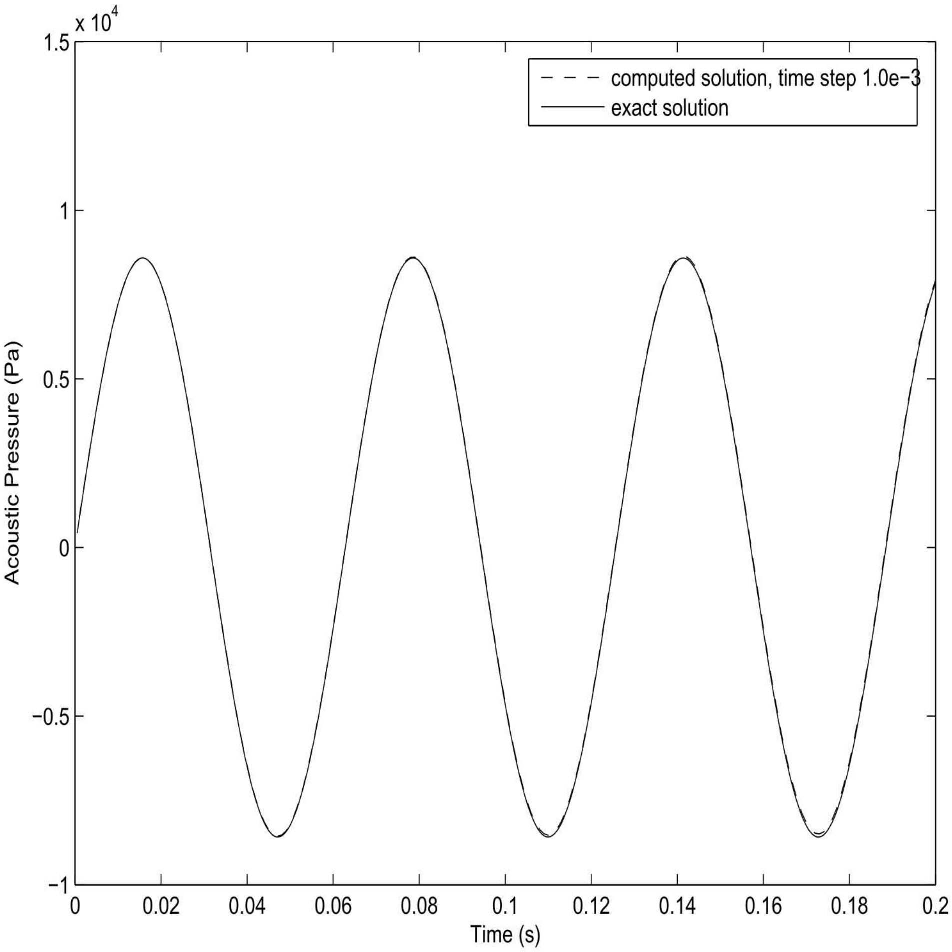


Figure 10.80: Acoustic radiated pressure at $x = 0$

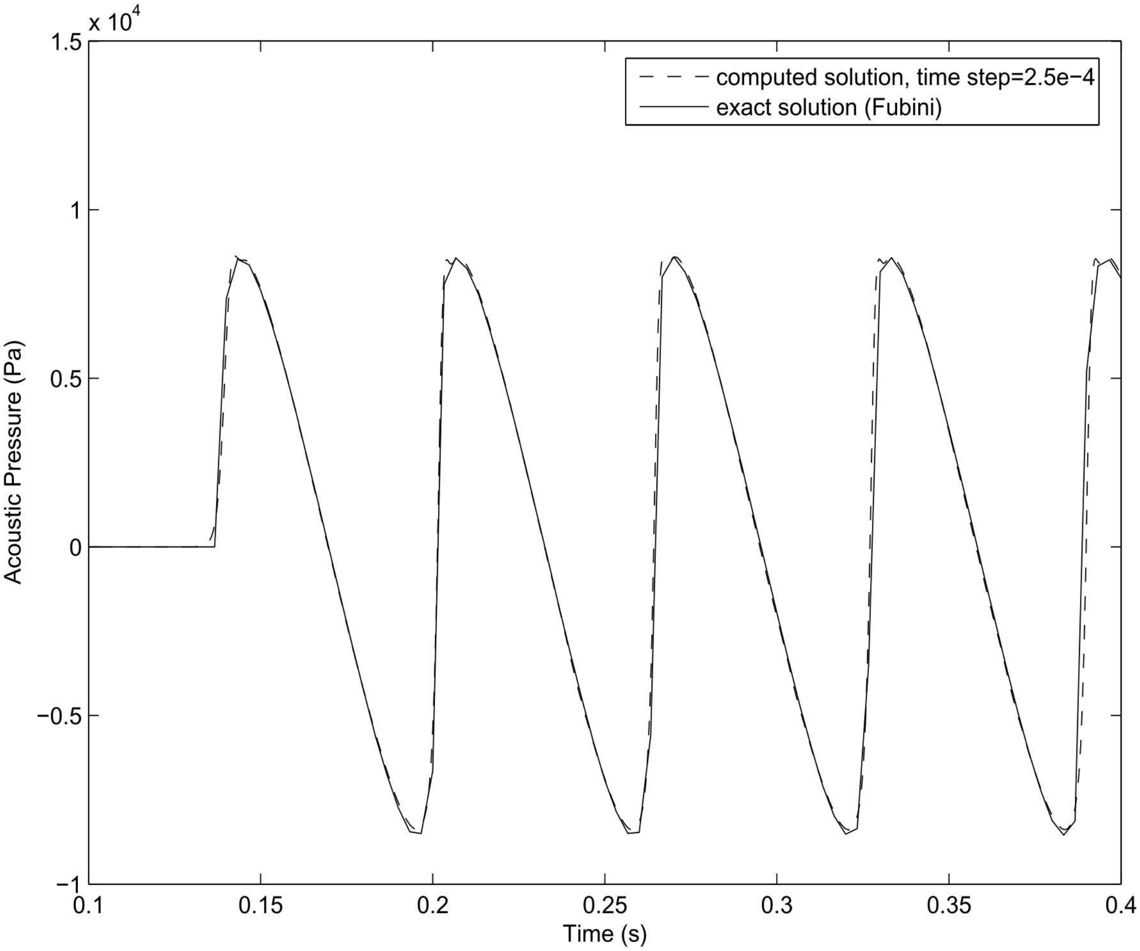


Figure 10.81: Acoustic radiated pressure at $x = \sigma$

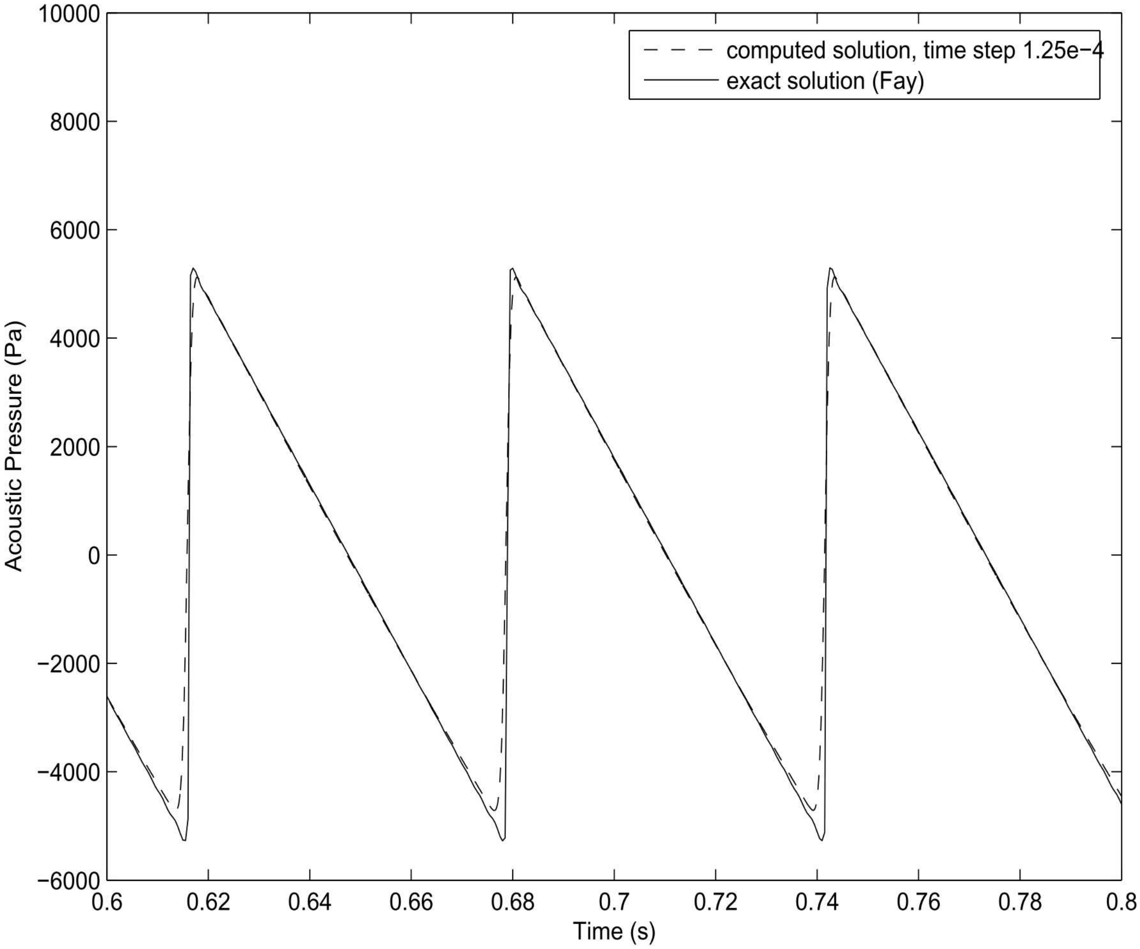


Figure 10.82: Acoustic radiated pressure at $x = 4\sigma$

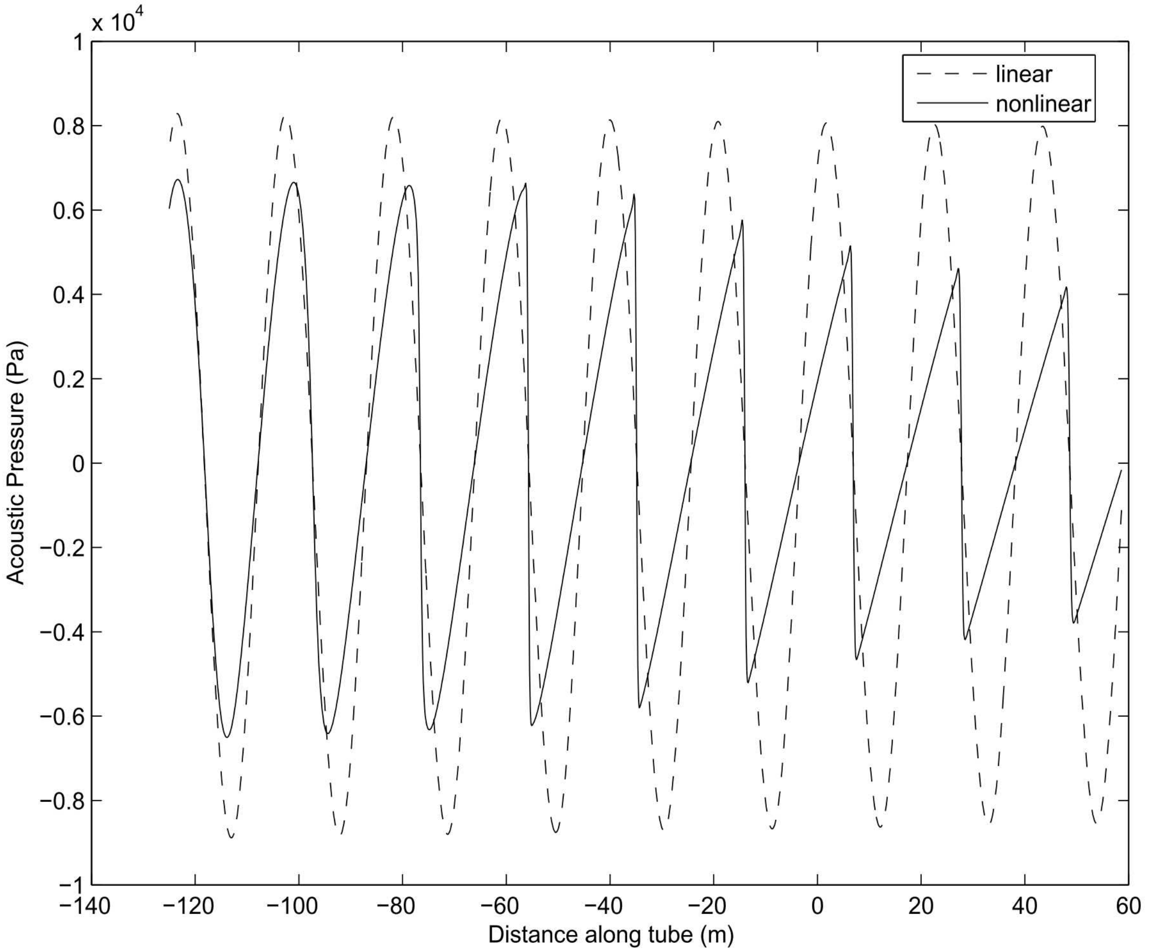


Figure 10.83: A comparison of radiated pressure using linear and nonlinear acoustic formulations

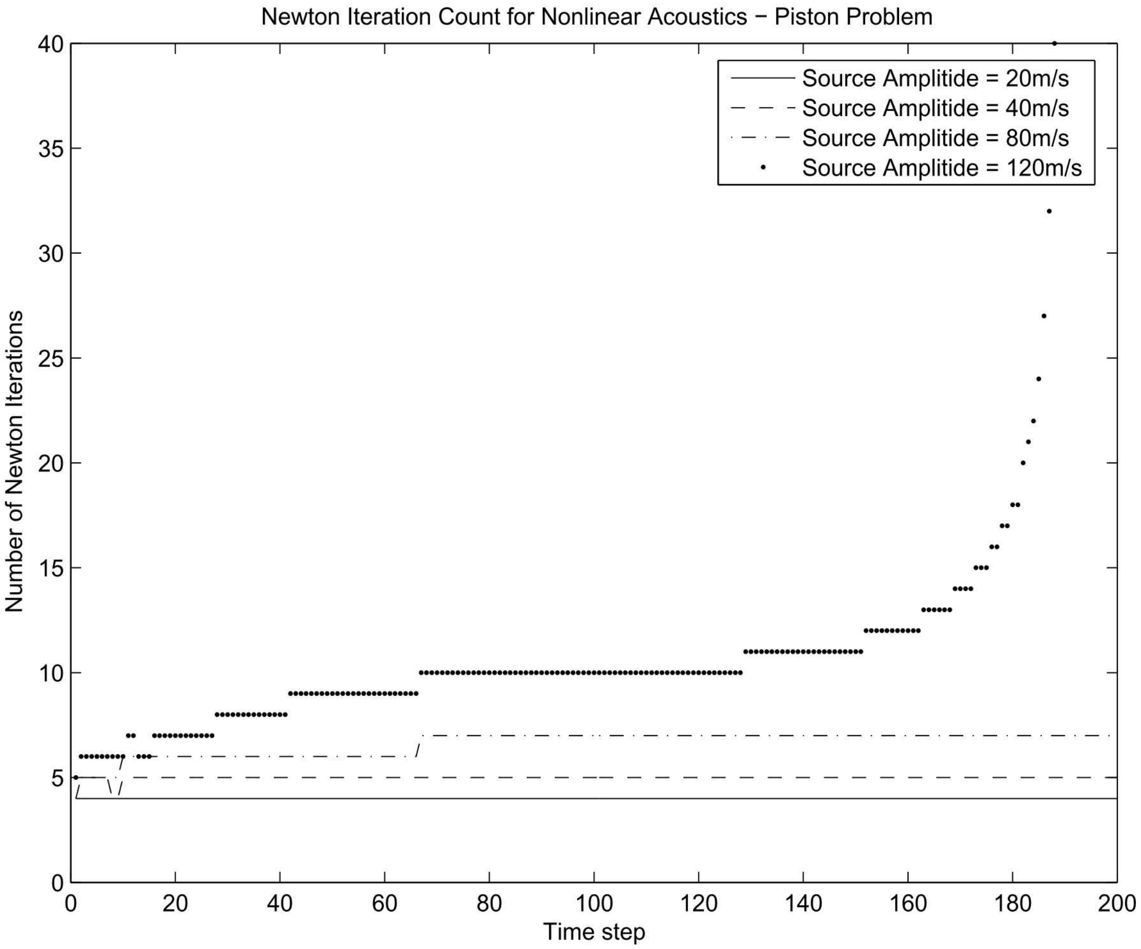


Figure 10.84: The number of Newton iterations required for convergence of the piston radiation problem.

levels of piston velocities (i.e. $20\frac{m}{s}$), the Newton iterations converge very rapidly, leveling off at about 4 iterations per time step. Interestingly, for source amplitudes that are within the range of validity of the Kusnetsov equation, the formation of shocks does not seem to 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/shockwave_SI.test for SI units and
Salinas_rtest/verification/acoustics/shockwave_english.test for english units
```

10.4 Material Identification

These verification problems are too computationally expensive to include in the automatic verification suite.

10.4.1 Elastic Material Inversion for a Tunnel

This verification problem is too computationally intense for the automatic verification suite.

In this section, we describe a materials inversion test performed on a hemispherical solid containing an embedded cylindrical tunnel of different material. Figure 10.85 shows the geometry of the model.

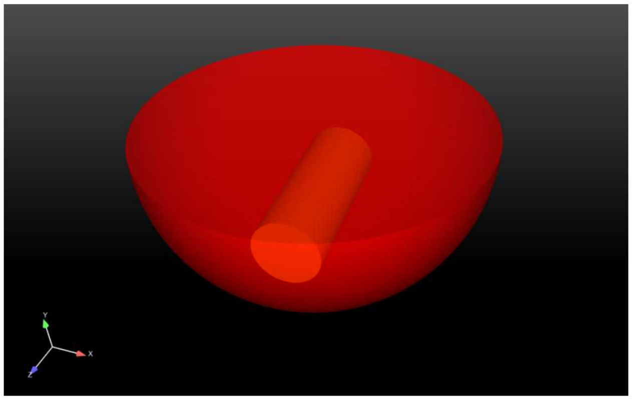


Figure 10.85: Force Inversion Test Geometry

In the model, a hemispherical solid contained a cylindrical *tunnel* region of a different solid material. A Dirichlet boundary condition was assigned on the solid, setting a fixed boundary on the hemispherical face of the model. A periodic structural loading was applied to a circular region on the flat face of the model. Figure 10.86 shows the side with the fixed boundary condition (pink) and the region of loading (orange). It was desired to determine the elastic material properties–the shear (G) and bulk (K) moduli– of the two material regions

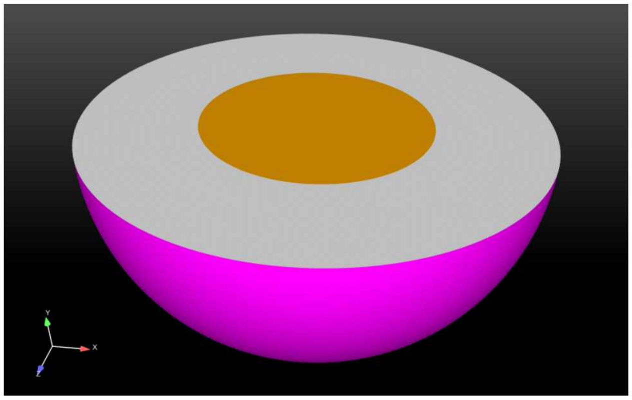


Figure 10.86: Sides with boundary (pink) and loading (orange) conditions

Synthetic input data for the inverse problem was generated by performing a forward run on the model. The data represented elastic displacements for element nodes caused by the loading on the hemisphere's face. In the forward run, the hemispherical and tunnel regions were assigned their true material properties, $\{G_h, K_h\}$ and $\{G_t, K_t\}$, displayed in Table 10.37. For the inverse run, initial guesses were chosen for the properties of the two material regions, also shown in Table 10.37. The two regions were designated as having heterogeneous, isotropic elastic materials, allowing the bulk and shear moduli to vary by element. The initial guesses, along with the input data, were used to verify that the true material properties could be recovered by the code. Figures 10.87 through 10.90 show results of the heterogeneous material-identification; cross-sections of the model are colored by the computed results for the shear or bulk moduli of the elements in the model. Figures 10.87 and 10.88 show results using a least-squares objective, while figures 10.89 and 10.90 show results using a Modified Error in Constitutive Equations (MECE) objective funcitonal.

Table 10.37: True material properties and intial guesses for tunnel-model material identification

Property	Exact	Initial Guess
G_h	150.0	90.0
K_h	150.0	90.0
G_t	50.0	90.0
K_t	50.0	90.0

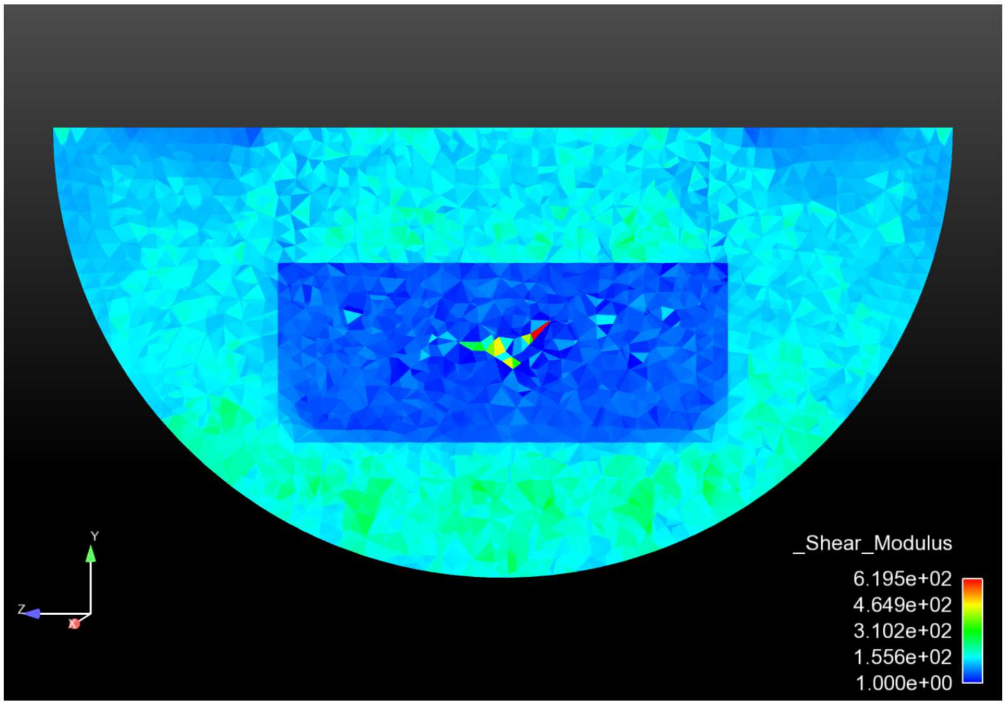


Figure 10.87: Shear modulus values of model elements, using least-squares objective

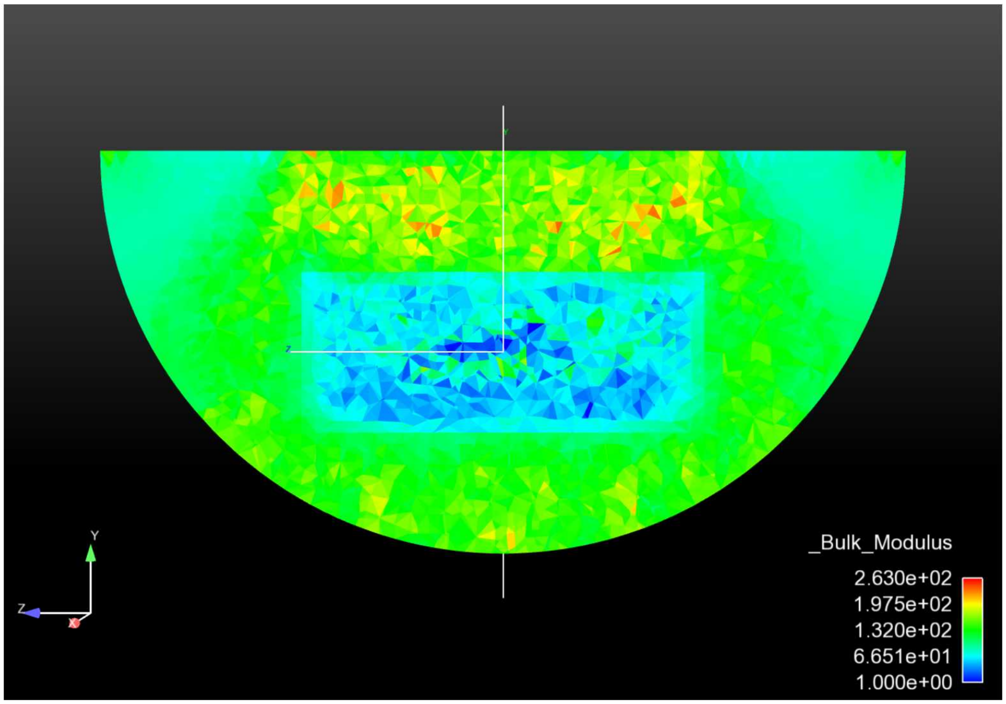


Figure 10.88: Bulk modulus results for model elements, using least-squares objective

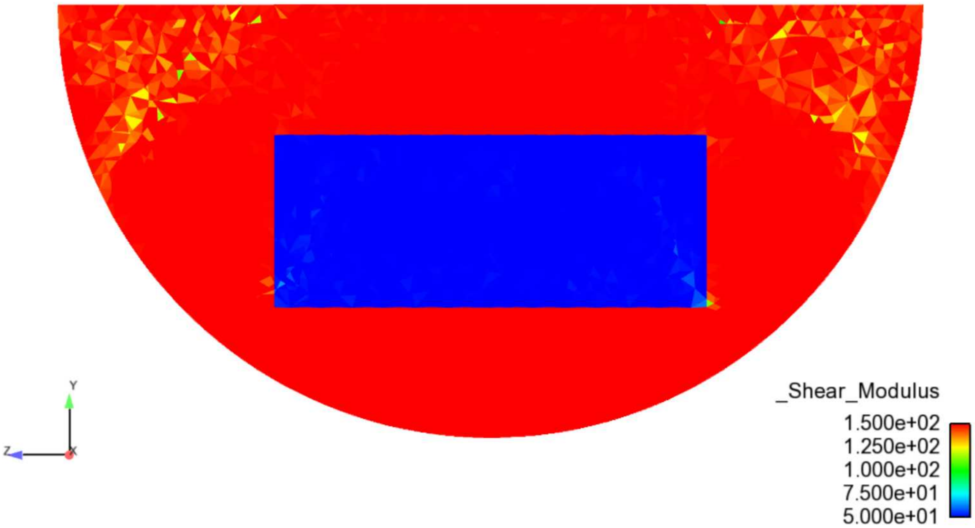


Figure 10.89: Shear modulus results, using MECE objective

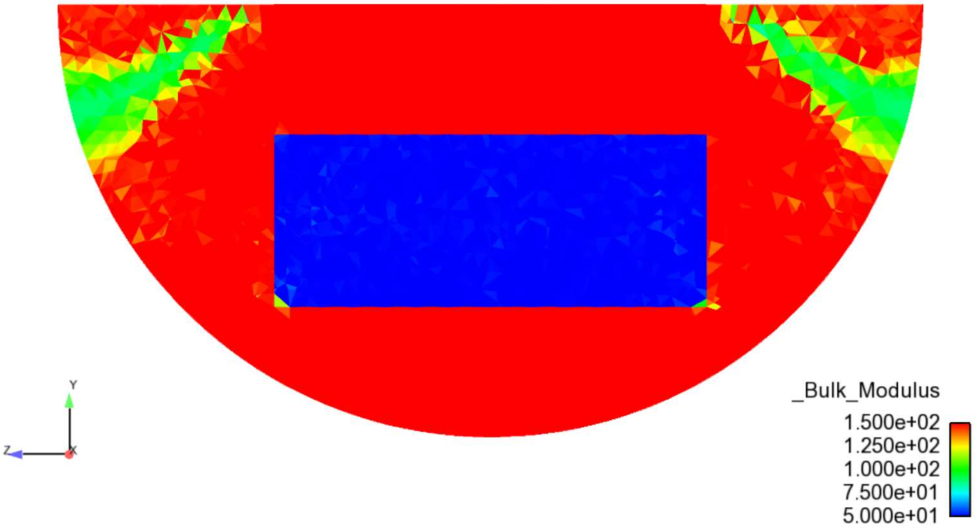


Figure 10.90: Bulk modulus results, using MECE objective

As shown in Figures 10.87 and 10.88, the elastic material properties calculated using the least-squares objective generally differentiate the two blocks and recover the blocks' original material properties. Due to the heterogeneous conditions on the block elements, element properties vary through the block volumes and include outliers. The properties recovered using the MECE objective, shown in Figures 10.89 and 10.90, much more closely recovered the original material properties of the two regions, though still demonstrated heterogeneous variations. The least-squares optimization, performed using a BFGS method, ran in parallel and underwent 30 iterations. Both the gradient and objective function were found to converge appreciably, though the objective function achieved much smaller error terms. Figure 10.91 shows the convergence behavior of the objective function and gradient for the least-squares optimization. The continuous optimization problem is solved using the Rapid Optimization Library (ROL) package in Trilinos.

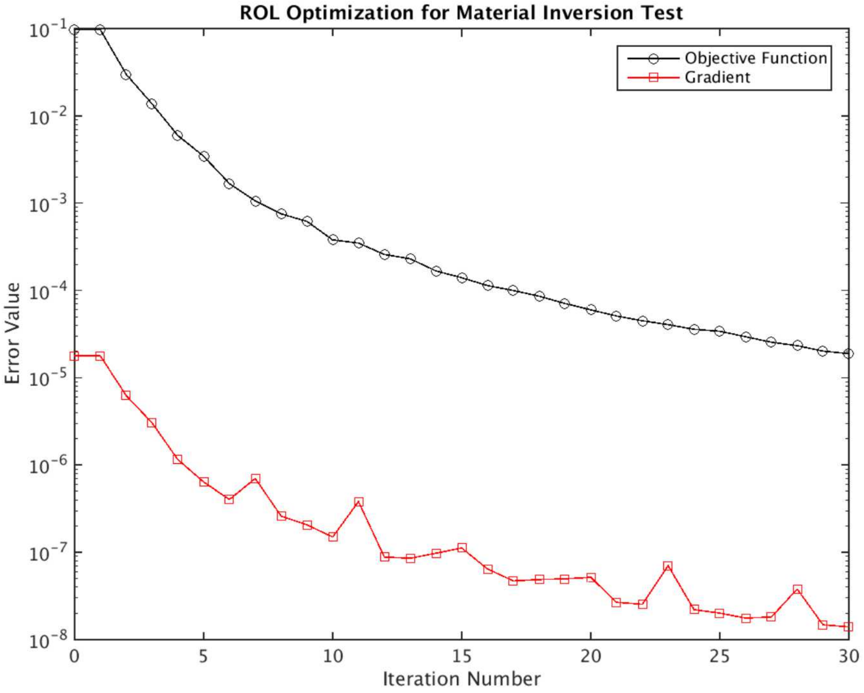


Figure 10.91: ROL optimization of objective function and gradient, using least-squares objective

10.4.2 Frequency Domain Viscoelastic Material Inversion

In this section, we describe a frequency-domain material inversion test performed on a solid assembly of two steel blocks joined by a region of viscoelastic foam material. Figure 10.92 shows the geometry of the test model.

As shown in Figure 10.92, the model assembly consists of two equally-sized steel blocks, depicted in yellow and green, joined by a region of viscoelastic foam material, shown in red.

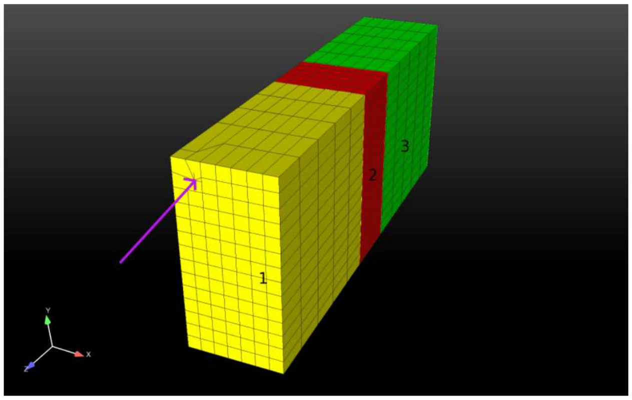


Figure 10.92: Foam block model with finite element mesh and force location

The model was discretized with a finite element mesh of Hex-8 elements. A periodic point load with a frequency of 500 Hz was applied to the yellow block, also as shown in the figure. It was desired to calculate the frequency-dependent viscoelastic material properties of the foam block, including complex values for the bulk (K) and shear (G) moduli.

Synthetic input data for the inverse problem was generated by performing a forward run in the frequency domain on the model. The data represented elastic displacements for element nodes caused by the point load acting on the model. Exact values for the foam block material properties, shown in Table 10.38, were used to generate the displacement data. For the inverse run, initial guesses were chosen for the complex valued properties of the foam block region. The foam block region was designated as isotropic and viscoelastic, the entire block sharing the same complex-valued material properties. The initial guesses, along with the input data, were used to verify that the code could recover exact material properties of the foam block. Results for the computed material properties of the foam block are also shown in Table 10.38.

Table 10.38: Exact and computed values for foam block's complex material properties

Property	Exact	Initial Guess	Computed
G Real	4000	2000	40000.001556
G Imag.	0	0	-0.005484
K Real	16000	8000	15999.999388
K Imag.	5000	0	5000.000827

As shown in Table 10.38, despite halved initial guesses for the real moduli and poor assumptions of no damping behavior, the code was able to recover material property values very well. The optimization, performed in parallel using a BFGS method, ran in parallel and underwent 95 iterations. Both the gradient and objective function were found to converge appreciably, the error decreasing especially rapidly following 80 iterations. Figure 10.93 illustrates the convergence behavior for the objective function and gradient. The continuous optimization problem is solved using the Rapid Optimization Library (ROL) package in Trilinos.

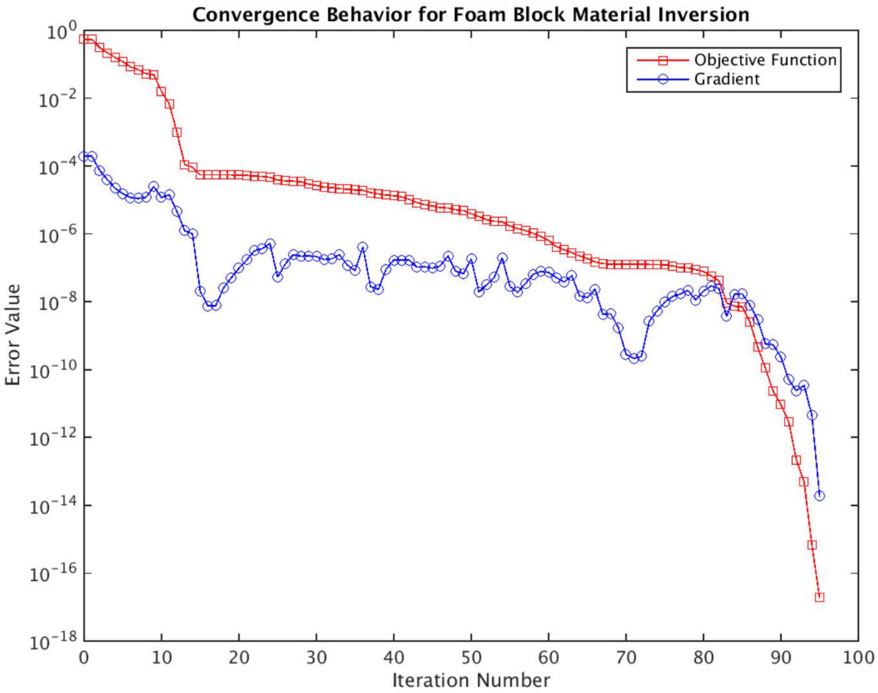


Figure 10.93: Convergence Behavior of Foam Block Material Inversion

10.5 Solution Procedures

10.5.1 Verification of Time Integration

10.5.1.1 Verification of generalized alpha damping

Though it is not always done in finite element code verification, it was deemed appropriate to verify that the generalized alpha time integrator[48] 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 so that 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

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

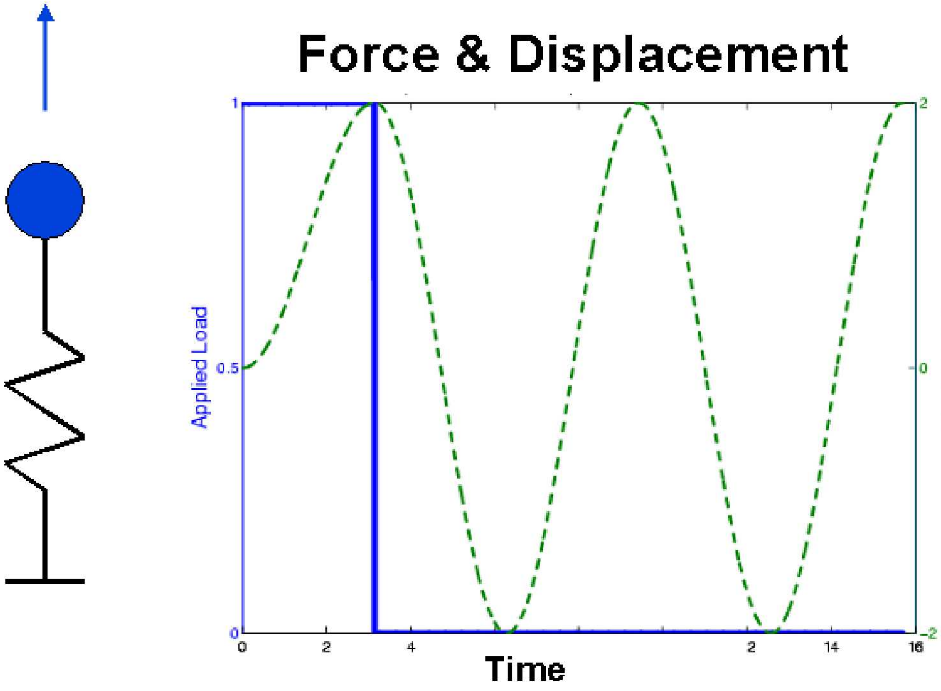


Figure 10.94: The time integrator is tested against a simple harmonic oscillator. Values of displacement at time 8π are compared and tested for convergence.

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 10.94.

Values at time 8π were compared and the resulting convergence plot is shown in Figure 10.95. We see that the convergence rate is almost exactly two – the theoretical value.

10.5.1.2 Verification of prescribed acceleration capability

In this section we present an example of verification for the prescribed acceleration capability. The example consists of 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

$$a(t) = cos(\omega t) \tag{10.111}$$

where $\omega = 2\pi f$, and $f = 16\text{Hz}$. 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

$$D(t) = \frac{1}{(32\pi)^2}(1 - cos(32\pi t)) \tag{10.112}$$

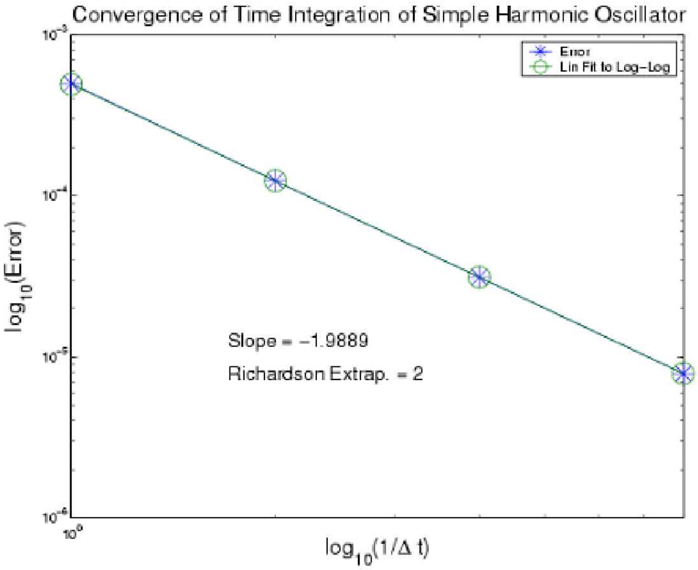


Figure 10.95: Convergence of Simple Harmonic Oscillator.

Figure 10.96 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
```

10.5.2 Direct Frequency Response

In this section we give two examples of verification of the direct frequency response driver in **Sierra/SD** . Both examples involve mass spring systems. The first is a mass spring system with stiffness proportional damping, and the second is a mass spring system with mass proportional damping.

The exact solution to this problem is given by equation 4.21a in Craig's book, 49.

$$D_s = \frac{U}{U_0} = \frac{1}{\sqrt{((1-r^2)+(2\zeta r)^2)}} \tag{10.113}$$

where 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 $\zeta = \frac{c}{2\sqrt{km}}$

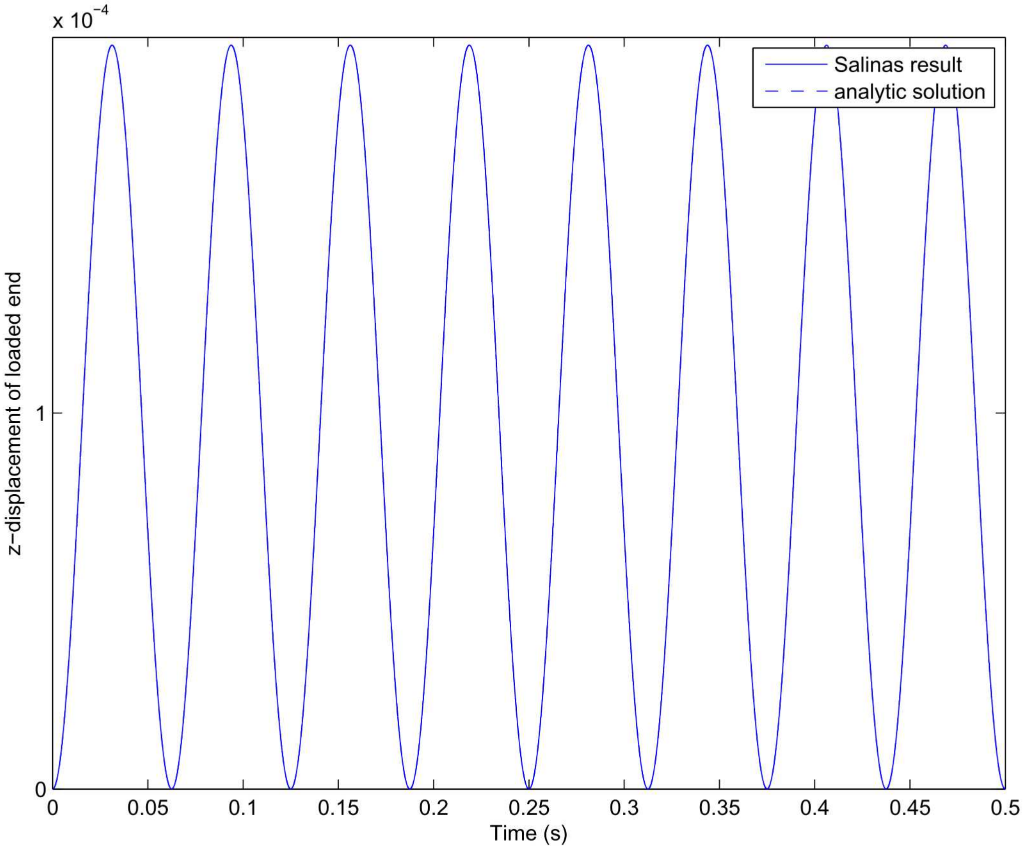


Figure 10.96: Comparison of **Sierra/SD** result with analytical solution of a beam with end-loaded prescribed acceleration.

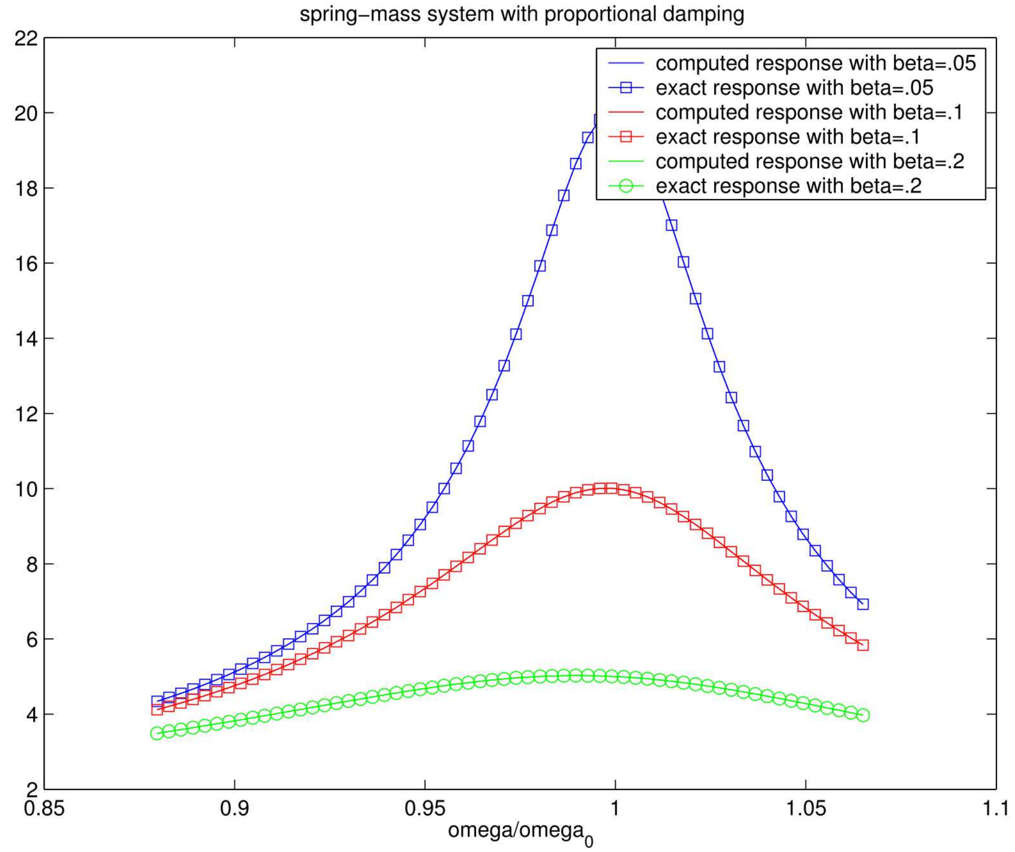


Figure 10.97: Comparison of exact and computed responses from direct frequency response of a damped spring mass system.

is the level of damping, normalized with respect to the stiffness and mass of the spring mass system. See Figure 3.1 in Craig 49 for a diagram of the problem.

For proportional damping, we have $c = \alpha m + \beta k$. The exact solutions corresponding to equation 10.113 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 10.97 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 slightly to the left. Excellent agreement is seen between **Sierra/SD** and the exact solutions.

10.5.3 Modal Frequency Response

This section presents verification examples for modal frequency response. The truth model used in these tests is the result from the corresponding direct frequency response analysis, and thus we are only verifying that the modal expansion is converging.

The first test involves a free-free beam composed of $2x2x20$ hex8 elements. The beam is subjected to a uniform pressure load on both ends and a modal frequency response solution is computed. 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 10.39. The modal frequency response results converge to the direct frequency response results as the number of modes in the modal expansion increases.

The second test 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 10.40. 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 10.39: 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 10.40: 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

10.5.4 Eigen Analysis

Eigen analysis is performed as part of the verification of the element quantities. Practically speaking, it is rather 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 10.3 and 10.4 for example.

Similarly, the elastodynamics tests examined in section 10.7.1 are built on the structure of eigenanalysis procedures. As these tests correspond to semi-analytic solutions (such as those from Blevins [8]) they constitute true verification.

10.5.5 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.

10.5.5.1 QEP – Proportionally Damped

The proportionally damped system is straightforward because the eigen vectors of the real system diagonalize the complex (or damped) solution. Consider

$$(K - \omega^2 M)\phi = 0 \tag{10.114}$$

For this system $\phi^T K \phi = \Lambda$ is diagonal, and $\phi^T M \phi = I$. The proportional damping matrix is given by $C = \alpha M + \beta K$. Clearly $\phi^T C \phi = \alpha I + \beta \Lambda$.

The solution to the j^{th} mode of the damped system is given by,

$$\Lambda_{jj} + \omega(\alpha + \beta\Lambda_{jj}) + \omega^2 = 0 \tag{10.115}$$

All quantities are known from the real eigenvalue analysis, and we can solve in terms of ω.

$$\omega_j = \frac{-(\alpha + \beta\Lambda_{jj}) \pm \sqrt{(\alpha + \beta\Lambda_{jj})^2 - 4\Lambda_{jj}}}{2} \tag{10.116}$$

Table 10.41 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 well within the expected round off. Notice that as the natural frequency increases, the fractional damping is increasing to almost 25%.

Table 10.41: 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

10.5.5.2 QEP – Viscoelastically Damped

There are no verification tests yet for this solution.

10.5.5.3 QEP – Discrete Dampers

There are no verification tests yet for this solution.

10.5.6 SA_eigen

Verification of the SA_eigen solution is complicated by the model reduction inherent in the process. Kinsler[31] 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 very 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 of the method to the untruncated solution, as the number of modes in the basis is increased.[2] Figure 10.98 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, and is not likely to be the source of this issue. At this time, we believe the problem stems from round off in transferring data from **Sierra/SD** to matlab.

[2] A subset of the tests in this section are stored in the repository in the `Salinas_test/verification/acoustic/sa_eigen` directory.

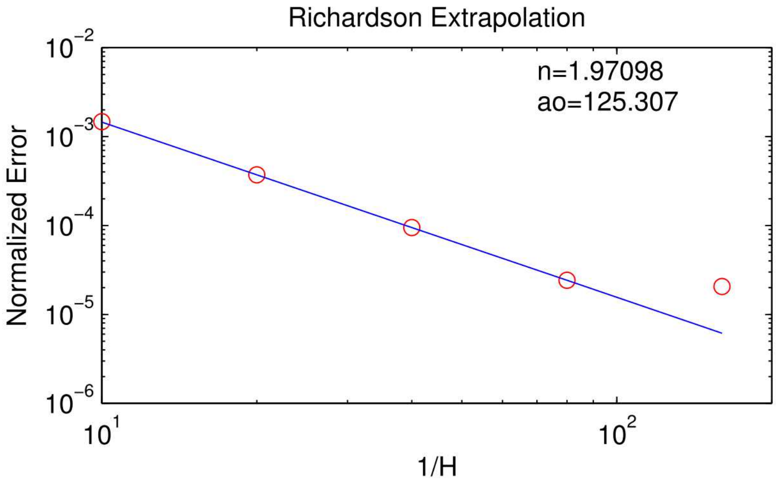


Figure 10.98: Mesh convergence to 1D Structural Acoustics Example. The example, taken from Kinsler[31] uses $a = 1/25$ and $b = 8/3$, where a and b are defined in the reference. The eigen solution is found using MATLAB's `polyeig()` function. The analytical solution from equation 9.42a of Kinsler[31] is 125.2783.

Convergence of the modally reduced model to the first coupled modal frequency when using 2 structural and 10 acoustic modes is indicated in Figure 10.99. Note that this mode converges to a value about 1% higher than the untruncated solution. Interestingly, the mode converges from below.

Figure 10.100 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 10.101, with radiation damping (or non-reflecting boundary conditions), the convergence is not as rapid and the ultimate solution less accurate than the undamped system.

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 fully fixed boundary conditions. Results shown in Figure 10.102, 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, just below the acoustic resonance (166 Hz). Figure 10.103 provides the results. The error is largest when the impedance approximates an open acoustic termination.

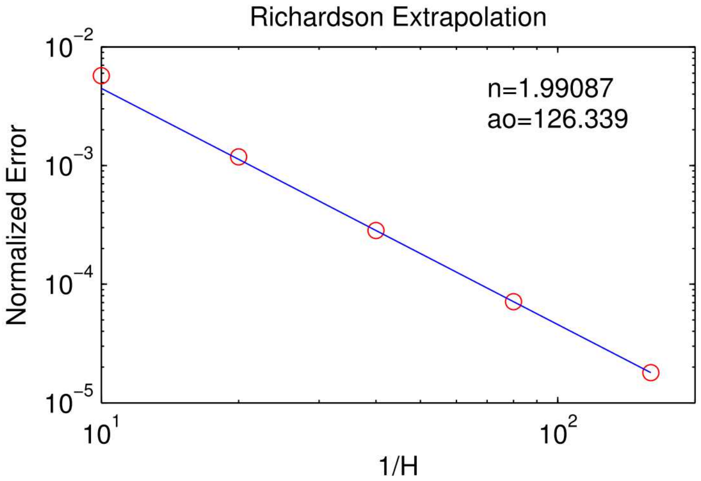


Figure 10.99: Mesh convergence to 1D Structural Acoustics Example using a modal basis. The example is that of Figure 10.98. The quadratic eigen solution is computed using 2 structural and 10 acoustic modes in **Sierra/SD** .

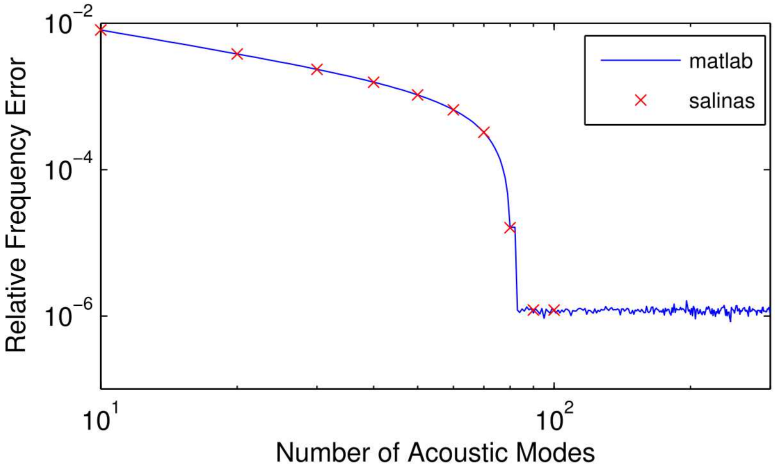


Figure 10.100: Modal convergence to 1D Structural Acoustics Example using a modal basis. The example is that of Figure 10.98, with $1/h = 80$. The quadratic eigen solution is computed using 2 structural modes, while the number of acoustic modes varied. Computation is in Matlab, with selective comparison to **Sierra/SD** . Convergence is not rapid as a full solution requires components of all axial modes. Clearly after about 80 modes, no further improvement is obtained.

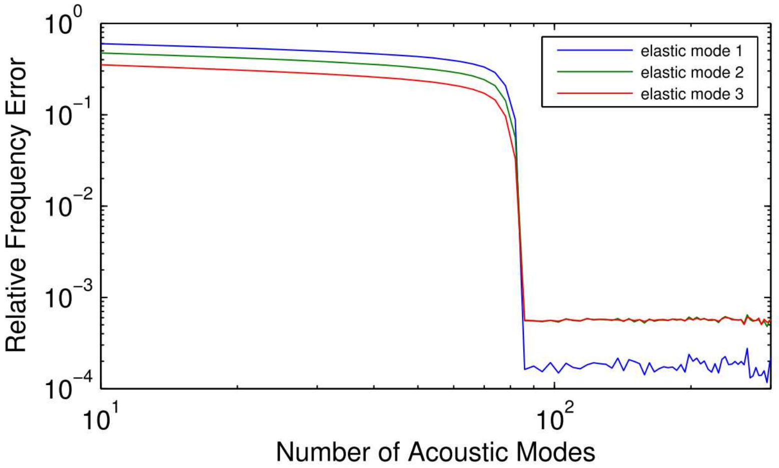


Figure 10.101: Mesh convergence to *Damped* 1D Structural Acoustics Example using a modal basis. The model is unchanged from Figure 10.100 except that there is a non-reflecting boundary condition applied on the end opposite to the structure. Matlab comparisons with `polyeig` truth model, with direct verification to **Sierra/SD** solution.

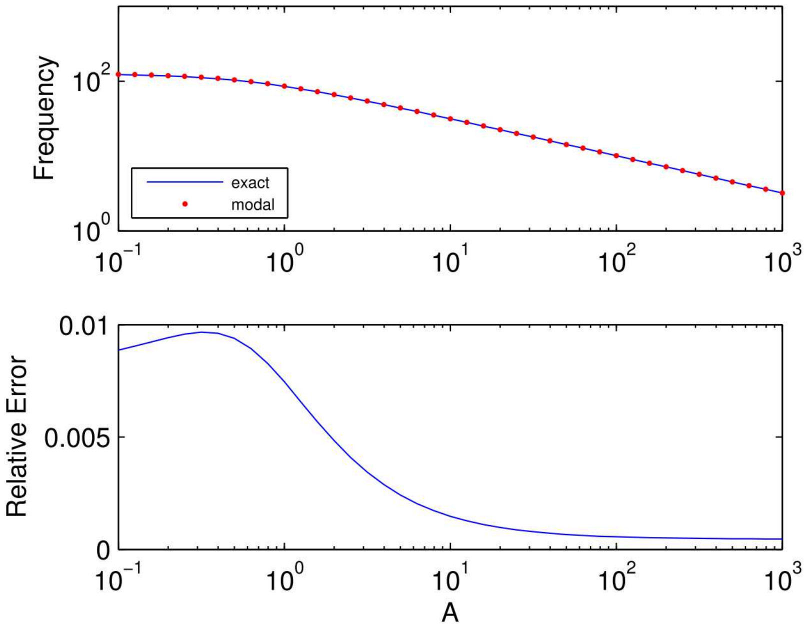


Figure 10.102: Modal convergence of 1D Structural Acoustics Example using a modal basis. The example is that of Figure 10.98, with $h = 1/80$. The quadratic eigen solution is computed using 2 structural modes and 10 acoustic modes in **Sierra/SD** , while the mass parameter, a is varied.

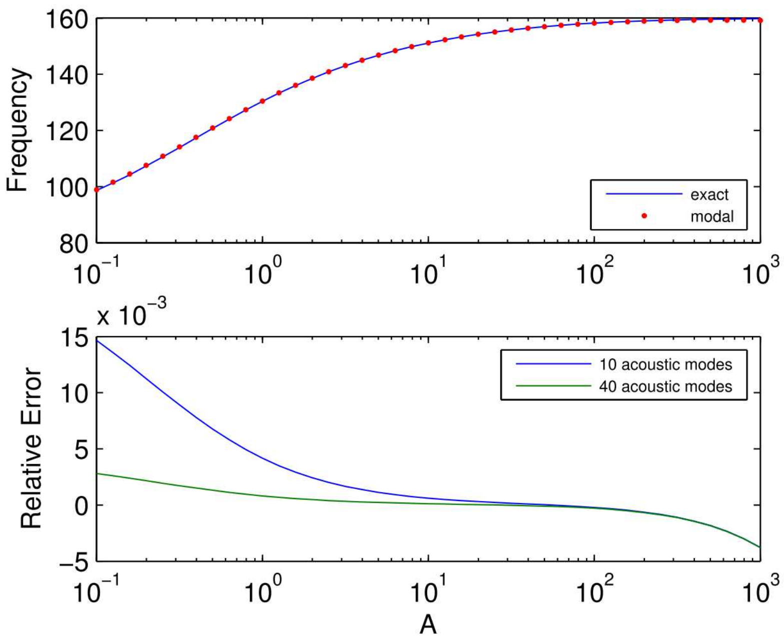


Figure 10.103: Modal convergence of 1D Structural Acoustics Example using a modal basis as the impedance is swept. The example is that of Figure 10.98, with $h = 1/80$. The quadratic eigen solution is computed while both the mass parameter, a and the stiffness parameter, b, are varied. We maintain a structural resonance of 160 Hz.

10.5.7 Buckling

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 two widely used buckling analytic solutions: buckling of a cantilever beam, and buckling of a circular ring. For now we only present the results using 3D solid elements. Once buckling for beams is fully implemented in **Sierra/SD** , we will generate the results using those elements as well.

10.5.7.1 Buckling of a Cantilever Beam

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

$$P_{cr} = \frac{2.4674EI}{L^2} \tag{10.117}$$

A simple mesh of this example was created, consisting of a 2 × 2 × 10 hex elements. The critical buckling load is predicted to be

$$P_{cr} = \frac{2.4676 \times 30 \times 10^6 \times \frac{1}{12}}{10^2} = 61675 \tag{10.118}$$

The computed buckling load was 61370.1.

10.5.7.2 Buckling of a Circular Ring

In this example, we consider buckling of a circular ring subjected to a uniform, external pressure. See section 5.9 for details.

10.5.8 Thermal Expansion

In this section we give verification examples for thermal expansion.

10.5.8.1 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

$$\Delta L = \alpha L \Delta T = 0.0001 \times 50 \times 178 = 0.89 \tag{10.119}$$

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.
```

10.5.8.2 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)

$$u(x) = \alpha(T_0 - T_i)x + \alpha(T_L - T_0)\frac{x^2}{2L} \tag{10.120}$$

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

$$u(L) = 0.0001 * 1 * 50 + 0.0001 * 1 * 25 = 0.0075 \tag{10.121}$$

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.

10.5.8.2.1 User Evaluation: A code to code comparison for a single thermal load is described in section 10.8.6.1.

10.5.9 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 very important that energy be provided at the gauss points especially for larger, higher order elements.

10.5.9.1 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 10.104. We have verified the following.

1. The specific energy is properly read into **Sierra/SD** , as verified with line sample output.
2. The specific energy is properly converted to temperature using the specific heat of the material.
3. The total energy input is determined properly.

4. Resulting displacements meet the analytic solutions (see Figure 10.105). 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.

$$\begin{aligned} \epsilon_{thermal} &= \frac{du}{dX} = \alpha_t T(X) && (10.122) \\ &= \alpha_t X / C_v && (10.123) \\ u(x) &= \frac{\alpha_t}{2C_v} X^2 && (10.124) \end{aligned}$$

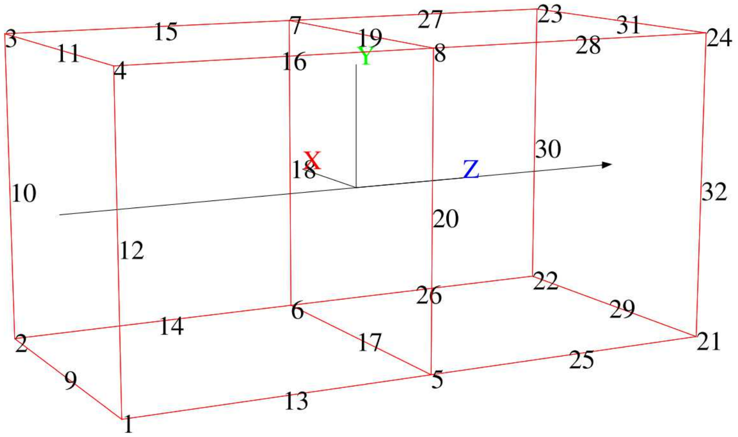


Figure 10.104: Simple Energy Deposition Test Geometry

The example is found in,

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

Resulting displacements are quadratic as from equation 10.120, with $\alpha = 0.001$, and $T_L = 1$.

10.5.9.2 Two Element Quadratic Variation Hex20

This test uses the same geometry described in section 10.5.9.1 and Figure 10.104, 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.

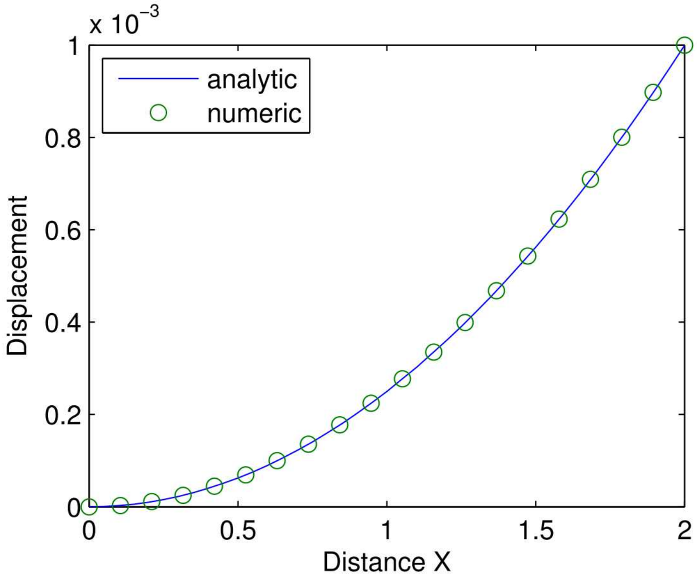


Figure 10.105: Displacements Resulting from Linear Temperature Profile

10.5.9.3 Two Element Exponential Decay Variation Hex20

This test uses the same geometry described in section 10.5.9.1, 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 10.106.

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,

$$\epsilon_{thermal} = \frac{du}{dX} = \frac{\alpha_t}{C_v} e^{-\gamma X} \tag{10.125}$$

   The solution for this equation is,

$$u = \frac{\alpha_t}{C_v \gamma} \left(1 - e^{-\gamma X}\right) \tag{10.126}$$

   Numeric and analytic solutions for this solution are shown in Figure 10.107.

The test is `edep_expx`.

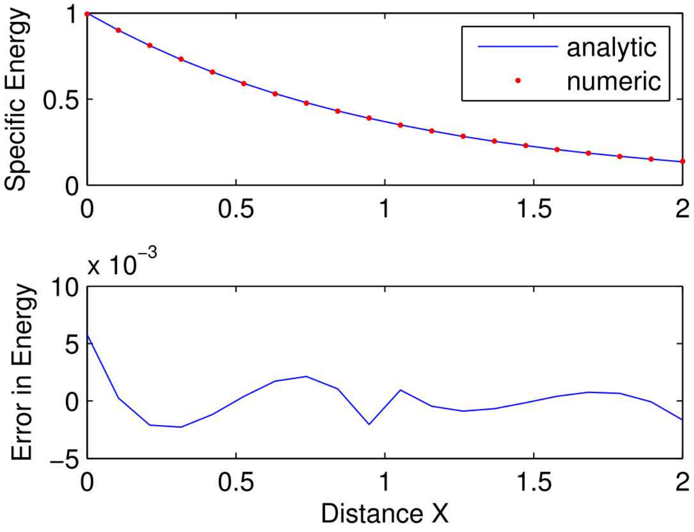


Figure 10.106: Exponential Energy Deposition. Comparison of exact and interpolated solutions from the Gauss Points.

10.5.9.4 Two Element, Two Material Hex20

Again, the same geometry is used, but with two different materials for the Hex20 elements. We require that temperature be a linear function of X, and compute specific energy, $\tilde{E} = C_v T$ to meet that requirement. This provides a simple solution for the quadratic displacement. The specific energy is shown in Figure 10.108, as extracted from line sample (`linesample` . The resulting quadratic displacement (and corresponding analytic solutions) is shown in Figure 10.109. For these solutions, the heat capacity is 1 in the first element, and 2 in the second.

10.5.10 Craig-Bampton Model Reduction

10.5.10.1 OTM Verification

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

2. check to ensure OTM is OK for displacement in serial.

   (a) Is data consistent with ϕ and ψ?

   This is checked in the debugger.

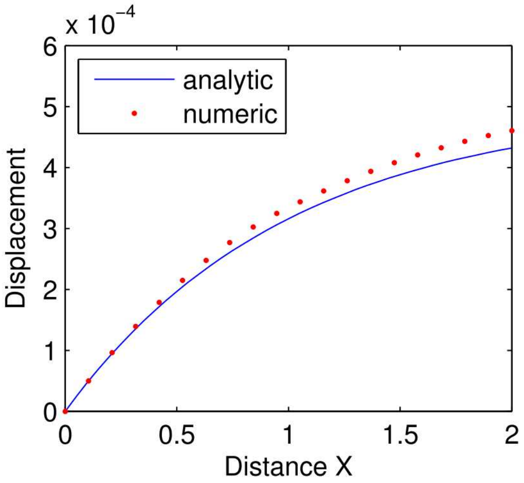


Figure 10.107: Exponential Energy Deposition, computed Displacements. The numerical results are measured at gauss points and interpolated within the elements. Displacements are interpolated from nodal values.

(b) does the product make sense (i.e.)

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

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

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 full 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 10.110.

(c) The first eigenvectors is expanded to the full 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 10.112. A comparison of the two vectors is shown in Figure 10.111. 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.

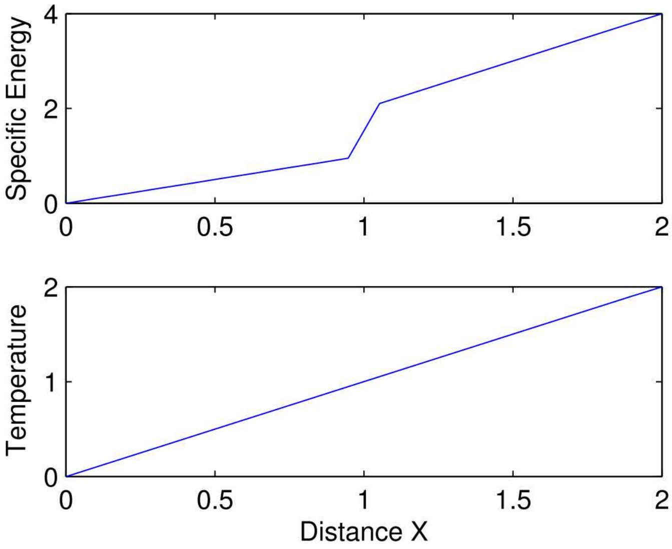


Figure 10.108: Linear Deposition on 2 Blocks. The sampled specific energy and temperature across the two blocks is shown.

10.5.11 Residual Vectors

As a small problem to test the residual vector computations in SALINAS, 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 10.42.

Table 10.42: 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[50] for a variety of boundary conditions.

Before beginning the analysis, a few words are probably necessary to explain the general anal-

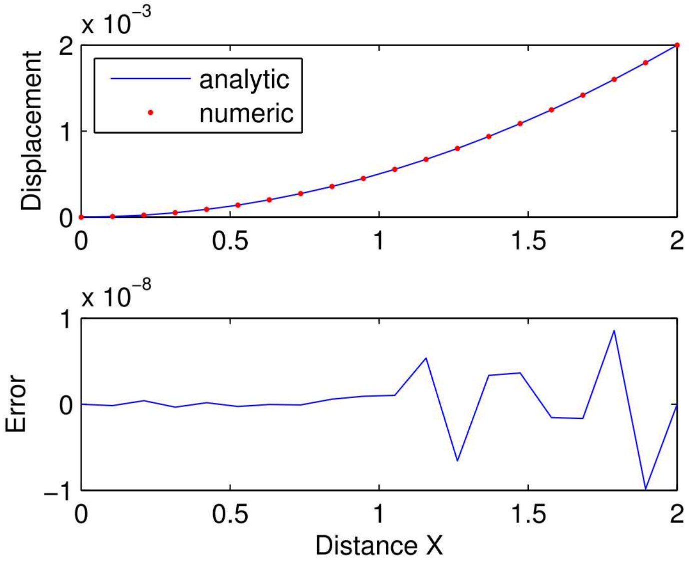


Figure 10.109: Linear Energy Deposition. The displacement response and associated error is shown.

ysis strategy. 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 usually 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 relatively 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.[51] These vectors are derived to be orthogonal to the normal modes with the same normalization so that they can simply be added to the overall basis set.

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 only occur at discrete nodal locations rather than over interfaces. This eliminates the need to compute Sinterface modes T, but is only applicable 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 centerline. Figure 104 illustrates the implementation of one of the rigid elements in NASTRAN.

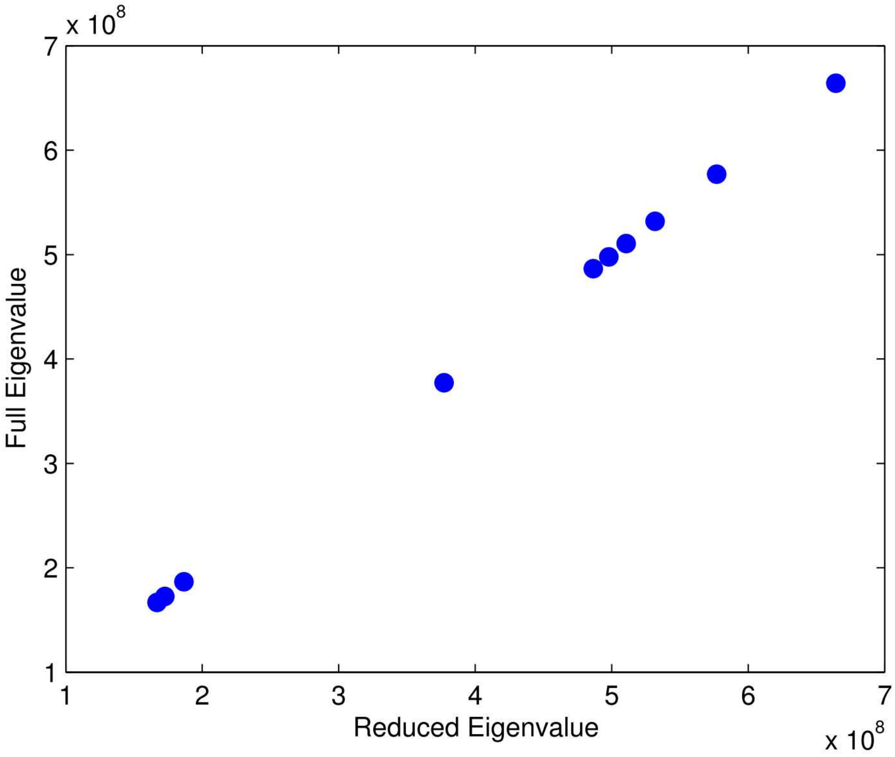


Figure 10.110: Comparison of reduced and full eigenvalues

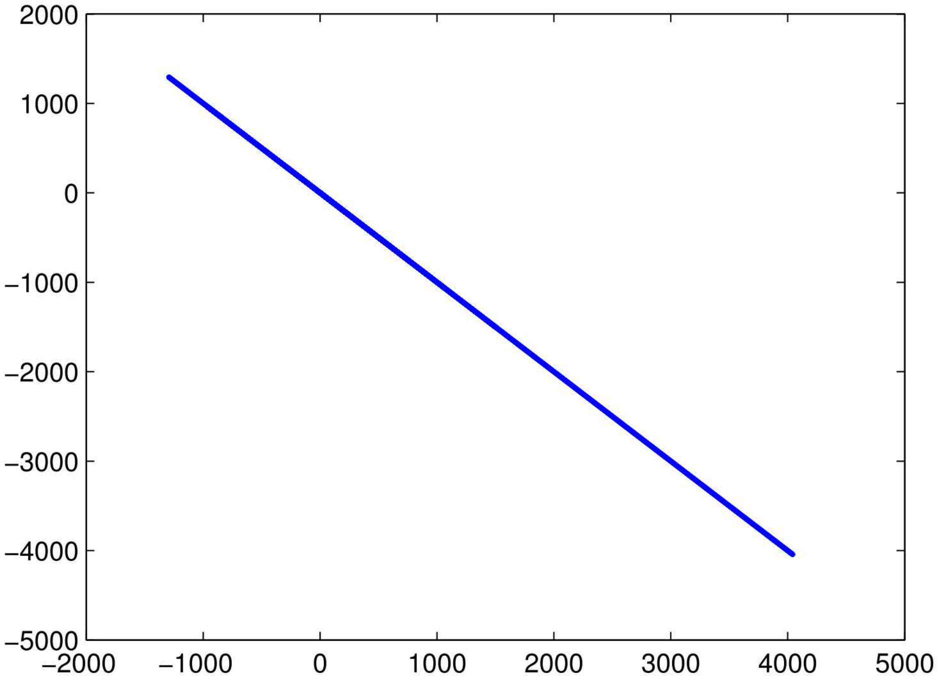


Figure 10.111: Comparison of reduced and full eigenvectors

```
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 perhaps 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 10.112: Matlab code to convert from reduced space.

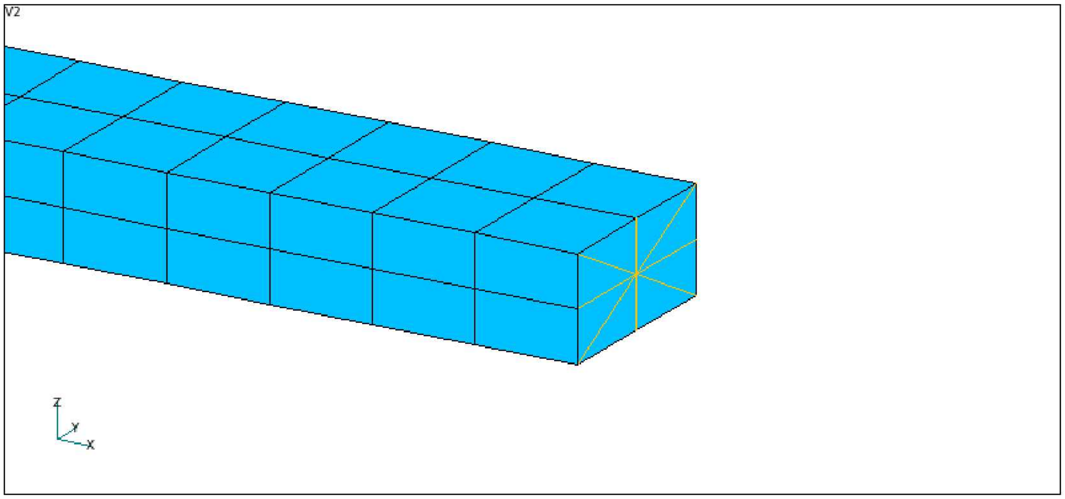


Figure 10.113: Illustration of a rigid element making all the nodes at the end of the beam dependent on a single node

This is a reasonable assumption for the beams under consideration because modes with significant variations across the cross-section occur well 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 rather than explicitly including the boundary conditions as nodal constraints in the finite element analysis. This allows the normal modes and residual vectors to only be extracted 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. Rather than extracting residual vectors for all the nodes at the ends of the beams, 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. Since the eventual goal is to only 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, so 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 10.43 lists the analytical solution for the resonance frequencies assuming free boundary conditions along with the two NASTRAN computations.

Table 10.43: 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

The integer N in the table lists the number of nodal lines along the beam's length. The table does 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 reasonably accurate results, although it is not clear why the resonance frequencies from NASTRAN are actually 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 10.44 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.[52] 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 full 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 10.45 Table 10.46 Table 10.47

Table 10.44: Resonance frequencies for the normal modes and residual modes and residual vectors in NASTRAN and SALINAS

Type	Beam 1 NASTRAN	Beam 1 SALINAS	Beam 2 NASTRAN	Beam 2 SALINAS
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

Table 10.48 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 simply changing the stiffnesses applied at the ends of the beams. The NASTRAN computations for the full model required a separate mode extraction analysis for each boundary condition.

Table 10.45: Comparison of the NASTRAN solution with an RBAR connecting the beams to the CMS solutions using NASTRAN and SALINAS for free-free boundary conditions.

N	Primary Direction	RBAR at Connection	CMS, NASTRAN	CMS, SALINAS	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 10.46: Comparison of the NASTRAN solution with an RBAR connecting the beams to the CMS solutions using NASTRAN and SALINAS for clamped-clamped boundary conditions.

N	Primary Direction	RBAR at Connection	CMS, NASTRAN	CMS, SALINAS	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 10.47: Comparison of the NASTRAN solution with an RBAR connecting the beams to the CMS solutions using NASTRAN and SALINAS for simply-supported boundary conditions

N	Primary Direction	RBAR at Connection	CMS, NASTRAN	CMS, SALINAS	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 10.48: Comparison of the NASTRAN solution with an RBAR connecting the beams to the CMS solutions using NASTRAN and SALINAS for clamped-free boundary conditions.

N	Primary Direction	RBAR at Connection	CMS, NASTRAN	CMS, SALINAS	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

10.6 Mass Properties Verification Tests

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.

10.6.1 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 10.114. 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

$$\begin{aligned} x_{cg} &= (mb + 0 - mb)/m_{total} = 0 && (10.127)\\ y_{cg} &= (0 + mb + 0)/m_{total} = b/3 && (10.128)\\ z_{cg} &= (0 + mb + 2mb)/m_{total} = 1 && (10.129) \end{aligned}$$

The components of the inertia tensor are

$$\begin{aligned} I_{xx} &= \bar{I}_{xx} + mr_x^2 && (10.130)\\ &= \bar{I}_{xx} + m\left[(2b)^2 + (b^2 + b^2) + 0\right] = \bar{I}_{xx} + 6mb^2 && (10.131)\\ I_{yy} &= \bar{I}_{yy} + mr_y^2 && (10.132)\\ &= \bar{I}_{yy} + m\left[\left(b^2 + (2b)^2\right) + b^2 + b^2\right] = \bar{I}_{yy} + 7mb^2 && (10.133)\\ I_{zz} &= \bar{I}_{zz} + mr_z^2 && (10.134)\\ &= \bar{I}_{zz} + m\left[b^2 + b^2 + b^2\right] = \bar{I}_{zz} + 3mb^2 && (10.135)\\ I_{xy} &= \bar{I}_{xy} + md_xd_y && (10.136)\\ &= \bar{I}_{xy} + m\left[0 + 0 + 0\right] = \bar{I}_{xy} && (10.137)\\ I_{xz} &= \bar{I}_{xz} + md_xd_z && (10.138)\\ &= \bar{I}_{xz} + m\left[0 + 0 - 2b^2\right] = \bar{I}_{xz} - 2b^2 && (10.139)\\ I_{yz} &= \bar{I}_{yz} + md_yd_z && (10.140)\\ &= \bar{I}_{yz} + m\left[0 + b^2 + 0\right] = \bar{I}_{yz} + b^2 && (10.141) \end{aligned}$$

A comparison between these answers and the **Sierra/SD** predictions is shown in Table 10.49. Parameters used for this problem were $m = 1$, $b = 1$, and $\bar{I}_{xx} = \bar{I}_{yy} = \bar{I}_{zz} = \bar{I}_{xy} = \bar{I}_{xz} = \bar{I}_{yz} = 0$.

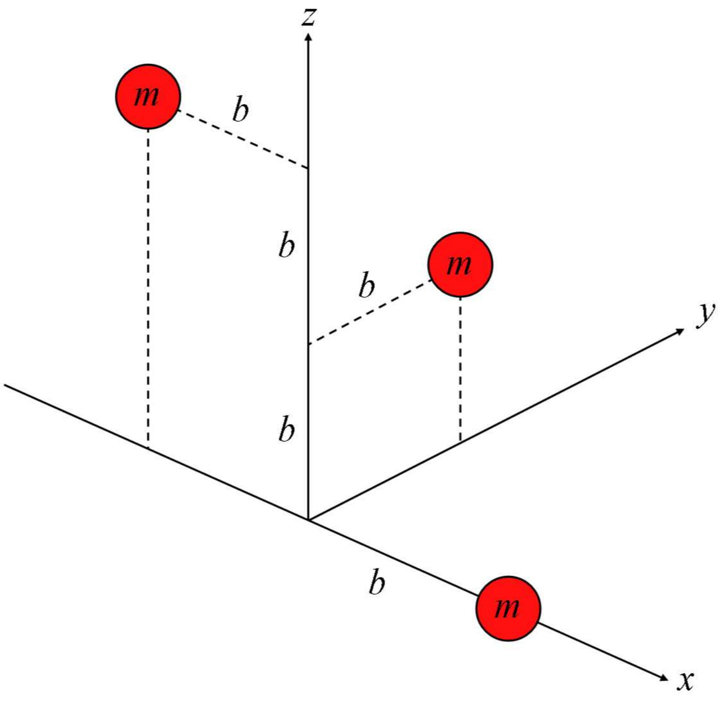


Figure 10.114: Verification problem for conmass elements

Table 10.49: 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

10.6.2 1D Verification Test

The following test was used to verify mass properties for the 1D elements which include the beam2, Obeam, Nbeam, and truss. This test case consists of a beam offset in all three dimensions from the coordinate frame as shown in Figure 10.115.

The total mass of the beam is

$$m_{total} = \rho V = \rho \pi r^2 l = 0.60kg \tag{10.142}$$

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

$$x_{cg} = 180mm - \left(\frac{180mm + 90mm}{2}\right) = 45mm \tag{10.143}$$

$$y_{cg} = 150mm \tag{10.144}$$

$$z_{cg} = 90mm \tag{10.145}$$

The components of the inertia tensor are

$$I_{xx} = \bar{I}_{xx} + mr_x^2 \tag{10.146}$$

$$= \frac{1}{2}mr^2 + m(d_y^2 + d_z^2) = 18367.5kg \cdot mm^2 \tag{10.147}$$

$$I_{yy} = \bar{I}_{yy} + mr_y^2 \tag{10.148}$$

$$= \left[\frac{1}{4}mr^2 + \frac{1}{12}ml^2\right] + m(d_x^2 + d_z^2) = 9723.75kg \cdot mm^2 \tag{10.149}$$

$$I_{zz} = \bar{I}_{zz} + mr_z^2 \tag{10.150}$$

$$= \left[\frac{1}{4}mr^2 + \frac{1}{12}ml^2\right] + m(d_x^2 + d_y^2) = 18363.75kg \cdot mm^2 \tag{10.151}$$

$$I_{xy} = \bar{I}_{xy} + md_xd_y \tag{10.152}$$

$$= 0 + md_xd_y = 4050.0kg \cdot mm^2 \tag{10.153}$$

$$I_{xz} = \bar{I}_{xz} + md_xd_z \tag{10.154}$$

$$= 0 + md_xd_z = 2430.0kg \cdot mm^2 \tag{10.155}$$

$$I_{yz} = \bar{I}_{yz} + md_yd_z \tag{10.156}$$

$$= 0 + md_yd_z = 8100.0kg \cdot mm^2 \tag{10.157}$$

A comparison between these answers and the **Sierra/SD** predictions for the 1D elements is shown in Table 10.50. The finite element model used to generate these results contained 27 elements.

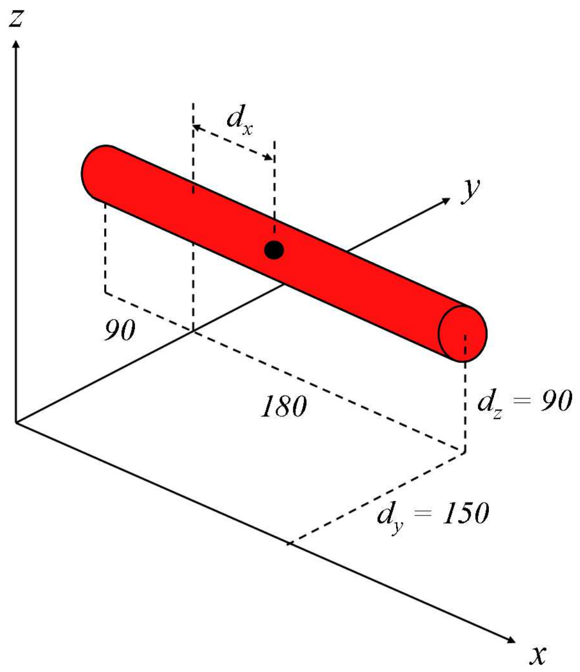


Figure 10.115: Verification problem for 1D elements

Table 10.50: Comparison of **Sierra/SD** with exact solutions for the 1D verification problem.

Property	Exact	Beam2	Nbeam	Obeam	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

10.6.3 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 10.116.

The total mass of the plate is

$$m_{total} = m_1 + m_2 = \rho\left(abt + \frac{1}{2}bct\right) \tag{10.158}$$

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

$$x_{cg} = -\frac{1}{m_{total}}\left[m_1 a + m_2\left(a + \frac{t}{2}\right)\right] \tag{10.159}$$

$$y_{cg} = \frac{1}{m_{total}}\left[m_1\left(\frac{b}{2}\right) + m_2\left(\frac{2}{3}b\right)\right] \tag{10.160}$$

$$z_{cg} = \frac{1}{m_{total}}\left[0 + m_2\left(\frac{c}{3}\right)\right] \tag{10.161}$$

The components of the inertia tensor are

$$I_{xx} = \left(\bar{I}_{xx} + m_1 r_x^2\right) + \left(\bar{I}_{xx} + m_2 r_x^2\right) \tag{10.162}$$

$$= \left(\rho_1 t_1 \bar{I}_x + m_1 d_y^2\right) + \left[\left(\bar{I}_{yy} + \bar{I}_{zz}\right) + m_2\left(d_x^2 + d_y^2\right)\right] \tag{10.163}$$

$$= \left(\frac{m_1 b^2}{12} + \frac{m_1 b^2}{4}\right) + \left(\frac{m_2 c^2}{6} + \frac{m_2 b^2}{2}\right) \tag{10.164}$$

$$= \frac{m_1 b^2}{3} + \frac{m_2 c^2}{6} + \frac{m_2 b^2}{2} \tag{10.165}$$

$$I_{yy} = \left(\bar{I}_{yy} + m_1 r_y^2\right) + \left(\bar{I}_{yy} + m_2 r_y^2\right) \tag{10.166}$$

$$= \left(\rho_1 t_1 \bar{I}_x + m_1 d_x^2\right) + \left[\rho_2 t_2 \bar{I}_x + m_2\left(d_x^2 + d_z^2\right)\right] \tag{10.167}$$

$$= \left(\frac{m_1 a^2}{12} + \frac{m_1 a^2}{4}\right) + \left(\frac{m_2 c^2}{18} + \frac{m_2 c^2}{9} + m_2 a^2\right) \tag{10.168}$$

$$= \frac{m_1 a^2}{3} + \frac{m_2 c^2}{6} + m_2 a^2 \tag{10.169}$$

$$I_{zz} = \left(\bar{I}_{zz} + m_1 r_z^2\right) + \left(\bar{I}_{zz} + m_2 r_z^2\right) \tag{10.170}$$

$$= \left(\rho_1 t_1 \bar{I}_z + m_1 d_x^2\right) + \left[\left(\bar{I}_{xx} + \bar{I}_{yy}\right) + m_2\left(d_x^2 + d_y^2\right)\right] \tag{10.171}$$

$$= \left(\frac{m_1 a^2}{3} + \frac{m_1 b^2}{3}\right) + \left(\frac{m_2 b^2}{18} + \frac{8 m_2 b^2}{18} + m_2 a^2\right) \tag{10.172}$$

$$= \frac{m_1 a^2}{3} + \frac{m_1 b^2}{3} + \frac{m_2 b^2}{2} + m_2 a^2 \tag{10.173}$$

$$I_{xy} = \left(\bar{I}_{xy} + m_1 d_x d_y\right) + \left(\bar{I}_{xy} + m_2 d_x d_y\right) \tag{10.174}$$

$$= \left[0 + m_1\left(-\frac{a}{2}\right)\left(\frac{b}{2}\right)\right] + \left[0 + m_2\left(-a\right)\left(\frac{2b}{3}\right)\right] \tag{10.175}$$

$$= -\frac{m_1 ab}{4} - \frac{2 m_2 ab}{3} \tag{10.176}$$

$$I_{xz} = \left(\bar{I}_{xz} + m_1 d_x d_z\right) + \left(\bar{I}_{xz} + m_2 d_x d_z\right) \tag{10.177}$$

$$= (0 + 0) + \left[0 + m_2\left(-a\right)\left(\frac{c}{3}\right)\right] \tag{10.178}$$

$$= -\frac{m_2 ac}{3} \tag{10.179}$$

$$I_{yz} = \left(\bar{I}_{yz} + m_1 d_y d_z\right) + \rho_2 t_2 \int_0^b \int_0^{\frac{c}{b}y} yz\,dz\,dy \tag{10.180}$$

$$= (0 + 0) + \frac{\rho_2 t_2 c^2}{2b^2} \int_0^b y^3 dy \tag{10.181}$$

$$= \frac{m_2 bc}{4} \tag{10.182}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 10.51. The 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 $\rho = 0.1lb/in^3$.

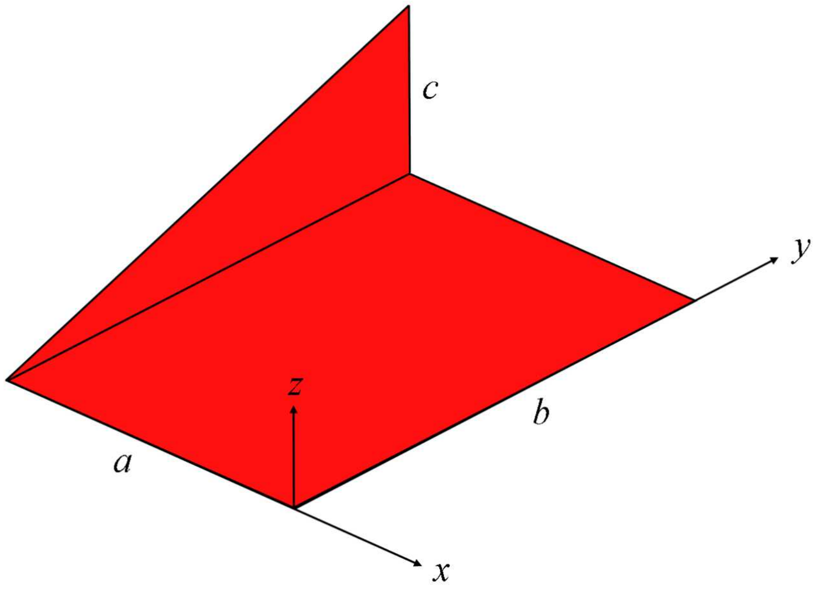


Figure 10.116: Verification problem for shell elements

Table 10.51: 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

10.6.4 3D Verification Tests

The following tests were used to verify mass properties for the 3D elements which include the hexahedral, tetrahedral, and wedge elements. Solutions for these problems were mostly taken from the dynamics text by Meriam and Kraige.[53]

10.6.4.1 Offset Block

The first 3D test consists of an offset cube as shown in Figure 10.117. The total mass of the block is given by

$$m_{total} = \rho l^3 = 3.375. \tag{10.183}$$

where ρ is the density of the block and l is the length of each side of the block. The center-of-gravity is

$$x_{cg} = y_{cg} = z_{cg} = 0.8 + \frac{1}{2}(1.5) = 1.55. \tag{10.184}$$

The components of the inertia tensor are

$$I_{xx} = \bar{I}_{xx} + mr_x^2 \tag{10.185}$$

$$= \frac{1}{12}m\left(2l^2\right) + m\left(d_y^2 + d_z^2\right) = 17.4825 \tag{10.186}$$

$$= I_{yy} = I_{zz} \tag{10.187}$$

$$I_{xy} = \bar{I}_{xy} + md_xd_y = 8.1084375 \tag{10.188}$$

$$= I_{xz} = I_{yz} \tag{10.189}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 10.52. The tet model contained 26,430 elements, and the hex model contained 343 elements. Parameters used for this problem were $\rho = 1.0$ and $l = 1.5$

10.6.4.2 Half-torus

This test consists of a half-torus as shown in Figure 10.118. The total mass is

$$m_{total} = \rho V = \rho\pi r^2\left(\pi R\right) = 0.61685. \tag{10.190}$$

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

$$x_{cg} = y_{cg} = 0 \tag{10.191}$$

$$z_{cg} = \frac{r^2 + 4R^2}{2\pi R} = -0.64657. \tag{10.192}$$

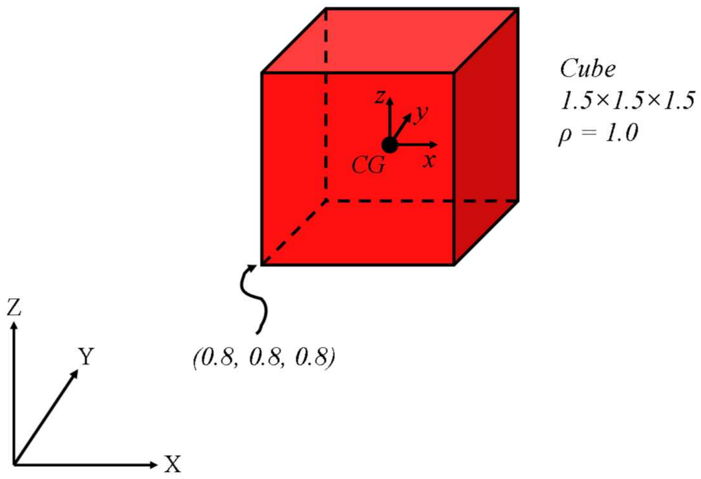


Figure 10.117: Verification problem for solid elements

Table 10.52: 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

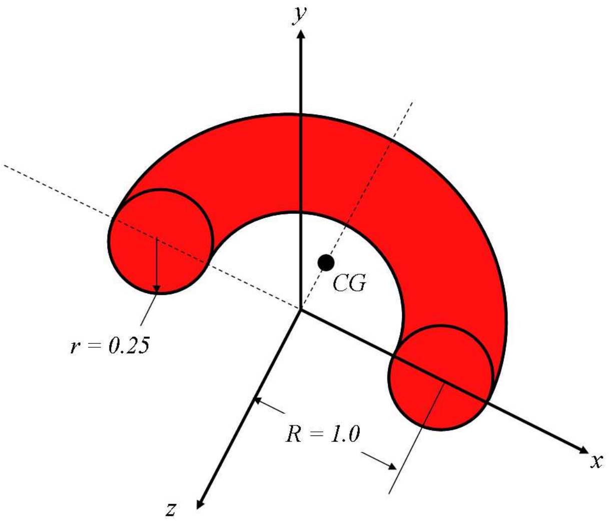


Figure 10.118: Verification problem for solid elements

The components of the inertia tensor are

$$I_{xx} = I_{zz} = \frac{1}{2}mR^2 + \frac{5}{8}mr^2 = 0.3474875 \tag{10.193}$$

$$I_{yy} = mR^2 + \frac{3}{4}mr^2 = 0.645765 \tag{10.194}$$

$$I_{xy} = I_{xz} = I_{yz} = 0. \tag{10.195}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 10.53. The tet model contained 175,592 elements. The hex model contained 62,300 elements.

10.6.4.3 Hemispherical Shell

This test consists of a hemispherical shell as shown in Figure 10.119. The total mass is

$$m_{total} = \rho V = \frac{1}{2}\left[\frac{4}{3}\pi\left(r_o^2 - r_i^2\right)\right] = 0.318348. \tag{10.196}$$

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

$$x_{cg} = \frac{r}{2} = 0.25 \tag{10.197}$$

$$y_{cg} = z_{cg} = 0. \tag{10.198}$$

Table 10.53: 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

The components of the inertia tensor are

$$I_{xx} = I_{yy} = I_{zz} = \frac{2}{3}mr^2 = 0.053058 \tag{10.199}$$

$$I_{xy} = I_{xz} = I_{yz} = 0. \tag{10.200}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table 10.54. The finite element model used to generate these results contained 108,000 hex elements.

10.6.4.4 Tetrahedron

This test consists of a tetrahedron with side lengths of a, b, and c as shown in Figure 10.120. The total mass is

$$m_{total} = \rho V = \rho \frac{1}{6} abc \tag{10.201}$$

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

$$x_{cg} = \frac{a}{4} \tag{10.202}$$

$$y_{cg} = \frac{b}{4} \tag{10.203}$$

$$z_{cg} = \frac{c}{4} \tag{10.204}$$

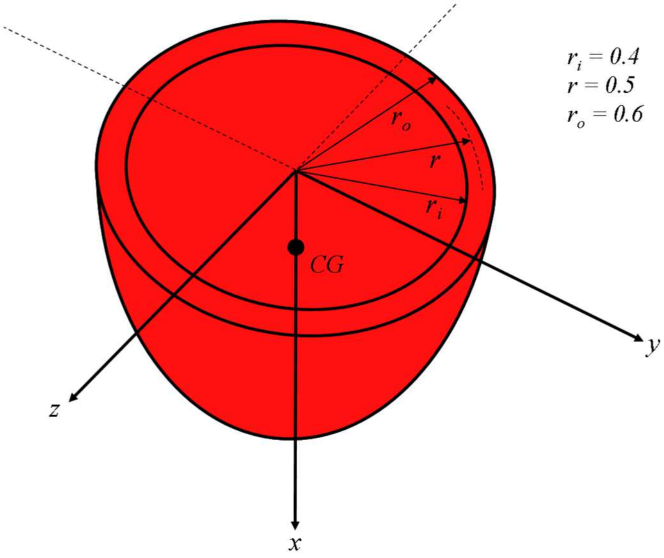


Figure 10.119: Verification problem for solid elements

Table 10.54: 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

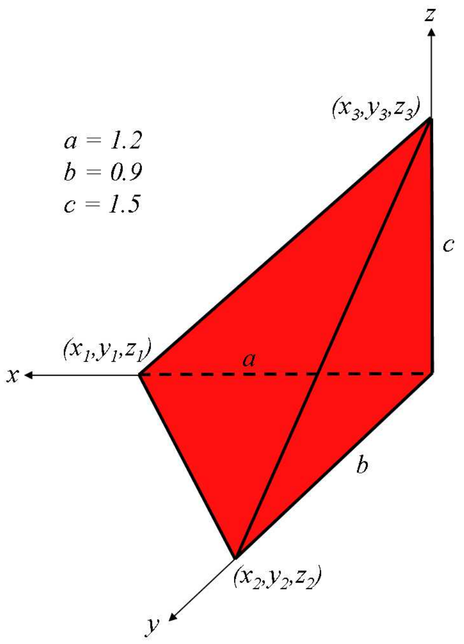


Figure 10.120: Verification problem for solid elements

The components of the inertia tensor are

$$I_{xx} = \frac{1}{10}\left(b^2 + c^2\right) \tag{10.205}$$

$$I_{yy} = \frac{1}{10}\left(a^2 + c^2\right) \tag{10.206}$$

$$I_{zz} = \frac{1}{10}\left(a^2 + b^2\right) \tag{10.207}$$

$$I_{xy} = \int_m xydm = \rho \int_V xydV \tag{10.208}$$

$$= \int_0^a \int_0^{1-\frac{x}{a}} \int_0^{1-\frac{x}{a}-\frac{z}{c}} xydydzdx = \frac{1}{20}mab \tag{10.209}$$

$$I_{xz} = \int_m xzdm = \rho \int_V xzdV \tag{10.210}$$

$$= \int_0^a \int_0^{1-\frac{x}{a}} \int_0^{1-\frac{x}{a}-\frac{y}{b}} xzdzdydx = \frac{1}{20}mac \tag{10.211}$$

$$I_{yz} = \int_m yzdm = \rho \int_V yzdV \tag{10.212}$$

$$= \int_0^b \int_0^{1-\frac{y}{b}} \int_0^{1-\frac{x}{a}-\frac{y}{b}} yzdzdxdy = \frac{1}{20}mbc \tag{10.213}$$

A comparison between these answers and the **Sierra/SD** predictions is listed in Table

Table 10.55: 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

10.55. 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.

10.7 Phenomenon Based Testing

Each of the phenomena identified in the Phenomenology Identification and Ranking Table (PIRT) from the V&V plan has specific tests for evaluation of the predictability of the software. Details are described in the sections below.

10.7.1 Elastodynamics

The requirements for elastodynamics are detailed in the requirements document and the computational plan. They may be summarized in Table 10.56. Verification aspects for each requirement will be detailed in sections of this chapter.

10.7.2 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*.[23] Note that these solutions are usually 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 10.57 and 10.58 will be used for Beam Elements: For Shell elements Tables 10.59 through 10.61 are used. The reference table is from *Roark*.[23]

For solids, we employ Table 10.7. In addition, examples from the beams and shells may be computed using solid elements and a suitable discretization.

Table 10.56: Elastodynamics Requirements

#	**Requirement**
1	Compute static responses
2	Compute eigenanalysis for large models (10M DOFS or greater). Include: - Frequency Response Functions - Random Vibration inputs and Response - Shock Spectra
3	Compute Time domain analysis of these models, using direct time integration. An interface to facilitate time domain analysis using modal superposition will also be provided.
4	Output Stresses, Strains, Displacements, Velocities and Accelerations
5	Provide a platform for development of additional structural dynamics capabilities. These will include system identification, design optimization, nondeterministic methods, coupled/multi-physics solutions and others.
6	Provide portability and scalability to allow effective use on ASCI-red and ASCI-white. Data file compatibility with other ASCI codes.
7	Loads: - point loads (applied though node sets) - gravity loads on elements - pressure loads
8	Support standard elements from FE analysis - solid elements (HEX,WEDGE,TET) - shells (Triangle, Quad) - Beams - point masses, springs - MultiPoint Constraints
9	Support linear, elastodynamic material models with full anisotropy.
10	Documentation: - a users manual -programmers manual - software engineering practices

Table 10.57: 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.	*Roark*	.6356	2.311
	Applied point force	Beam2	.6356	2.312
		Tria3	.5783	2.104
		Tria3⊥	.5785	2.104
3 3b	cantilever guided.	*Roark*	4.032	-8.064
	Applied point moment	Beam2	4.032	-8.064

The "Tria6⊥" model is rotated so a pure membrane deformation occurs. A finer mesh is required.

Table 10.58: Curved Beam Element Analytic Solutions

Roark Table	**Description**	**Case**	**Dv**
17 1	opposed radial loading on circular ring	*Roark*	-5.9513
		Beam2	-5.950
17 2	opposed in-line loading on circular ring	*Roark*	.8263
	(measured at $\theta = 30^o$)	Beam2	.8259
17 3	opposed moments on circular ring	*Roark*	7.9743
	(measured at $\theta = 30^o$)	Beam2	7.967

Table 10.59: 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*	0.01701
		Tria3	0.01696
24 1b	Outer edge simply supported. Inner edge guided	*Roark*	.0068853
		Tria3	.006885
24 1e	Outer edge fixed. Inner edge free	*Roark*	.0034952
		Tria3	.0034946
24 5a	Outer edge simply supported. Inner free		

Table 10.60: 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 10.61: 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 10.62: 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				

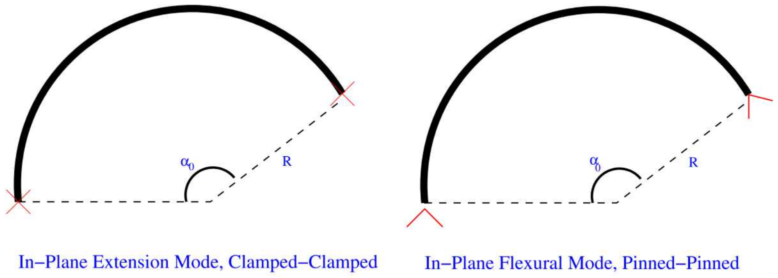

α0
R
α0
R
In−Plane Extension Mode, Clamped−Clamped
In−Plane Flexural Mode, Pinned−Pinned

Figure 10.121: Blevins Table 9-2.1 and 9-2.2 Geometries

10.7.3 Verification With Respect to Semi-Analytical Eigen Analysis

Analytic and semi-analytic solutions for eigensolutions have been determined for many geometries and have been reported in *Blevins*.[8] Note however, that these solutions are usually 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.

An eigensolution 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 eigenfrequencies 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 10.63. For beam elements, eigensolutions are described in Tables 10.64 through 10.66. Shell elements use Tables 10.67 through 10.69. 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 10.121.

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.

For all the following examples in this section (i.e. Tables 10.63 through 10.69, unless otherwise noted we use material properties for steel, i.e. $E = 30\text{x}10^6$ psi, $\nu = 0.30$ and $\rho = 0.288$ lbs/in^3 (7.4592x10^{-4} slugs/in^3).

10.7.4 Linear MultiPoint Constraints

MultiPoint Constraints (MPCs) are applied in structural dynamics for a number of reasons. Typical uses include spreading a load over many input nodes, attaching dissimilar

Table 10.63: Spring Mass System Eigenproblems

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 10.64: Beam Mass System Eigenproblems

Massless beam has square cross section with I1=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

meshes, connecting lumped structures, applying boundary conditions and approximating rigid structures. The variety of uses for MPCs makes verification of their application quite difficult. Only very 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.

Table 10.65: Straight Beam Eigenproblems - 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 10.66: 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 apply strictly only to 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 quite well 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 10.67: 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 10.68: Circular Plates - Bending

Circular disk made of QuadT elements.

Blevins	**Description**	**Sol'n**	**Mode Number**			
Table		**Type**	**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 mass at center (M large)	Estim.	25.98	N/A	N/A	N/A
		FE	25.83	N/A	N/A	N/A

Table 10.69: Rectangular Plates - Bending

Using *Tria3* elements, aspect ratio $a/b = 1.5$ in all cases.

Blevins	**Description**	**Sol'n**	**Mode Number**			
Table		**Type**	**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-clamped-clamped.	Exact	2608.74	4029.22	6387.69	6428.04
		FE	2608.29	4027.90	6387.04	6425.11
11-4.16	Simply supported (all 4 edges)	Exact	1377.13	2648.23	4237.00	4765.01
		FE	1376.97	2648.01	4237.05	4766.57
11-4.6	Clamped-free-simply supported-free	Exact	652.94	1103.68	2127.08	2747.82
		FE	648.82	1100.31	2113.90	2733.90

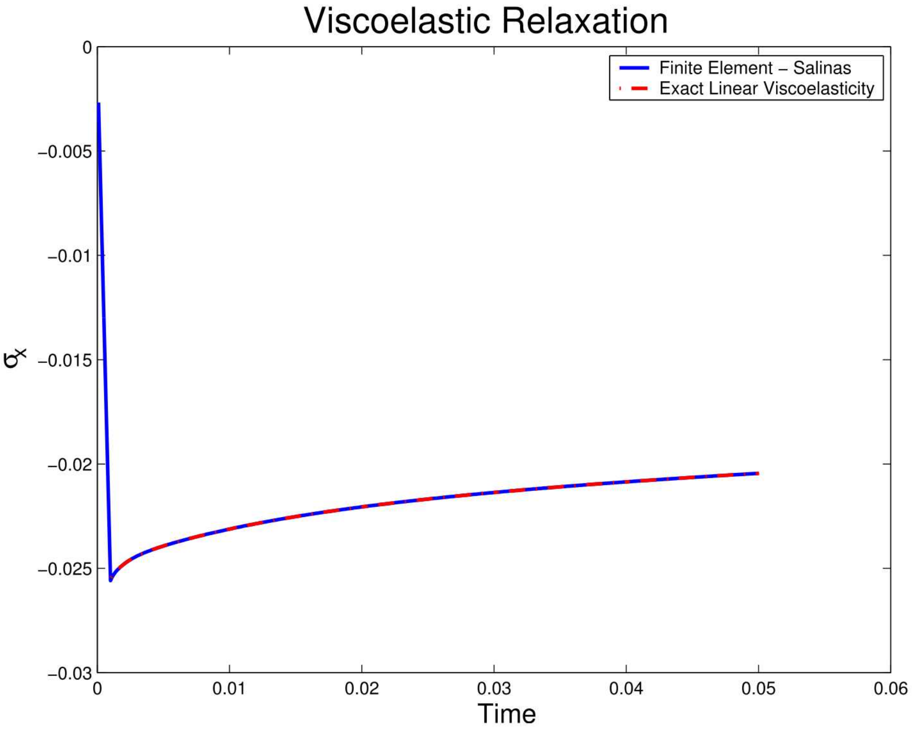


Figure 10.122: Viscoelastic Relaxation. The **Sierra/SD** results reproduce the exact solution viscoelastic relaxation after ramp and hold deformation.

10.7.5 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 isochronic, 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,

$$\sigma(t) = \dot{\gamma} E_\infty \Delta t - \dot{\gamma} \sum_n (E_G - E_\infty) \tau_n (1 - e^{\Delta t/\tau_n}) e^{-(t+\Delta t)/\tau_n} \tag{10.214}$$

A plot of the above exact solution and the predictions of **Sierra/SD** are presented in Figure 10.122.

10.7.6 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*[54]). 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 made to asymptotic quantities, i.e. quantities for which the accuracy of the truth model code must ultimately converge.

- The value gained by the comparison provides important insight not readily obtained by solution of analytic problems.

It should always be remembered that verification can never be complete, i.e. it is impossible to fully test or verify any but the simplest of applications. Therefore, any method that can provide additional examination of the application provides a value.

A number of benchmark problems exist in the literature (see for example *MacNeal*[20]). Some of these 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 10.70.

Table 10.70: 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%

Figure 10.123: Tire Analysis Model

10.7.6.1 Membranes and Transfer from SierraSM

In this case, analyses of a preloaded (inflated) tire from Sierra are compared to Abaqus^{TM}. The tire model (Figure 10.123) 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 10.71. 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 10.71: Comparison of Eigen Frequencies of the Mooney-Rivlin Inflated Tire. As many of the modes come in pairs, only the first of the paired frequencies is listed.

10.8 User Evaluations

While not rigorous in the same sense as closed form solutions, most analysts would agree that evaluation by independent outside analysts is a very valuable criteria in determining the suitability of an analysis package. Such evaluation measures not only the answers to well defined problems, but it provides confidence in the entire process and product. For example, if the tools are lacking to provide a reasonable model, this becomes readily apparent.

Where outside evaluations have been performed, we provide a summary and contact information so the analyst may follow up on the data.

10.8.1 Newport News Shipyard

Contact: Travis Kerr `kerr_te@nns.com` and Jay Warren `warren_je@nns.com`

On two separate occasions, Newport News shipyard has worked with Sandia to model their aircraft carriers. In October of 2000, and then again in October of 2002, they sent analysts to Sandia to perform a whole ship model eigen and transient dynamics analysis. Part of the first visit involved evaluation of a suite of tests. Unfortunately, Sandia was not provided with any report on this evaluation. NNS has continued interest in using **Sierra/SD** .

10.8.2 British Atomic Weapons Establishment (AWE)

Contact: Trevor Hensley. `Trevor.Hensley@awe.co.uk`

From June to December of 2002, Trevor Hensley of the AWE evaluated Sandia's ASCI applications in Albuquerque. **Sierra/SD** was among the first evaluated. One problem was identified in statics. While **Sierra/SD** converged to the proper displacement, it did not appear to have the proper stress concentration. This turned out to be an issue of understanding **Sierra/SD** 's stress output which on shells is in the element coordinate system.

The AWE is currently negotiating to obtain a copy of **Sierra/SD** for their analysis at their site.

10.8.3 NASA

Contact: Lloyd Purves, `lpurves@hist.nasa.gov`

This evaluation did not go well for several reasons.

1. There was a shortage of manpower. Lloyd had a summer student who was doing most of the work, but the student did not have sufficient expertise to finish.
2. There were hardware and software installation problems. NASA personnel were not able to visit us here, nor were **Sierra/SD** personnel given access to NASA machines. Thus installation of the software became a real road block.
3. Probably most importantly, the goals did not match well. **Sierra/SD** is not a plug-in replacement for NASTRAN. It has a wide variety of elements, but it also lacks capability that may be unique to NASTRAN. For example, **Sierra/SD** has no axially symmetric elements. Other translation issues (such as differences in spring formulations) caused a good deal of difficulty.

10.8.4 Lockheed Martin – Denver

Contact: Dan Morganthaler, `daniel.r.morganthaler@lmco.com`

This interaction was funded under the Lockheed Martin shared vision program. It met only limited success. The main impediments were with the difficulty in getting **Sierra/SD** to run properly on the parallel platforms. Dan visited Sandia for a few days, but our parallel machines were too heavily used to get the runs through. Eventually, Dan was able to get the analysis done using superelement capabilities in Nastran. The report is available in draft form.[55]

10.8.5 Advatech Pacific

Contact: Peter Rohl, `peter.rohl@advatechpacific.com`

Advatech prepared a fairly extensive study comparing the results of **Sierra/SD** and NE/Nastran on a variety of structures. This is available as conference proceedings.[56]

10.8.6 Sandia Labs

10.8.6.1 Thermal Strains: comparison with Abaqus

In December 2005, Wil Holzmann performed a code to code comparison of abaqus and **Sierra/SD** for thermal expansion. This provides a real world comparison – most would not consider it a verification.

Figures 10.124 and 10.125 relate to the analysis of a thin walled frustum for thermal loads. Two model versions were created, one in abaqus and the other in **Sierra/SD** . The energy deposition data was provided by mapping the data from the element centroids to the nodes using paraview. A scaling term was applied to convert energy deposition to equivalent thermal loads. The figures compare plots of e33 volumetric strains. The two approaches compare very well.

The model is about 135,000 degrees of freedom, which is too large for our standard test suite. However, it is available for comparison purposes.

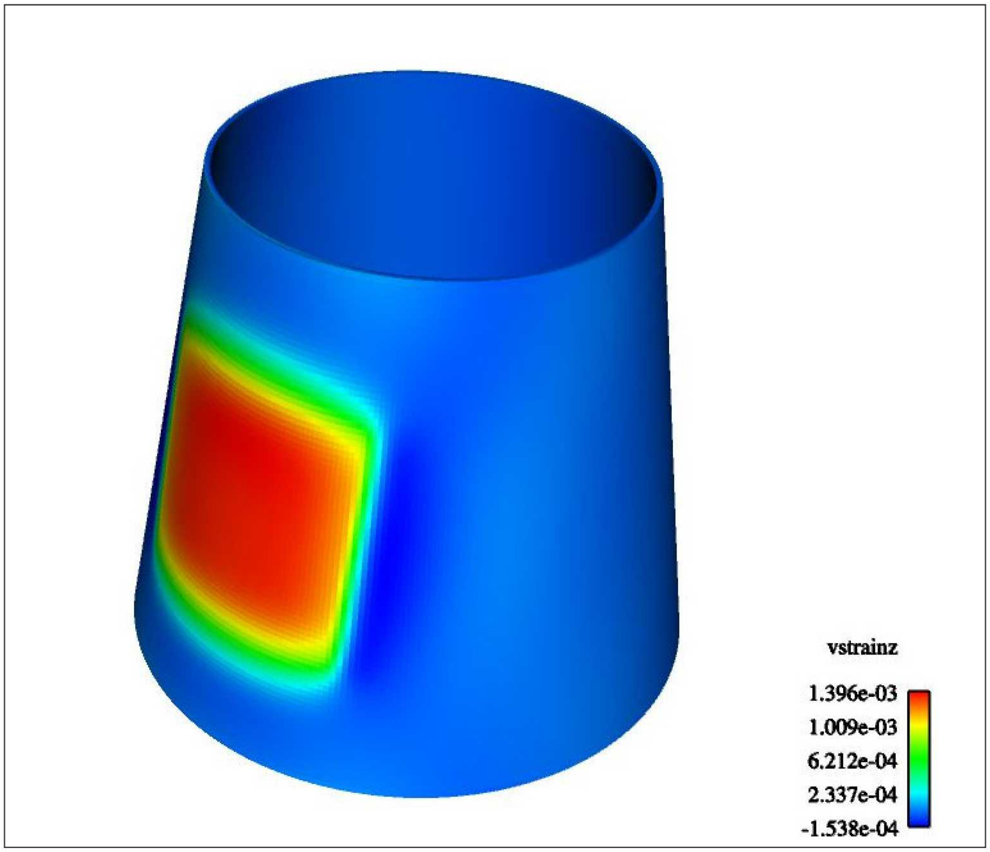


Figure 10.124: **Sierra/SD** Thermal Strains

10.8.6.2 Superelement User Verification

Superelement insertion was examined by Fernando Bitsie, the product manager for Sierra/SD, and a lead analyst in a sister organization. A nonlinear time domain analysis was used to compare results between a full model and a greatly reduced superelement model (using two superelements). Of particular concern is the generation of high frequency response. The coupling element between the two superelements is an *Iwan* element, which generates shot noise as the spring/sliders alternatively open and close. This can be amplified as it is fed into a superelement. In most respects, this is an extremely challenging test of model reduction. We do not anticipate that the high frequency response of the reduced order model will be correct, and loading may generate significant contributions in this part of the spectrum.

The full and reduced order models are illustrated in Figure 10.126. In the reduced model, the top and bottom solid sections are replaced by superelements. There are only 8 nodes in the reduced model, while the full model is composed of about 33,000 nodes.

Figure 10.127 shows the acceleration of the top and bottom as a function of time. Clearly, there is a much greater response for the superelement than for the full model. This is also illustrated in Figure 10.128, where the force across the joints is examined in the time domain for both the full and reduced models. Again, there is a significant difference.

However, examination of the response of the model in the frequency domain reveals that the differences are primarily in the high frequency. Figure 10.129 shows the frequency response of the accelerations in Figure 10.127. As seen in the figure, there is very good agreement between the models at lower frequencies. The discrepancies occur at the Nyquist frequency (50 kHz), and twice that (the sampling frequency). The response at 100 kHz is extremely strong for the CMS model, and it is this reponse which is dominating the time response.

Figure 10.130 illustrates the same issue for the loading across the *Iwan* element. Compare this with the time domain in Figure 10.128. Again, the low frequency response is reasonably accurate, while there are significant issues at the sampling frequency.

Overall, the response of the reduced order model is entirely as expected. Agreement with the full order model is not attained at higher frequencies, but at lower frequencies the agreement is good.

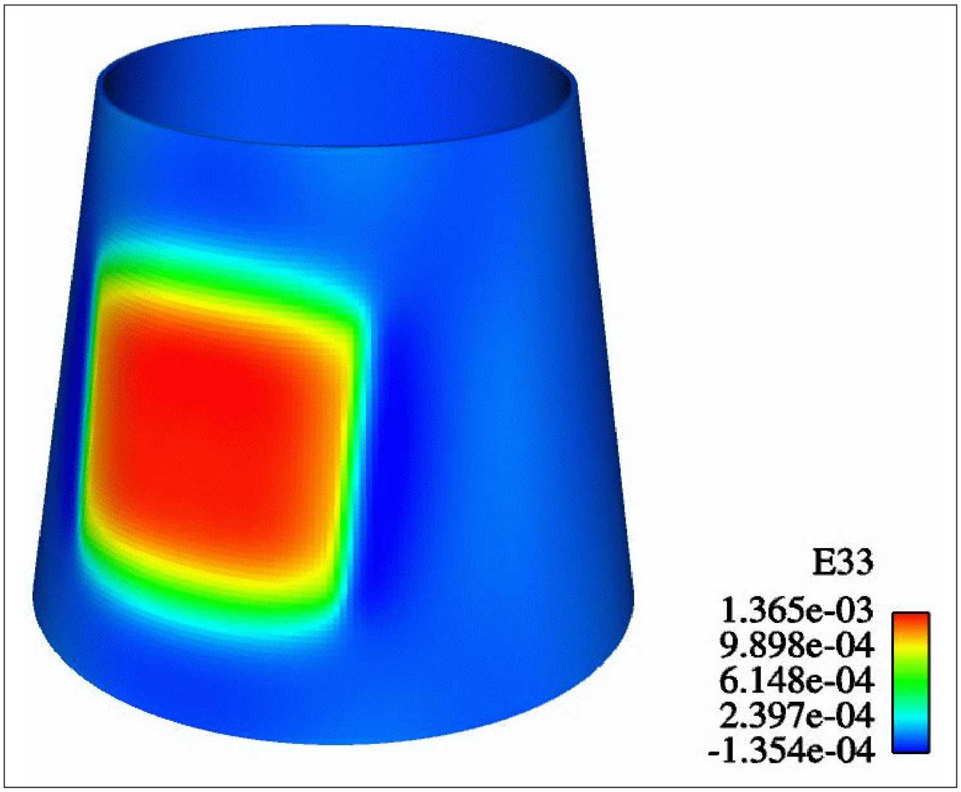


Figure 10.125: Abaqus Thermal Strains

images currently not available. under review and approval.

Figure 10.126: Exploded view of 3 Leg structure with the full model on the left and the reduced model on the right. Iwan elements connect the top and bottom structures of both models. These Iwan elements are not shown because they connect co-located nodes.

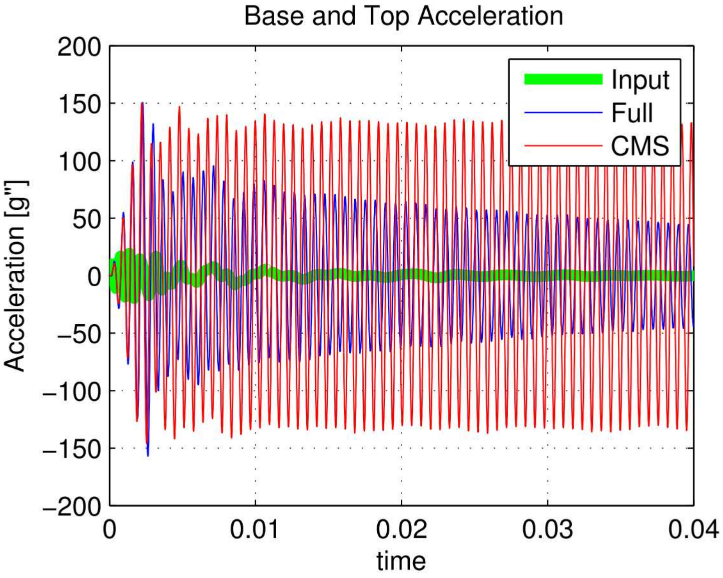


Figure 10.127: Time Domain Acceleration Response of Comparative Model

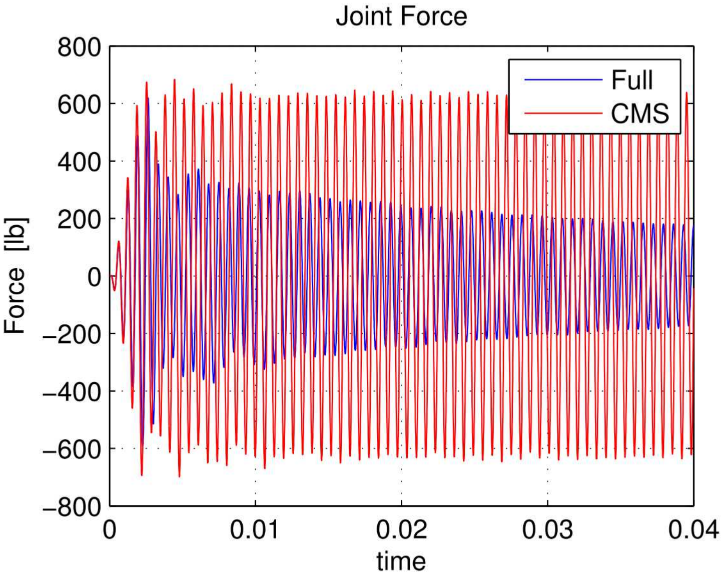


Figure 10.128: Time Domain Element Force of Comparative Model

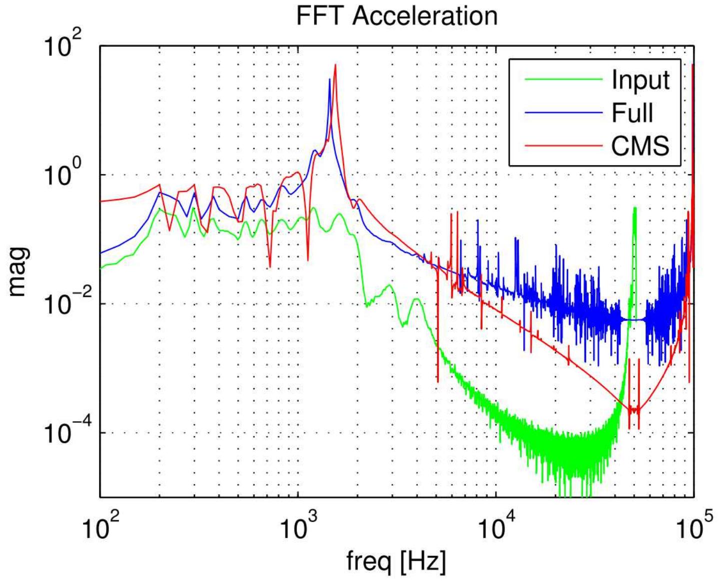


Figure 10.129: Frequency Domain Acceleration Response of Comparative Model

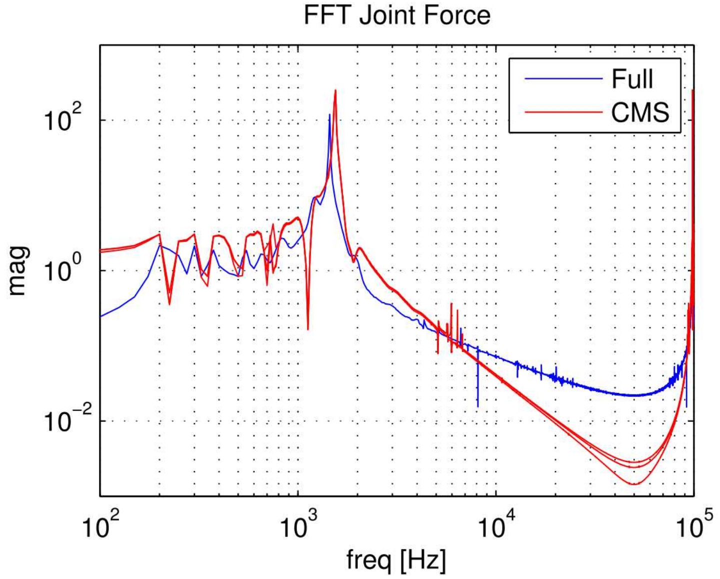


Figure 10.130: Frequency Domain Element Force of Comparative Model

10.9 Other Tests

The goal of any software verification effort is to ensure that the equations are being solved properly. This includes input and output, and to an extent documentation as well. The crucial question is whether analysts can trust the results of the calculations. *Any* test or evaluation which improves confidence in this process is of value. As stated by Myers,[19] "A good test is one that has a high probability of detecting an as-yet undetected error." Interestingly enough, the tests that catch most of our errors are emphatically *not* those that have been presented in previous sections!

10.9.1 Regression Tests

Part of the process development for **Sierra/SD** is a nightly regression test. These are typically small tests that have been assembled to examine parts of the code. These examples are usually results of either artifacts of development, or of bugs that have been identified and fixed in the code. They are in no way rigorous verification tests; instead they report only when results have changed for some reason. These changed results may be introduced by additions or changes in the software, or they may be introduced by operating system variations (including new libraries and new platforms). These regression tests are evaluated and reported on our web site nightly, and they have been responsible for identifying the vast majority of the issues in the software.

At the time of this writing (October 2003), we evaluate approximately 800 regression results. Approximately half of these tests are repeated in parallel. Detailing these tests is well beyond the scope of this document. Indeed, the nature of the regression test is different from the nature of standard verification tests, and it is not clear that we can easily break the tests down into categories that verify element formulations for example.

10.9.2 Static Tests

Static tests provide a mechanism for evaluating the software outside of the operational environment. They include source code evaluations as well as software to test our software. Source code compliance with standards as well as dangerous practices may be evaluated.

We have found limited to no value in source code evaluations in our group. They are also extremely resource demanding. As a consequence we have discontinued source code evaluations. We use source code walk through occasionally only as an aid in our understanding of the development.

However, there has been some evidence of improved software though other static tests. As specified in our *Procedures*[1] document, Some sections of the code are subjected to this type of evaluation at each release. To date, we have performed a full evaluation of all lines of the finite element portion of the code at each release.[3]

10.9.3 Dynamic Testing

Another important aspect of software testing includes memory errors in the code. As part of our release process, we run the regression tests through memory checking software before release. Typically all the regression tests are run through the software, and if the tools are available, we run through both serial and parallel tests. These tests are also run periodically through the development process. They are effective in finding bugs that are not readily apparent through other tests.

[3] In other words, we look at all the code except the third party libraries.

Appendix A

Input Decks For Verification Problems

A.1 Parallel Distribution of Load through Rbars

Refer to section 4.1

A.2 RigidSet Compared to Rbar

Refer to section 4.2

A.3 Multiple Tied-Surfaces and Curved Surfaces

Refer to section 4.3

A.4 Craig Bampton Reduction

Refer to section 3.1

A.5 Superelement Damping

Refer to section 3.2

A.6 Euler Beam Bending

Refer to section 6.1

```
SOLUTION
    solver=gdsw
statics
title 'single beam model. 100 elements. xy only'
lumped
END

FILE
geometry_file temp%1d/100.par.2.%.1d
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
//     warninglevel 0
     deform
//     eorient
END

ECHO
  // MATERIALS
  //ELEMENTS
  // JACOBIAN
//  ALL_JACOBIANS
   TIMING
  // MESH
//  INPUT
  // NODES
  // FETI_INPUT
//  DISP
//  STRAIN
  // STRESS
  // MFILE
  // mass
//none
END



BLOCK 1
material 2
Beam2
Area 0.1
orientation 0 .1 0
I1 .2
I2 .3
J .5
END
```

```
Material 2
name 'Aluminum'
E 10.0E6
nu 0.33
density 253.82e-6
END
```

A.7 Euler Beam Properties

Refer to section 6.2

A.8 A Navy Beam

Refer to section 6.3

A.9 Two Layered Hexshell

Refer to section 6.4

A.10 Spring Dashpot

Refer to section 6.21

A.10.1 General Spring Dashpot: static solution

A.10.2 General Spring Dashpot: transient solution

A.10.3 General Spring Dashpot: transient solution with coordinate transformation

A.11 Preloaded Beam

Refer to section 6.5

A.12 Partial Cylinder Patch

Refer to section 6.6

A.13 Membrane Geometrical Stiffness

Refer to section 6.7

A.14 Membrane Quad

Refer to section 6.8

A.15 QuadM membrane Patch

Refer to section 6.9

A.16 QuadS_GY Shear Membrane Shell

Refer to section 6.10

A.17 QuadS_GY Shear Membrane Shell - Geometric Stiffness and Preload

Refer to section 6.11

A.18 Hex Membrane Sandwich

Refer to section 6.12

A.19 SierraSM to SierraSD Coupling

Refer to section 3.3

A.19.1 Sierra/SD input file for SM/SD Coupled Simulation

A.20 Waterline of a ship

Refer to section 5.1

A.21 Transient Convergence

Refer to section 5.2

A.22 Modal Transient Temporal Convergence

Refer to section 5.3

A.23 Transient Restart Examples

A.23.1 Linear Transient in Step 1

Refer to section 5.4 for results of the tests.

A.23.2 Restarted Modal Transient in Step 2

A.24 Eigenvalue Restart with Virtual Nodes and Elements

Refer to section 3.4

A.25 Filter Rigid Modes from Loads

Refer to section 3.5

A.26 Q Modal Transient

Refer to section 5.5

A.27 Q Modal Frequency Response

Refer to section 5.6

A.28 Sensitivity to Parameters

Refer to section 3.6

A.29 Sensitivity Analysis with a Superelement

Refer to section 3.7

A.30 Shock Tube SI

Refer to section 3.8

A.31 Fluid Structure Interaction Added Mass

Refer to section 5.7

A.32 Fluid Structure Cavitation

Refer to section 5.8

A.33 Higher Order Hex Acoustic Element Convergence

Refer to section 6.13

A.34 Higher Order Tet Acoustic Element Convergence

Refer to section 6.14

A.35 P-elements on 1-D waveguide up to order 6

Refer to section 6.15

A.36 P-elements on Acoustic Sphere for Multiple Refined Hex-Meshes

Refer to section 6.16

A.37 P-elements on Acoustic Sphere for Multiple Refined Tet4-Meshes

Refer to section 6.17

A.38 Tied-Joint with Joint2G and Spring

Inputs for comparison of manually generated constraints with TiedJoint.

A.38.1 Manual Constraints

A.38.2 Tied Joint Constraints

Refer to section 6.18 for details of the test.

A.39 Beam CBR

Refer to section 3.9 for details of the test.

A.40 Slide RBE2. Selected DOFS

Refer to section 6.19 for details of the test.

A.41 Thin Plate Bending

Refer to section 6.20 for details of the test.

A.42 Modal Force on a Biplane Model

Refer to section 3.10 for details of the test.

A.43 Lighthill Analogy - Helmholtz Resonator

Refer to section 3.11 for details of the test.

A.44 LightHill Tensor Verification Input

Refer to section 3.12 for details of the test.

A.45 Superelement Superposition

Refer to section 3.42 for details of the test.

A.45.1 Full Model

A.45.2 CB Reduction

A.45.3 System Analysis with Superelement

A.46 Superelement Inertia Tensor Input

Refer to section 3.14 for details of the test.

beam_model

A.47 Nastran/SierraSD Interoperability with Superelements

Refer to section 3.15 for details of the test.

Sierra/SD full model

Nastran full model

A.48 Contact Verification

Refer to section 4.4 for details of the test.

A.49 Buckling of Constant Pressure Ring Input

Refer to section 5.9 for details of the test.

A.50 Rotating Dumbbell Statics

Refer to section 7.1

A.51 Rotating Beam Statics

Refer to section 7.2

A.52 Rotating Shell Statics

Refer to section 7.3

A.53 Rotating Ring Statics

Refer to section 7.4

A.54 Rotating Ring Acceleration

Refer to section 7.5

A.55 Rotating Superelement Statics

Refer to section 7.6

A.56 Rotating Superelement Beam Statics

Refer to section 7.7

A.57 Point Mass in a Rotating Frame

Refer to section 7.8

A.58 Force Identification from Structural Acoustic Frequency Responses

8.1

A.59 Force Identification from Frequency Responses

8.2

A.60 Force Identification from Temporal Pressures

8.3

A.61 Force Identification from Temporal Tractions

8.4

A.62 Force Identification from Temporal Acoustic Pressures

8.5

A.63 Force Identification with Modal Transient

8.6

A.64 Random Vibration Moments

Refer to section 9.

A.65 Fatigue Output of Single DOF in Random Vibration

Refer to section 9.2

A.65.1 Modal Random Vibration

A.65.2 Fatigue Solution

A.66 Fatigue Output of Dogbone

Refer to section 9.3

A.67 Fatigue Output of Pinned Shell

Refer to section 9.4

Appendix B

Making the Verification Document

This appendix provides instructions to developers to assist in building this reference. It is not of general use to analysts. The reference is,

```
http://sierra-trac.sandia.gov/trac/sierra/wiki/Modules/APS/VerificationDocumentGeneration
```

There are two steps. If an issue arises, and its necessary to repeat this process, it is necessary to restart from step 1 if one or more tests have changed significantly.

Step 1 is to run the tests. Remove a pre-existing `results` directory and everything below it. Be aware that even if `makeLocalDocuments.py` claims success, one or more on the individual LaTeX files may be broken.

```
bake Salinas adagio -e release
rm -rf results
assign -p Salinas_rtest -k self-documenting
testrun -e release --save-all-results
pushd results
pushd latest
module load viz seacas
export PATH=$PATH:/scratch/$USER/toolset/contrib/testTools/adagio/
makeLocalDocuments.py
```

The `makeLocalDocuments.py` step accomplishes two things. First, any necessary local scripts are run. That may include `blot` for example to generate figures. Second `pdflatex` is run locally to generate a local document. Note that `testrun ... -r wrong` is compatible.

If it is necessary to repeat the second step, and no tests have changed, then (fortunately) skip the first step and start here. Generation of a single, concatenated verification document is the last step. As many supporting files are in the docs/Salinas/doc directory, we go to that directory to run the scripts.

```
cd /scratch/$USER/docs/Salinas/doc
ln -s /scratch/$USER/code/results .
make clean
make cleantex
make snllineblk.pdf > makedoc.log
make SANDbackground.pdf >> makedoc.log
make verificationAutodoc.pdf >> makedoc.log
acroread verificationAutodoc.pdf
```

If step 2 fails, the tail of an `.aux` will point to the cause.

Finally, you may clean up that directory.

```
make cleantex
rm results
```

Note however, the verificationAutodocSrc and verificationAutodocInp files may need to be manually updated. There is a tool to help! Use “gatherLocalTests.sh” to generate a list of all tests in the results directory. These are in the right format to be added to verificationAutodocSrc, but must be copied over by hand. I’ve also recently found that the graphicspath should be terminated with “/”, and not with a space. LaTeX is picky about that.

Appendix C

Richardson Extrapolation

Richardson extrapolation[21] is a numerical technique whereby the convergence of a solution is identified and used to provide an improved accuracy solution. We here discuss this technique as applied to a finite element model.*

Assume that an exact solution, a_o is sought and that the mesh with a characteristic element length h is within the region of geometric convergence.† The solution, a_o, may be an eigenvalue for example. In that region, the error may be written,

$$err_h = a(h) - a_o = C\,h^n \tag{C.1}$$

where C and n are unknown.

Take another mesh of characteristic element size, αh.‡

$$a(\alpha h) - a_o = C\alpha^n h^n \tag{C.2}$$

We further refine the mesh.

$$a(\alpha^2 h) - a_o = C\alpha^{2n} h^n \tag{C.3}$$

There are thus three equations to solve for the three unknowns, C, n and a_o.

$$a(h) - a(\alpha h) = Ch^n - \alpha^n Ch^n = Ch^n(1-\alpha^n) \tag{C.4}$$

$$a(\alpha h) - a(\alpha^2 h) = \alpha^n Ch^n - \alpha^{2n} Ch^n = Ch^n \alpha^n (1-\alpha^n) \tag{C.5}$$

Thus,

$$\alpha^n = \frac{a(\alpha h) - a(\alpha^2 h)}{a(h) - a(\alpha h)} \tag{C.6}$$

And,

$$n = \frac{\log\left(a(\alpha h) - a(\alpha^2 h)\right) - \log\left(a(h) - a(\alpha h)\right)}{\log(\alpha)} \tag{C.7}$$

Knowing n, we solve for a_o.

$$a_o = \frac{a(\alpha h) - \alpha^n a(h)}{1-\alpha^n} \tag{C.8}$$

Having a_o, one may plot $a(h) - a_o$ versus mesh size on a log-log plot and achieve a line. A fourth mesh is necessary to confirm that we are in the region of geometric convergence.

* Richardson extrapolation was first developed in 1910. It is a well established technique. This description is based on notes from Dan Segalman.

† The region of geometric convergence is that part of the solution where the error is decreasing monotonically, and may be well represented by a decaying exponential. Richardson's extrapolation allows an approach either from above or below (i.e. the error may have either sign). This write up describes convergence from above.

‡ Usually we take $\alpha = 1/2$, but other values are sometimes useful. Also, the mesh need not be uniform, but the mesh does need to be scaled uniformly. For example, slicing each element in half in each dimension does result in a uniform refinement with $\alpha = 0.5$.

The extrapolation must be performed using the FEM predictions at a node or element center which does not change spatial location during mesh refinement. If a nodal variable is chosen, $1/\alpha$ will be even (most likely 2). For element centroids, odd values of $1/\alpha$ are needed so the element centroid does not move during refinement.

Richardson extrapolation is valuable not only because it provides an improved estimate for a_o, but also because it provides a formal means of determining the rate of convergence, n. Typically *a priori* estimates for this rate exist. While it is not practical to accomplish 4 levels of mesh refinement on most real models, the technique can be valuable for determining the convergence rates of simpler examples.

Appendix D

Legacy Test Matrix

The following tables identifies the verification tests for Sierra/SD, and provides a cross reference between the descriptions in this document and the tests run. Tests are found in two major test systems. The `Salinas/test_tool` tests contain the regression tests, and some of the verification tests. The `Salinas_tests` directory contains the remainder of the tests.

Table D.1: Test Matrix

Dir/Name of Test	Doc. Table	Row	Ref Table	Element Type
beam_analytic/cantilever_free_beam2_test	10.57	2	*3 1a*	Beam2
./cantilever_free_tria3_test	10.57	3	*3 1a*	Tria3
./cantilever_free_tria3r_test	10.57	4	*3 1a*	Tria3⊥
./simply_simply_beam2_test	10.57	7	*3 1e*	Beam2
./simply_simply_tria3_test	10.57	8	*3 1a*	Tria3
./simply_simply_tria3r_test	10.57	9	*3 1a*	Tria3⊥
./cantilever_guided_beam2_test	10.57	11	*3.3b*	Beam2
beam-curved/roark_table17_1_test	10.58	2	*17.1*	Beam2
./roark_table17_2_test	10.58	4	*17.1*	Beam2
./roark_table17_3_test	10.58	6	*17.1*	Beam2
beam_eigen/free_free_test	10.65	2	*8-1.1*	Beam2
beam_eigen/free_sliding_test	10.65	4	*8-1.2*	Beam2
beam_eigen/clamped_free_test	10.65	6	*8-1.3*	Beam2
beam_eigen/pinned_pinned_test	10.65	8	*8-1.5*	Beam2
beam-mass/blevins_table6-2_19_test	10.64	2	*6-2.19*	Beam2
beam-mass/blevins_table6-2_20_test	10.64	4	*6-2.20*	Beam2
beam-mass/blevins_table6-2_22_test	10.64	6	*6-2.22*	Beam2
plate_annular/roark_table24_1a_test	10.59	2	*24.1a*	Tria3
plate_annular/roark_table24_1b_test	10.59	4	*24.1b*	Tria3
plate_annular/roark_table24_1e_test	10.59	6	*24.1e*	Tria3
plate_rectangular/roark_table26_1a_test	10.60	3	*26.1a*	QuadT
plate_rectangular/roark_table26_1a_t_test	10.60	2	*26.1a*	Tria3
plate_rectangular/roark_table26_8a_test	10.60	6	*26.8a*	QuadT
plate_rectangular/roark_table26_8a_t_test	10.60	5	*26.8a*	Tria3
spring-mass/blevins_table6-2_2_test	10.63	2	*6-2.2*	spring
spring-mass/blevins_table6-2_18_test	10.63	4	*6-2.18*	spring

Table D.2: Test Matrix (cont)

Dir/Name of Test	**Doc. Table**	**Row**	**Ref Table**	**Element Type**
thinShellsOfRevolution/.				
./roark_table28_1a_hex8_test	10.61	3	*28.1a*	Hex8
./roark_table28_1a_tria3_test	10.61	2	*28.1a*	Tria3
./roark_table28_1b_hex8_test	10.61	6	*28.1b*	Hex8
./roark_table28_1b_tria3_test	10.61	5	*28.1b*	Tria3
shaft/fixed_free_beam2_test	10.66	2	*8-19.2*	Beam2
shaft/fixed_free_hex8_test	10.66	3	*8-19.2*	Hex8
plate_eigen_circ/free_test	10.68	2	*11-1.1*	QuadT
plate_eigen_circ/simple_test	10.68	4	*11-1.2*	QuadT
plate_eigen_circ/clamped_test	10.68	6	*11-1.3*	QuadT
plate_eigen_circ/clamped_mass_test	10.68	8	*11-1.12*	QuadT
plate_eigen_rect/all_edges_free_test	10.69	2	*11-4.1*	Tria3
plate_eigen_rect/all_edges_fixed_test	10.69	4	*11.4.21*	Tria3
plate_eigen_rect/all_edges_simple_test	10.69	6	*11-4.16*	Tria3
plate_eigen_rect/sFixed_lFree_sSS_lFree_test	10.69	8	*11-4.6*	Tria3

References

[1] Reese, G., Bhardwaj, M., Segalman, D., Pierson, K., and Walsh, T., "Salinas – Procedures," 2011.

[2] Bourcheron, E. A., Drake, R. R., Edwards, H. C., Ellis, M. A., Forsythe, C. A., Heaphy, R., Hodges, A. L., Pavlakos, C., Schofield, J. R., Sturtevand, J. E., and Williamson, C. M., "Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan Part 1: ASC Software Quality Engineering Practices Version 1.0," Tech. Rep. SAND2004-6602, Sandia National Laboratories, January 2005.

[3] Zepper, J. D., Aragon, K., Ellis, M., Ble, K., and Eaton, D., "ASCI Applications Software Quality Engineering Practices," Tech. Rep. SAND2002-0121, Sandia National Laboratories, January 2002.

[4] S D Team, "Sierra Structural Dynamics Users' Notes," 2015.

[5] S D Team, "Sierra/SD Theory Manual," 2015.

[6] Martin, R., *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008.

[7] Software, M., *MSC Nastran 2017 Superelements User's Guide*, 2016.

[8] Blevins, R. D., *Formulas for Natural Frequency and Mode Shape*, Krieger, Malabar, FL, USA, 1984.

[9] Bleich, H. H. and Sandler, I. S., "Interaction between structures and bilinear fluids," *International J. Solids and Structures*, **vol. 6**, no. 5, 1970, pp. 617–639.

[10] Felippa, C. and DeRuntz, J., "Finite Element Analysis of Shock-Induced Hull Cavitation," *Computer Meth. in Appl. Mech. Eng.*, **vol. 44**, 1984, pp. 297–337.

[11] Timoshenko, S., *Theory of Elastic Stability*, McGraw Hill Book Company, 1936.

[12] Felippa, C. A., "The SS8 Solid-Shell Element: Formulation and a Mathematica Implementation," Tech. Rep. CU-CAS-02-03, Univ. Colo. at Boulder, 2002.

[13] Felippa, C. A., "The SS8 Solid-Shell Element: A Fortran Implementation," Tech. Rep. CU-CAS-02-04, Univ. Colo. at Boulder, 2002.

[14] Segalman, D. J., Fulcher, C. W., Reese, G. M., and Field, Jr., R. V., "An Efficient Method for Calculating RMS Von Mises Stress in a Random Vibration Environment," *Journal of Sound and Vibration*, **vol. 230**, no. 2, 2000, pp. 393–410.

[15] Anes, V., Reis, L., Li, B., Fonte, M., and de Freitas, M., "New approach for analysis of complex multiaxial loading paths," *International Journal of Fatigue*, **vol. 62**, 2014, pp. 21–33.

[16] Martinez, D. R. and Reese, G. M., "Structural Dynamics Computational Plan," Tech. Rep. Unpublished, Sandia National Laboratories, 1996.

[17] Reese, G. M., "Salinas - RV Modeling," Tech. Rep. Unpublished Presentation, Sandia National Laboratories, October 1998.

[18] Beizer, B., *Software Testing Techniques*, Intnl Thompson Computer Press, 1990.

[19] Myers, G. J., *The Art of Software Testing*, John Wiley & Sons, 1979.

[20] MacNeal, R. H. and Harder, R. L., "A Proposed Standard Set of Problems to Test Finite Element Accuracy," *Finite Elements in Analysis and Design*, **vol. 1**, June 1985, pp. 320.

[21] Richardson, L. F., "The Approximate Arithmetic Solution by Finite Differences of Physical Problems Involving Differential Equations, With Applications to the Stresses in a Masonry Dam," *Philosophical Transactions of the Royal Society of London*, **vol. 210**, 1910, pp. 307–357.

[22] Alvin, K. F., "A method for Treating Discretization Error in Nondeterministic Analysis," *AIAA*, **vol. 99**, no. 1611, 1999.

[23] Young, W. C., *Roark's Formulas for Stress & Strain*, McGraw-Hill Book Company, 6th edn., 1989.

[24] Timoshenko, S., *Strength of Materials*, D. Van Nostrand Company, Inc., third edn., 1955.

[25] Allman, D. J., "A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Problems," *Computers and Structures*, **vol. 19**, no. 1-2, 1996, pp. 1–8.

[26] Batoz, J.-L., Bathe, K.-J., and Ho, L.-W., "A Study of Three-Node Triangular Plate Bending Elements," *Int. J. Numer. Meth. Engng.*, **vol. 15**, 1980, pp. 1771–1812.

[27] Ertas, A., Krafcik, J. T., and Ekwaro-Osire, S., "Explicit Formulation of an Anisotropic Allman/DKT 3-Node Thin Triangular Flat Shell Elements," *Composite Material Technololgy*, **vol. 37**, 1991, pp. 249–255.

[28] A., E., Krafcik, J. T., and Ekwaro-Osire, S., "Performance Of An Anisotropic Allman/DKT 3-Node Thing Triangular Flat Shell Element," *Composite Engineering*, **vol. 2**, no. 4, 1992, pp. 269–280.

[29] Hammerand, D. C., "Laminated Composites Modeling In Adagio/Presto," Tech. Rep. SAND2004-2143, Sandia National Laboratories, May 2004.

[30] Segalman, D. J., "A Four-Parameter Iwan Model for Lap-Type Joints," Tech. Rep. SAND 2002-3828, Sandia National Laboratories, November 2002.

[31] Kinsler, Frey, Coppens, and Sanders, *Fundamentals of Acoustics*, John Wiley & Sons, 1982.

[32] Moyer, T., Stergiou, J., Reese, G., Luton, J., and Abboud, N., "Navy Enhanced Sierra Mechanics (NESM): Toolbox for Predicting Navy Shock and Damage," *Computing in Science and Engineering*, **vol. 18**, no. 6, 2016, pp. 10–18.

[33] Post, D., Newmeyer, K., Landsberg, S., Shull, F., and Sundt, S., "The Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program," *Computing in Science and Engineering*, **vol. 18**, no. 6, 2016, pp. 7–9.

[34] Pierce, A., *Acoustics: An Introduction to Its Physical Principles and Applications*, ASA, 1989.

[35] S D Team, "Sierra Structural Dynamics - Theory Manual," Technical Report SAND2015-9131, Sandia National Laboratory, PO Box 5800, Albuquerque, NM 87185-5800, October 2015.

[36] Dohrmann, C., Key, S., and Heinstein, M., "A Method for Connecting Dissimilar Finite Element Meshes in Two Dimensions," *Int. J. Numer. Meth. Engng.*, **vol. 48**, 2000, pp. 655–678.

[37] Dohrmann, C., Key, S., and Heinstein, M., "Methods for Connecting Dissimilar Three-Dimensional Finite Element Meshes," *Int. J. Numer. Meth. Engng.*, **vol. 47**, 2000, pp. 1057–1080.

[38] Everstine, G. C. and Henderson, F. M., "Coupled finite element/boundary element approach for fluid/structure interaction," *JASA*, **vol. 87**, no. 5, 1990, pp. 1938–1947.

[39] Corp., L. S. T., "USA LS-DynaV̊erification & Validation Manual, USA Release 7.3," June 2010.

[40] Huang, H., "Transient Interaction of Plane Acoustic Waves with a Spherical Elastic Shell," *Journal of the Acoustical Society of America*, **vol. 45**, no. 3, 1969, pp. 661–670.

[41] Sprague, M. and Geers, T., "Response of Empty and Fluid-Filled, Submerged Spherical Shells to Plane and Spherical, Step-Exponential Acoustic Waves," *Shock and Vibration*, **vol. 6**, 1999, pp. 147–157.

[42] Morse and Ingard, *Theoretical Acoustics*, Mcgraw-Hill Book Company, 1968.

[43] Hoffelner, J., Landes, H., Kaltenbacher, M., and Lerch, R., "Finite Element Simulation of Nonlinear Wave Propagation in Thermoviscous Fluids Including Dissipation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, **vol. 48**, no. 3, 2001, pp. 779–786.

[44] Hoffelner, J., Landes, H., and Lerch, R., "Calculation of Acoustic Streaming Velocity and Radiation Force Based on Finite Element Simulations of Nonlinear Wave Propagation," in *Proceedings of IEEE Ultrasonics Symposium*, vol. 1, 2000, pp. 585–588.

[45] Kuznetsov, V. P., "Equations of Nonlinear Acoustics," *Sov. Phys. Acoust.*, **vol. 16**, 1971, pp. 467–470.

[46] Hamilton, M. F. and D. T. Blackstock, E., *Nonlinear Acoustics*, Academic Press, 1998.

[47] Blackstock, D., "Connection between the Fay and Fubini Solutions for Plane Sound Waves of Finite Amplitude," *JASA*, **vol. 39**, 1966, pp. 1019–1026.

[48] Chung, J. and Hulbert, G., "A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized alpha method," *Journal of Applied Mechanics*, **vol. 60**, 1993, pp. 371–375.

[49] Craig, R. R., *Structural Dynamics: An Introduction to Computer Methods*, John Wiley & Sons, 1981.

[50] Weaver, Jr. , W. and Timoshenko, S. P., *Vibration Problems in Engineering*, John Wiley and Sons, fifth edn., 1990.

[51] Dickens, J., Nagawa, J., and Wittbrodt, M., "A critique of mode acceleration and modal truncation augmentation methods for modal response analysis," *Computers and Structures*, **vol. 62**, no. 6, 1997, pp. 985–998.

[52] Blakely, K., *MSC/NASTRAN Users Guide: Basic Dynamic Analysis*, vol. 68, The MacNeal-Schwendler Corporation, 1993.

[53] Meriam, J. L. and Kraige, L. G., *Engineering Mechanics Volume 2 - Dynamics*, John Wiley & Sons, 1986.

[54] Trucano, T. G., Pilch, M., and Oberkampf, W. L., "On the role of Code Comparisons in Verification and Validation," Tech. Rep. SAND 2003-2752, Sandia National Laboratories, August 2003.

[55] Simmermacher, T. and Morganthaler, D., "Lockheed Martin Shared Vision Virtual Testing Final Report," Tech. Rep. Internal Report, Sandia National Laboratories, August 2003.

[56] Rohl, P. J. and Reese, G., "A Practical Look at the Massively Parallel Code Salinas," *Presented at the 47nd AIAA/ASME/ASCE/AHS/ASC SDM*, April 2006.

DISTRIBUTION:

1	MS 0380	M. J. Skroch, 1542
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories