

SANDIA REPORT

SAND2010-7746
Unlimited Release
Printed 11/2010

Coupled Thermomechanical Modeling using Dissimilar Geometries in Arpeggio

Timothy D. Kostka, Jeremy A. Templeton

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

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

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>



Coupled Thermomechanical Modeling using Dissimilar Geometries in Arpeggio

Timothy D. Kostka
Multi-physics Modeling and Simulation
Sandia National Laboratories
P.O. Box 969
Livermore, CA 94550-0969
tdkostk@sandia.gov

Jeremy A. Templeton
Thermal/Fluid Science and Engineering
Sandia National Laboratories
P.O. Box 969
Livermore, CA 94550-0969
jatempl@sandia.gov

Abstract

Performing coupled thermomechanical simulations is becoming an increasingly important aspect of nuclear weapon (NW) safety assessments in abnormal thermal environments. While such capabilities exist in SIERRA, they have thus far been used only in a limited sense to investigate NW safety themes. An important limiting factor is the difficulty associated with developing geometries and meshes appropriate for both thermal and mechanical finite element models, which has limited thermomechanical analysis to simplified configurations. This work addresses the issue of how to perform coupled analyses on models where the underlying geometries and associated meshes are different and tailored to their relevant physics. Such an approach will reduce the model building effort and enable previously developed single-physics models to be leveraged in future coupled simulations. A combined-environment approach is presented in this report using SIERRA tools, with quantitative comparisons made between different options in SIERRA.

Acknowledgment

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000. Funding for this project was provided by the Advanced Strategic Computing Program Verification and Validation program. This work would not have been possible without interactions with the Aria development team, including their implementation of new features. In particular, the contributions of R. Baca, T. Okusanya, S. Subia, and G. Wagner are greatly appreciated. The authors are grateful to F. Dempsey and S. Subia for providing helpful comments on a draft of this manuscript.

Contents

- 1 Introduction** **9**

- 2 Modeling procedure** **11**
 - Coupling scheme 11
 - Meshes 13
 - Boundary conditions 14
 - Thermal loading 14
 - Mechanical loading 14
 - Field transfers 14
 - Foam decomposition 15

- 3 Capabilities and results** **17**
 - Displacement transfer 17
 - Alternate method for displacements 19
 - Temperature transfer 22
 - Using the PROJECT operator 22
 - Foam decomposition 24
 - Encore quantitative comparisons 25
 - Convergence analysis 26
 - Methods comparisons 28
 - Computational cost of coupled simulations 30

- 4 Conclusions** **33**

Appendix

A Sample Arpeggio input deck	35
B Sample Encore input deck	41
References	43

List of Figures

2.1	The thermal and mechanical parts to the simple model.	12
2.2	A flowchart of the coupling process.	12
2.3	The thermal and mechanical meshes.	13
3.1	The conceptual idea of transferring displacements and solving for temperature.	18
3.2	The original transfer of displacements.	18
3.3	An alternate method for obtaining internal foam displacements in Aria.	19
3.4	The resulting displacements when letting Aria solve for the displacement of the internal foam parts.	20
3.5	The revised timestep solution order to have displacements calculated in Aria and Adagio match in the output file.	21
3.6	The resulting displacements from the alternate method when solving Adagio first, then Aria.	21
3.7	A schematic of the temperature transfer operation.	22
3.8	The transfer of temperature while using the TRUNCATE operator.	23
3.9	The transfer of temperature while using the PROJECT operator.	23
3.10	The internal pressure created by foam decomposition causing the walls to bow out. Temperature is in Kelvin.	24
3.11	The temperature field and effective stress field obtained for the most refined mesh.	25
3.12	A transfer of the logarithmic strain field from a coarser mesh onto a finer mesh.	27
3.13	A convergence analysis of each metric on a series of mesh refinements.	27
3.14	The error in foam displacement for the tets run and the encompassing run.	30
3.15	The error in external feature temperature on the external feature for the dissimilar run and the project run.	31

Chapter 1

Introduction

This report summarizes our efforts on running a coupled thermomechanical analysis using the Arpeggio code. Historically, problems of interest containing both thermal and mechanical aspects have usually been run separately. For example, the thermal problem would be analyzed using an initial temperature distribution, appropriate boundary conditions (radiation, convection), and conduction throughout the body. In parallel, the mechanical problem would also be analyzed using prescribed displacements or velocities and appropriate material models to solve for the displacements and evaluate the mechanical response. Coupling between these two physics types was often extremely limited, as for example, has been done for electrostatic interactions with structures [5]. For example, current state-of-the-art mechanical simulations typically adjust material parameters based on temperature or including adiabatic heating, but conduction, convection and radiation can not take place as the heat diffusion equation is not solved.

Coupling the thermal and mechanical simulations has the potential to increase the fidelity of both of their respective results. On the thermal side, allowing for displacement can alter how long it takes for certain parts to heat up. In addition, as parts are deformed, contact surfaces which are initially touching may separate and new ones may form which would alter the flow of heat into the material. On the mechanical side, the temperature field may be used to alter the mechanical properties.

In this report, we have explored a number of ways of dealing with dissimilarities between the thermal and mechanical meshes and quantified the error produced in each. Schemes which attempt to extrapolate displacement to internal parts tended to perform poorly, with large discontinuities developing, which often lead to inverted elements. Alternative schemes which calculate displacement of these internal parts fared much better. Methods for extrapolating temperature performed well, with modest errors in the temperature fields. In addition, this work was used to identify technology gaps in SIERRA impeding the widespread use of these simulation techniques. Consequently, as a result of this work, a new extrapolation scheme, PROJECT was added to the codebase.

Chapter 2

Modeling procedure

In this chapter, we describe the tools and framework used to run the simulation. The problem being solved is that of a simplified model which was created to mimic the features and behavior relevant to more complicated models. By focusing on a simplified model, we are able to examine issues and develop solutions in a more controlled environment. Such an approach is advantageous when using many of the SIERRA Framework utilities outside of their previously exercised bounds. Of particular interest in this work is the behavior of interpolation and extrapolation operators between disparate geometries. By being able to easily identify these differences in the model, and communicate them with the SIERRA development team, the end goals of assessing and improving the code for thermomechanical simulation can be facilitated.

The geometry of the simplified model can be seen in Figure 2.1 with the thermal model on the left and the mechanical model on the right. Both of these models include the outer shell and an access door which are composed of stainless steel. The geometry of these parts matches exactly. In addition, the thermal model includes two foam blocks in the interior. These blocks are deemed important to the thermal response, but negligible to the mechanical response, as they cannot carry stresses on the same order of magnitude as the outer steel shell. For this reason, the mechanical model does not include these foam blocks. The mechanical model includes a small external feature meant to represent a fixture which may be important to the mechanical response, but negligible to the thermal response and, therefore, is not included in the thermal model.

Coupling scheme

The equations we are solving are loosely coupled using the Sierra code Arpeggio, which uses the Sierra Framework Transfer utility [2] to exchange information between the thermal/fluids code Aria and the quasistatic solid mechanics code Adagio using transfer operators [6]. Within a given timestep, the thermal equations are first solved implicitly using Aria. The temperature is then transferred to the structural mesh and the structural equations are then solved implicitly using Adagio. After this step, the displacements are transferred to Aria and the scheme repeats every timestep until the simulation is completed. A visual representation of this process can be seen in Figure 2.2.

It should be noted that although the equations are coupled, we are not solving them simultaneously, nor are we using a staggered scheme on a single timestep to converge. In this sense, we say the scheme is *loosely coupled*. To fully couple the equations, we would need a closed loop within the timestep solution to ensure convergence has occurred. Although the current scheme is only loosely coupled, it represents a significant step forward compared to running the two problems separately.

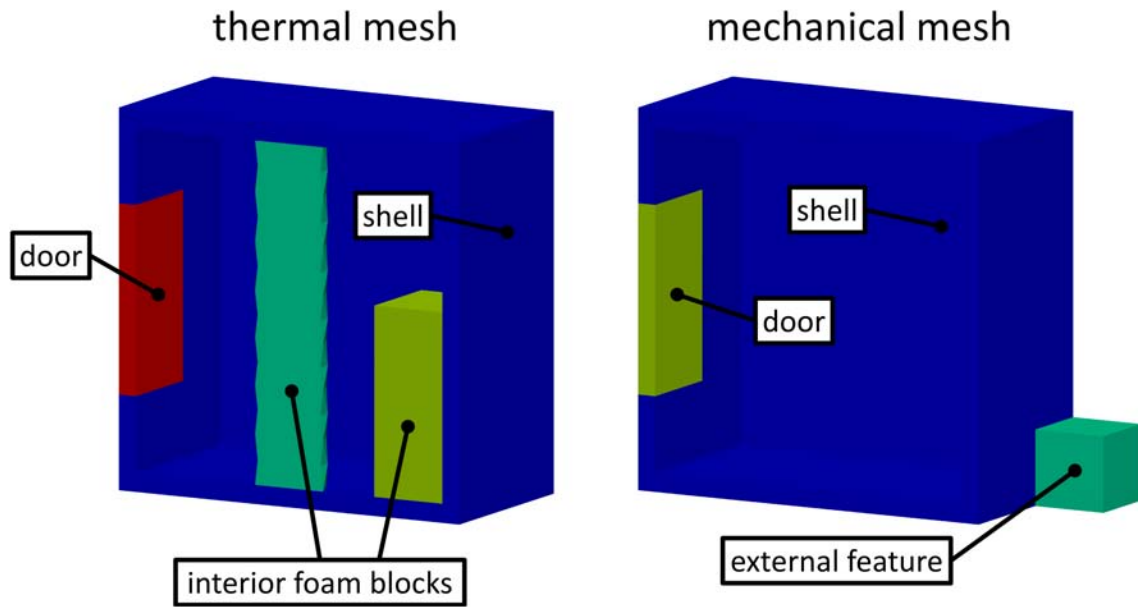


Figure 2.1. The thermal (left) and mechanical (right) parts to the simple model.

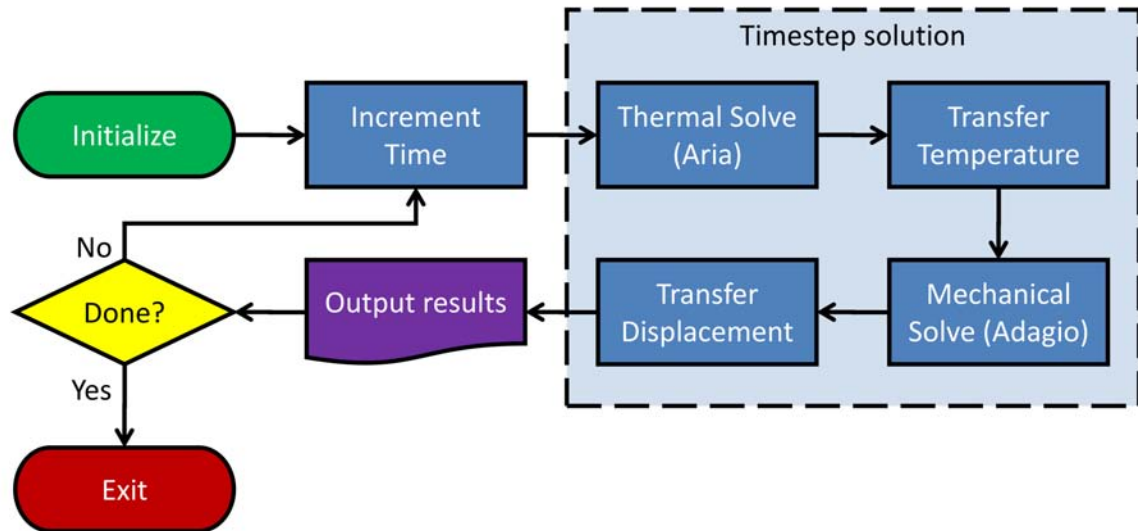


Figure 2.2. A flowchart of the coupling process.

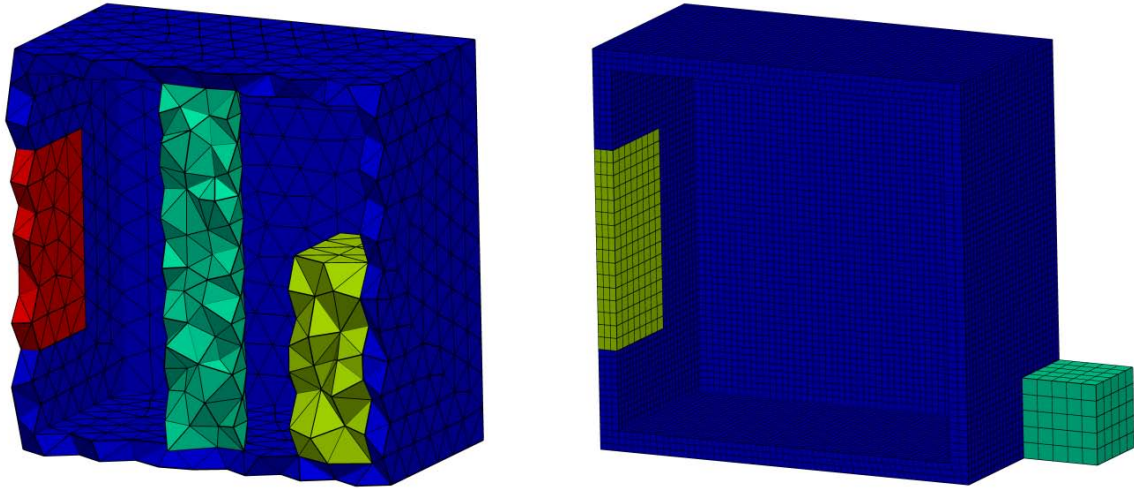


Figure 2.3. The thermal (left) and mechanical (right) meshes.

Additionally, as we shall see in the results section, the thermal problem is only weakly influenced by the mechanical problem. Thus, the effect of the coupling is mostly one-way.

In order to have a tighter coupling (i.e. converge to a solution for both equations simultaneously), one is left with several options. Aria could be used to solve both equations, provided the appropriate material models and capabilities are implemented into Aria. A significant drawback with that approach is that capabilities in Aria have not been used nearly as extensively as those in Adagio, and some advanced mechanical simulation capabilities, such as contact and plastic deformation, are not present. Alternatively, one could use a closed loop staggering scheme to ensure the solution is converged before moving on to the next timestep. This method has the advantage of being able to use the existing uncoupled codes, but it has less favorable convergence behavior.

Meshes

A cropped view of the meshes can be seen in Figure 2.3. The thermal model (left) is composed of 6,178 linear tet elements, and nodes between different parts are merged. By merging nodes, the heat conduction path between different parts is taken care of automatically without the need for defining each thermal contact separately. This also means there is zero thermal contact resistance between parts. The mechanical mesh (right) is composed of 50,691 linear hex elements, with each part meshed separately. The parts are kept together by using a tied contact surface at each interface.

The use of tet elements in the thermal mesh is largely due to the robustness with which meshes can be generated for arbitrary geometries. The meshing process is mostly automatic. However, these elements do not perform well for mechanical simulations, which is why hex elements were used. The generation of a mechanical mesh is often a time consuming process.

Boundary conditions

To replicate many of the target applications, the boundary conditions are chosen to approximate placing the assembly into an engulfing fire. Therefore, the loading is thermally driven, with the mechanical response being driven indirectly through thermal strains and thermally-driven pressure buildup.

Thermal loading

The primary thermal loading is through radiation on the outside surface of the assembly. Heat is transferred internally through conduction and internal enclosure radiation. The thermal boundary and initial conditions can be summarized as:

- Initial temperature condition of 300 K
- Radiation between the exterior surface and environment at 1280 K,

with internal heat transfer mitigated by:

- Radiation inside internal enclosure
- Conduction between different parts

Mechanical loading

Mechanically, the only loading conditions are those which are applied indirectly as a result of temperature changes. For example, stainless steel expands when the temperature increases, so under thermal loading conditions gradients in the temperature will produce stresses within the material. In addition, the structural material models have properties which vary with temperature which can also induce a corresponding stress. The boundary and initial conditions on the structural side can be summarized by:

- Indirect loading due to thermal expansion
- Interior pressure condition due to foam decomposition and elevated temperatures

Field transfers

As was discussed earlier, we are using Arpeggio to perform a loosely coupled thermomechanical simulation using dissimilar meshes. In order to use the temperature field as an input for the mechanical equations, we employ the transfer operators found in the Sierra framework. These operators can take a field defined on one mesh and reproduce it on a separate mesh. This capability is what allows Adagio to use a temperature field obtained by Aria.

For example, the interpolation of the temperature field from Aria to Adagio is done through the transfer block

```
Begin Transfer aria_to_adagio
  Interpolate Volume Nodes From aria_region To adagio_region
  Send Field solution->temperature State New To temperature State New
  Nodes Outside Region = Extrapolate
End
```

which is called by the solution procedure block. This transfer block is also able to extrapolate values for regions which do not exist in the sending mesh. We shall discuss this later.

Foam decomposition

A significant portion of the interior volume contains foam which, when heated, decomposes to produce CO₂ gas as one of the byproducts. Gas pressure, combined with increasing thermal stresses, drives the deformation of the system. For this reason, we have developed a model to track the decomposition of the interior foam and calculate the resulting pressure [3]. The module does this by (a) calculating the percentage of foam that has decomposed in each element as a function of temperature, (b) calculating the internal volume, and (c) calculating the pressure based on the ideal gas law.

Including the physics of foam decomposition is important to us as it contributes significantly to the mechanical boundary conditions. At sufficiently high temperatures, the pressure generated internally has the potential to cause plastic deformation and potentially failure. For this reason, it is essential that we model this phenomenon.

This capability was implemented as a user subroutine in Aria. Results are calculated as global variables which are then used to drive the internal pressure in Adagio.

Chapter 3

Capabilities and results

This chapter details the capabilities we have established along with the issues we encountered and the proposed solutions.

Displacement transfer

While running the first few iterations of a more complicated model, we would experience run failure due to inverted elements in the Aria mesh. As inverted elements yield a negative mass and lead to a non positive definite stiffness matrix (not to mention the non-physical implications), the simulation aborts at this point. Upon investigating this issue, it was found that inverted elements occur at locations in the thermal mesh which do not match the geometry of the mechanical mesh. Thus, the transfer operator must use extrapolation to determine values for the displacement at these points.

Our initial process for transferring displacements and then solving for temperature can be seen in Figure 3.1. The displacement in the enclosing shell is interpolated (since this part has an identical geometry in both models), while the displacements in the internal foam must be extrapolated. Initially, we were using the TRUNCATE extrapolation operator in SIERRA to perform this. This operator works on the following basis: For nodes on the receiving mesh lying outside the geometry of the sending mesh, it projects the node to the nearest surface on the receiving mesh, and uses the value at that point. It performs this in parametric (master element) space.

The displacement field resulting from this transfer can be seen in Figure 3.2. As displacement is a vector field, we have plotted the magnitude instead of a particular component. Within the shell, the displacement field matches very well. No qualitative difference can be seen between the two. While there is a small difference, this is simply a result of the coarse thermal mesh not being able to represent a field as accurately as the finer mechanical mesh.

However, within the internal foam blocks, instead of the displacement field being smooth, one can see a number of local variations. Since this simulation is quasistatic, one would expect a much smoother displacement field which agrees with that of the mechanical mesh. The displacements seen in this figure are orders of magnitude less than length scales in the mesh, so they do not cause issues. However, as the simulation progresses, the displacement field can grow and these local variations can lead to inverted elements and run failure.

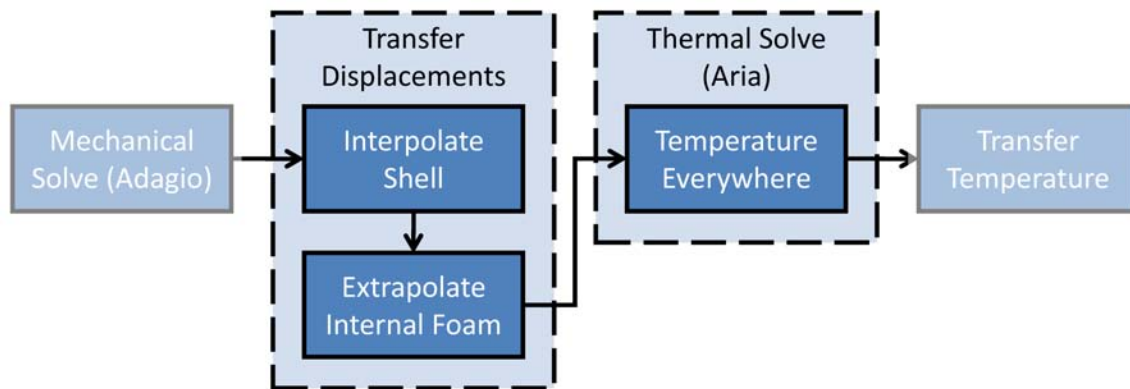


Figure 3.1. The conceptual idea of transferring displacements and solving for temperature.

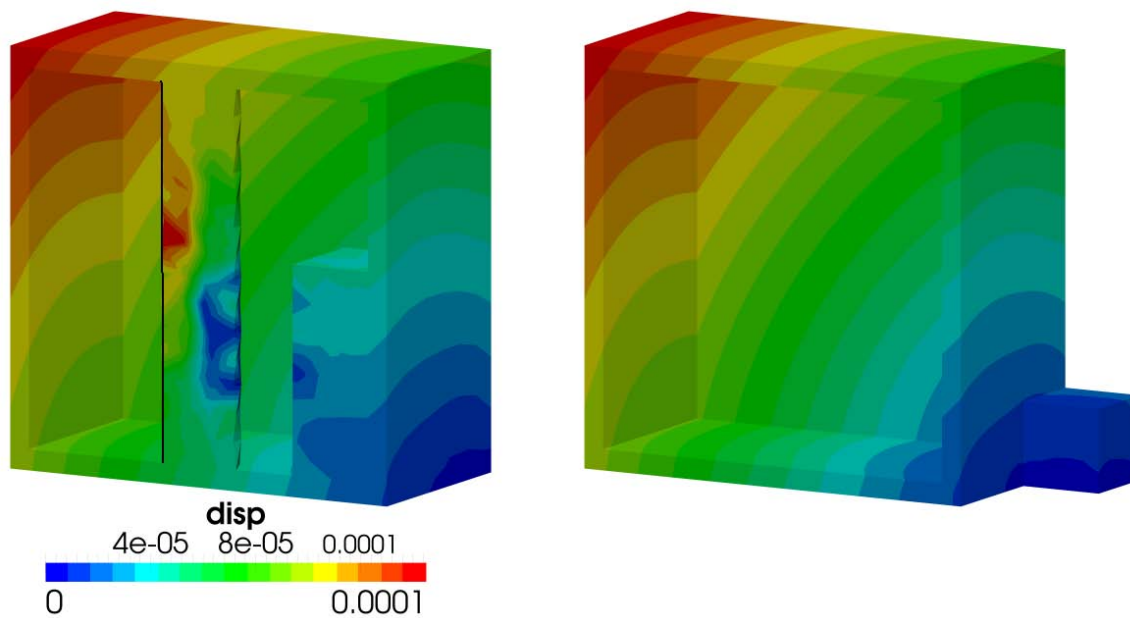


Figure 3.2. The transfer of displacement from the mechanical mesh (right) onto the thermal mesh (left).

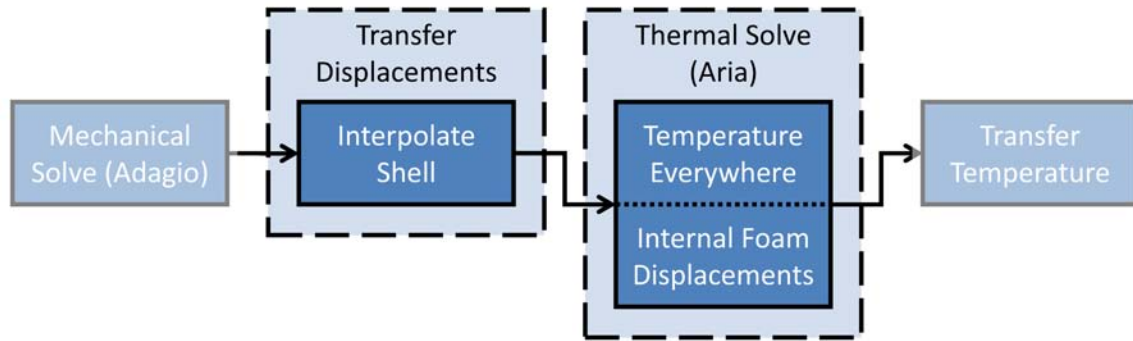


Figure 3.3. An alternate method for obtaining internal foam displacements in Aria.

Alternate method for displacements

As mentioned earlier, we are using Aria to calculate the thermal problem and Adagio to calculate the mechanical problem. In its current state, Aria also has basic capabilities in computing a mechanical response. We were able to utilize this feature to obtain a better solution for the displacement of the internal foam blocks. To do so, we modified the displacement transfer operator to skip the interior foam blocks, and set up Aria to solve for the displacements of these blocks based on an specified elastic material model. We do so by adding the lines

```

Begin Aria Material AWEThermal
  # set up thermal properties
  ...
  # set up mechanical properties
  Two Mu = Constant Two_Mu = 1.538e11 # Pa
  Lambda = Constant Lambda = 1.15e11 # Pa
  Mesh Stress = NeoHookean_Elastic
End
  
```

into the material definition for the internal foam blocks, and then activate solving for displacements by adding

```

# set up the aria solve region
Begin Aria Region aria_region
  # set up mesh groups
  Mesh Group adagio_blocks = block_1 block_4
  Mesh Group interior_blocks = block_2 block_3
  # transfer displacements for blocks which exist on both meshes
  EQ Mesh For mesh_displacements On adagio_blocks Using q1 With xfer
  # solve for the interior foam displacements
  EQ Mesh For mesh_displacements On interior_blocks Using q1 With diff
  ...
End
  
```

into the definition for the Aria region. Although the use of tet elements is not ideal in mechanical simulations, this process can eliminate the possibility of inverted elements in the interior foam blocks. Furthermore, this inclusion has no direct effect on the Adagio solution, since the blocks are not present in the mechanical model. This process is depicted in Figure 3.3.

The results from this method are shown in Figure 3.4. In comparison to Figure 3.2, one can see the displacement field in the internal foam blocks is much smoother, without the previously seen

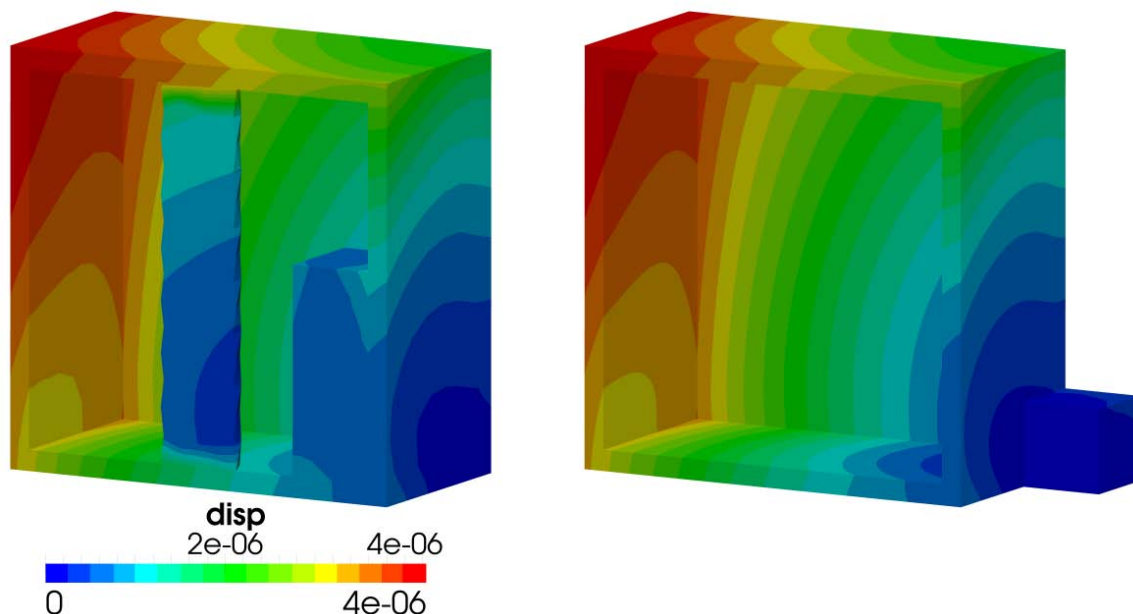


Figure 3.4. The resulting displacements when letting Aria solve for the displacement of the internal foam parts. Displacement is in meters.

local fluctuations. This behavior is closer to what we expect.

However, we did notice some odd behavior in this example. Especially in the center cylindrical foam block, the displacements seems to be lower than the surrounding material by more than a factor of 2. One can see this by noticing the large change in the displacement field where it transitions from the foam cylinder to the outer shell. This figure is shown at one of the early timesteps. At later timesteps, this large change gradually decreases and toward the end of the simulation it is no longer apparent.

It was discovered that this discrepancy resulted from the order we are solving the equations along with the step when we output the results. In this case, we were solving Aria first, then Adagio, then outputting results at the end of each step. During the Adagio solve, the displacements in the foam blocks are not updated. Since output occurred after Adagio solves, the displacements in the interior foam blocks was always one timestep behind the displacements in the shell.

To fix this issue, we changed the solution order within each timestep to Adagio first, then Aria. The effect is that Aria has the most updated masses and radiation enclosures, which is more important to thermal transport than the updated temperature is for computing the deformations. With this new ordering, as shown in Figure 3.5, the displacements calculated in Aria and Adagio match each other in the output file (i.e., the internal foam displacements do not lag by one timestep). The results from this switch can be seen in Figure 3.6. In this figure, the displacement field has a smooth transition from the internal foam blocks to the shell.

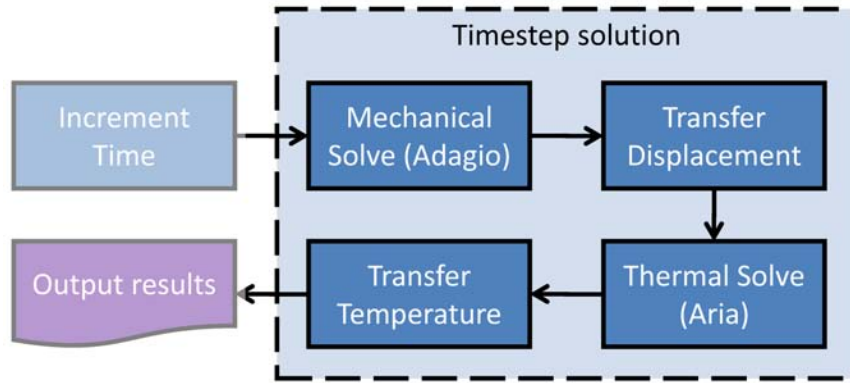


Figure 3.5. The revised timestep solution order to have displacements calculated in Aria and Adagio match in the output file.

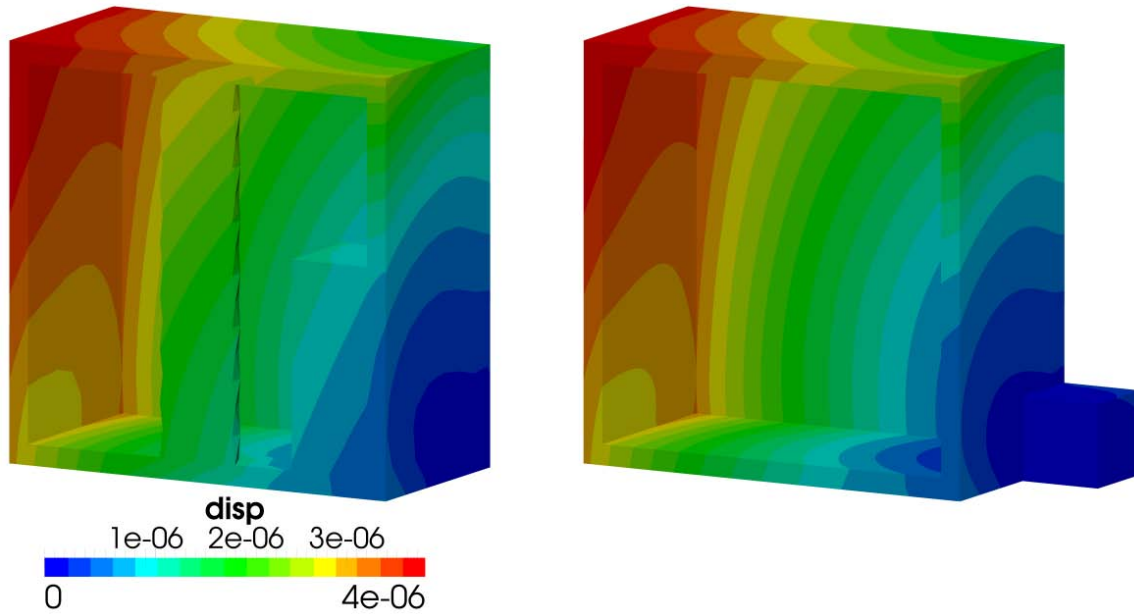


Figure 3.6. The resulting displacements from the alternate method when solving Adagio first, then Aria. Displacements are in meters.

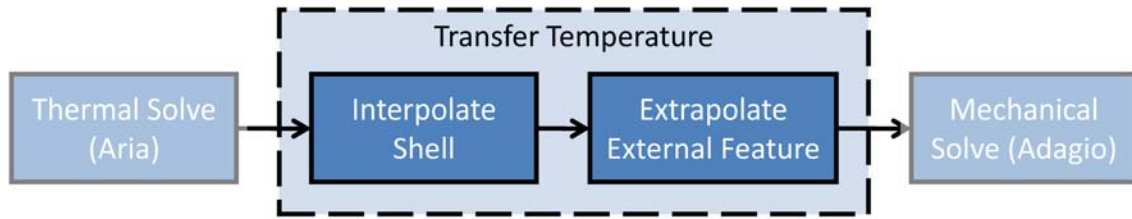


Figure 3.7. A schematic of the temperature transfer operation.

Temperature transfer

Once Aria solves the heat equation, the temperature field is transferred onto the mechanical mesh. This process is summarized in Figure 3.7. In the shell, which exists on both the mechanical and thermal meshes, the temperature is interpolated. It cannot be copied exactly since the two meshes use different element types. The temperature of the internal foam blocks is simply not used, as these blocks are not present in the mechanical mesh. Temperatures in the external feature are obtained by extrapolating temperatures from the shell. In the case, we have used the TRUNCATE operator for extrapolation.

The results of the temperature transfer can be seen in Figure 3.8. Here, the temperature is calculated on the thermal mesh (left) and transferred onto the mechanical mesh (right). On the enclosing shell, the temperature transfer performs very well. Despite there being differences in the two meshes, the resulting field looks identical.

However, the temperatures in the external feature do not appear reasonable. Although the temperature field in the shell is quite smooth, the extrapolated field is not smooth at all, with many local variations. These nonphysical variations in the temperature field are a problem as they produce nonphysical variations in the material parameters, as well as possibly significant amount of plastic strain. Although the simulation may still complete, having these artifacts present does not inspire confidence in the results. It should be noted that the default extrapolation operator, EXTRAPOLATE, was inappropriate because large temperature gradients at surfaces gave rise to nonphysically high temperatures in the extrapolated parts in the mechanical model.

Using the PROJECT operator

Upon communicating these results to the SIERRA developers, it was determined that the transfer operators were using parametric (i.e., master element) coordinates in order to perform the extrapolation. Since parametric coordinates and current coordinates have considerable differences, the closest point in one parametric space is not necessarily the closest point in current coordinates. The result of this was the odd behavior we were experiencing.

To correct this issue, a new extrapolation operator, PROJECT, was implemented into the code. This behaves the same as the old operator, TRUNCATE, except that it uses current coordinates instead of parametric ones. The results using this operator can be seen in Figure 3.9.

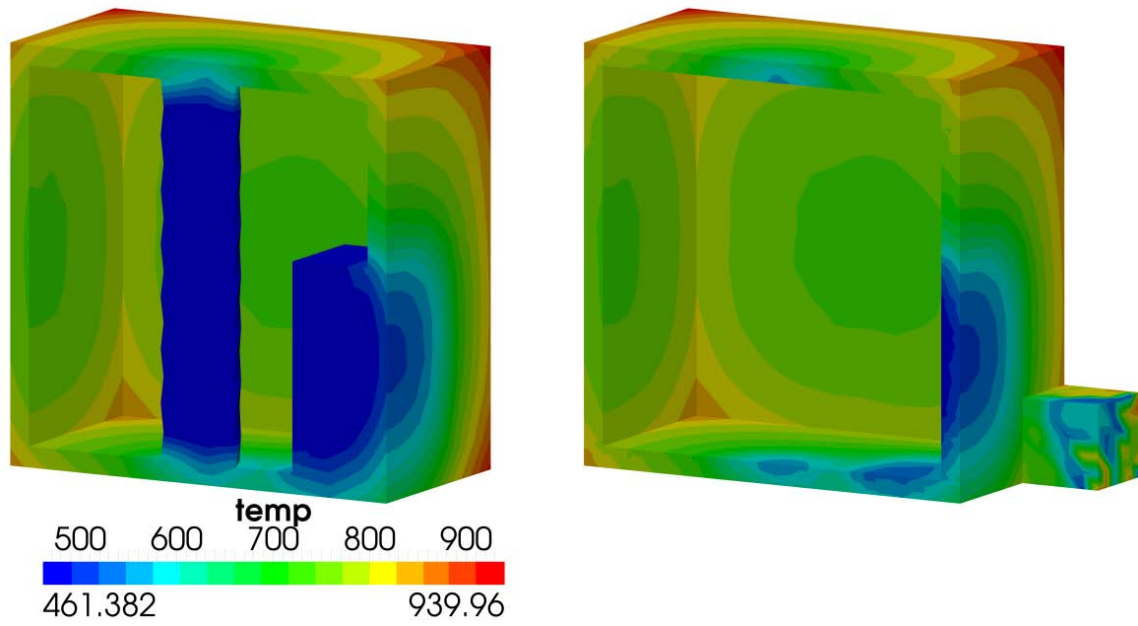


Figure 3.8. The transfer of temperature from the thermal mesh (left) onto the mechanical mesh (right) while using the TRUNCATE operator. Temperature is in Kelvin.

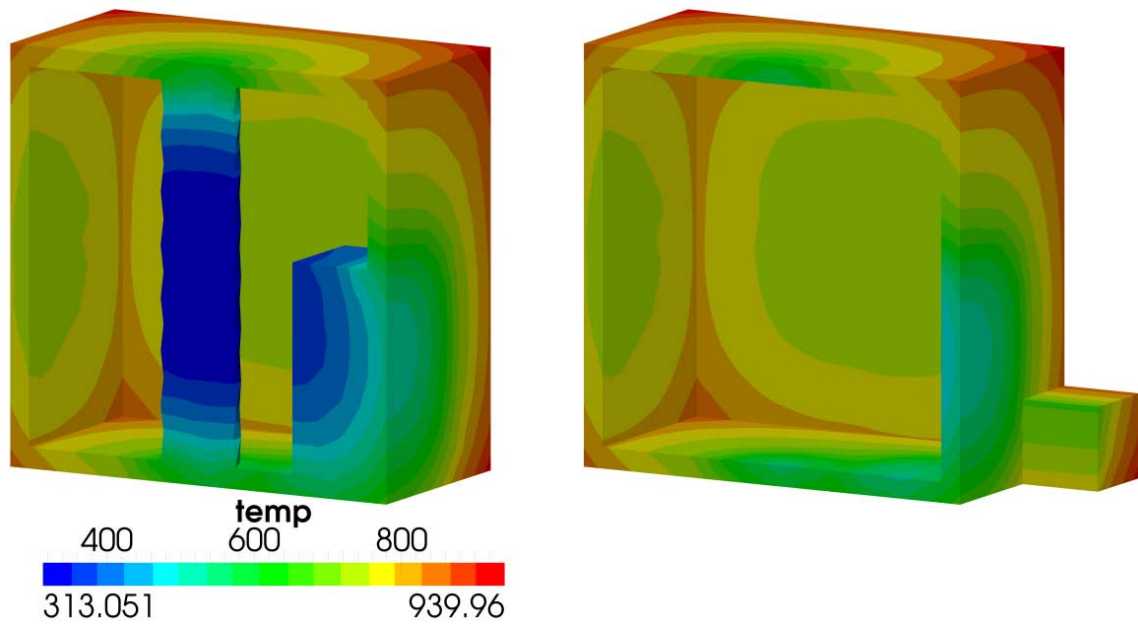


Figure 3.9. The transfer of temperature from the thermal mesh (left) onto the mechanical mesh (right) while using the PROJECT operator. Temperature is in Kelvin.

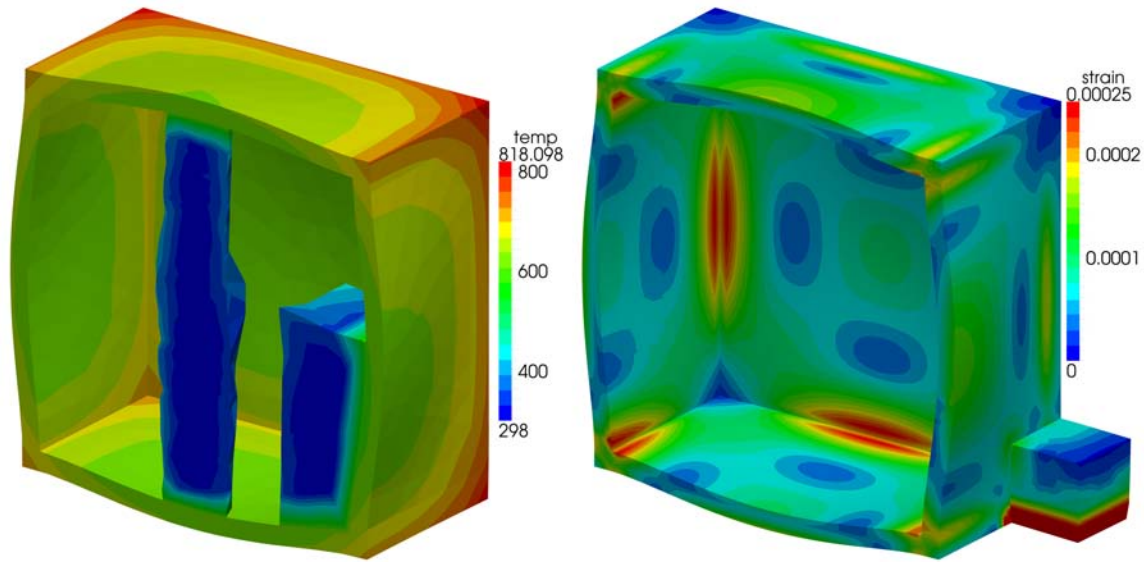


Figure 3.10. The internal pressure created by foam decomposition causing the walls to bow out. Temperature is in Kelvin.

Examining the temperature field on the mechanical mesh on the right, one can see the field is much smoother in the external block than that of the previous figure. The extrapolation operator correctly chooses the nearest point by projecting onto the face of the cube. Thus, the resulting field in the external block is simply an extrusion of the field (i.e. any cross-section of the field along the external feature looks identical).

Foam decomposition

In Figure 3.10, one can see the displacement caused by the internal pressure from the foam decomposition model. In this figure, displacements have been exaggerated 100x for clarity. On the left, one can see the temperature distribution in the thermal model. At this point, most of the internal foam is still near room temperature, but at the surfaces it is much hotter due to conduction and radiation. On the right, the effective strain field is plotted showing high values along the inside edges of the shell, and on the bottom of the external feature where it is being held in place.

On the left, one can also see some odd behavior in the displacements of the internal foam parts. In this simulation, we have used the PROJECT extrapolation operator. Since displacements are small at this stage, these artifacts have not caused element inversion, but at larger displacements this is likely to occur.

In this case, the ideal solution would be to use the alternate method for obtaining displacements in the internal foam parts. However, while combining this method along with the foam decomposition model, we encountered an error during the first Aria solve. As of the time of writing, this issue has not been resolved, but it has been reported to the SIERRA developers and a support ticket has been created.

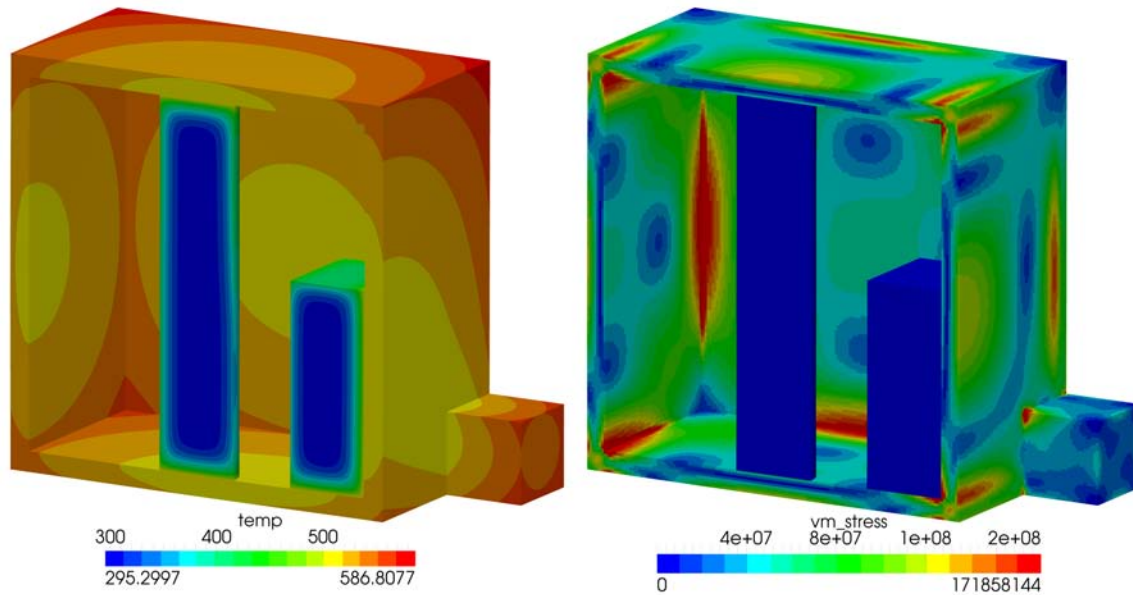


Figure 3.11. The temperature field (left) and effective stress field (right) obtained for the most refined mesh. Temperature is in Kelvin, and stress is in Pascals.

Encore quantitative comparisons

In this section, we will report quantitative comparisons using Encore between the various methods described in the previous sections. In order to generate an accurate “true” solution for the coupled problem without transfer operators, a combined model was built. The mesh was composed of about 1.5 million HEX8 elements, and the same mesh was used for both the thermal and mechanical analyses. There are 8 elements through the shell thickness, and all elements have the same edge lengths. For the most refined mesh, we have plotted the temperature and effective stress fields in Figure 3.11. Results are plotted and compared at one hour into the simulation.

The results using this mesh will be used as a standard to compare the various methods. For comparison, we are interested in several different metrics:

- **Relative L2 difference norm of the temperature over the thermal region (enclosing shell, door, internal foam)** - This will be used as a general measure of the fidelity of the thermal solution.
- **Relative L2 difference norm of the displacement over the mechanical region (enclosing shell, door, external feature)** - This will be used as a general measure of the fidelity of the mechanical solution.
- **Relative L2 difference norm of the temperature over the external feature** - This will be used to gauge the various methods of obtaining temperatures for parts which do not exist in the thermal model.

- **Relative L2 difference norm of the displacement over the internal foam** - This will be used to gauge the various methods of obtaining displacements for parts which do not exist in the mechanical model.

The relative L2 difference norm for the temperature is defined as

$$\psi_{\theta} \stackrel{\text{def}}{=} \frac{\int_{\Omega} \|\theta - \hat{\theta}\| d\Omega}{\int_{\Omega} \|\hat{\theta}\| d\Omega} \quad (3.1)$$

where $\hat{\theta}$ is the exact temperature field and θ is the comparison temperature field. The relative L2 difference norm for the displacement is defined as

$$\psi_{\mathbf{u}} \stackrel{\text{def}}{=} \frac{\int_{\Omega} \|\mathbf{u} - \hat{\mathbf{u}}\| d\Omega}{\int_{\Omega} \|\hat{\mathbf{u}}\| d\Omega} \quad (3.2)$$

where $\hat{\mathbf{u}}$ is the exact displacement field and \mathbf{u} is the comparison displacement field. Integration for both of these is performed over the reference configuration.

Note that we are choosing the displacement field as a metric for the mechanical solution, rather than the stress field. Since rigid body motions can affect the displacement without changing the stress, this measure is not ideal. (We have eliminated rigid body motions by prescribing exactly six nodal constraints.) However, practical considerations prevent us from comparing the stress fields between different meshes. In Encore, calculations are performed over a single mesh, so in order to do a comparison, one must transfer the field from the "comparison" mesh onto the "exact" mesh using some sort of transfer operator. Through this method, Encore assumes the field to be continuous to and defines correspondingly smooth interpolation operators. Since the stress field is discontinuous, the transfer operators act to smooth out the field, causing the transferred field to be considerably different from the original field and preventing any meaningful quantitative comparison between the two. These artifacts can be seen in Figure 3.12 for the strain field. Since the displacement field is continuous, we have chosen to use that instead of the strain or stress. While imperfect, if the displacement field is converging, then both the strain and stress fields are also converging so it is still a useful metric.

Convergence analysis

In order to verify our "exact" unified mesh is close enough to the converged solution, we compared results against a few coarser discretizations. The results from this can be seen in Table 3.1. In comparison to the "exact" mesh which has 8 elements through the thickness, the unified 2x, unified 3x, and unified 4x meshes have 2, 3, and 4 elements through the thickness, respectively. The timestep size has also been varied proportionally to element edge length. With linear elements, we expect the convergence rate to be linear. As shown in Figure 3.13, we see convergence rates for our metrics between 0.85 and 2.20. Unfortunately, due to computational cost, the range of mesh sizes studied was relatively small. For this reason, variations in the convergence rate are expected, but all metrics are convergent.

In Table 3.1, the values of the error norms of the finest refinements give us an indication as to how close we are to the true solution. As we shall see, the error norms for the various methods are

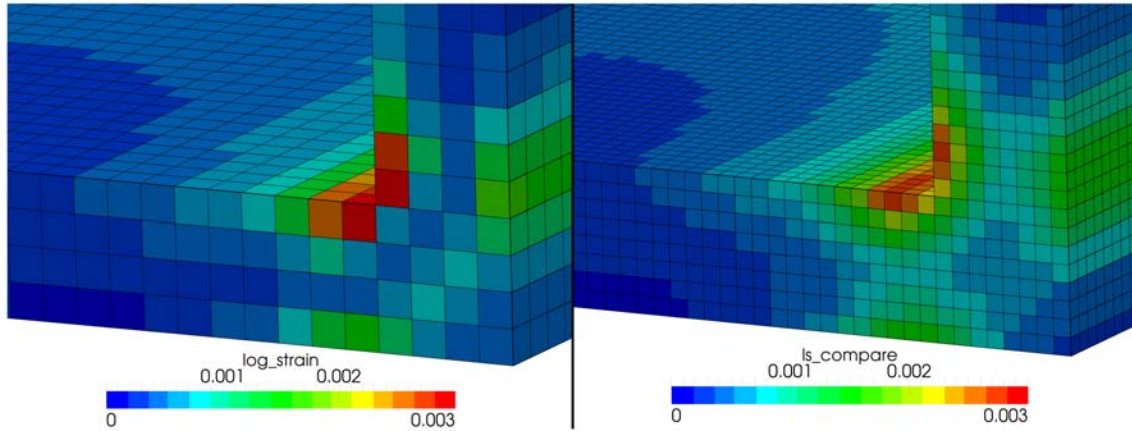


Figure 3.12. A transfer of the logarithmic strain field from a coarser mesh (left) onto a finer mesh (right). Note how the transfer smooths out the field which alters it significantly.

run	h	thermal	mechanical	feature	foam
unified 4x	0.25	0.000731	0.003804	0.000170	0.019260
unified 3x	0.33	0.001544	0.005831	0.000264	0.030841
unified 2x	0.50	0.003397	0.007024	0.000511	0.049332

Table 3.1. The element edge length relative L2 difference norms for a series of mesh refinements.

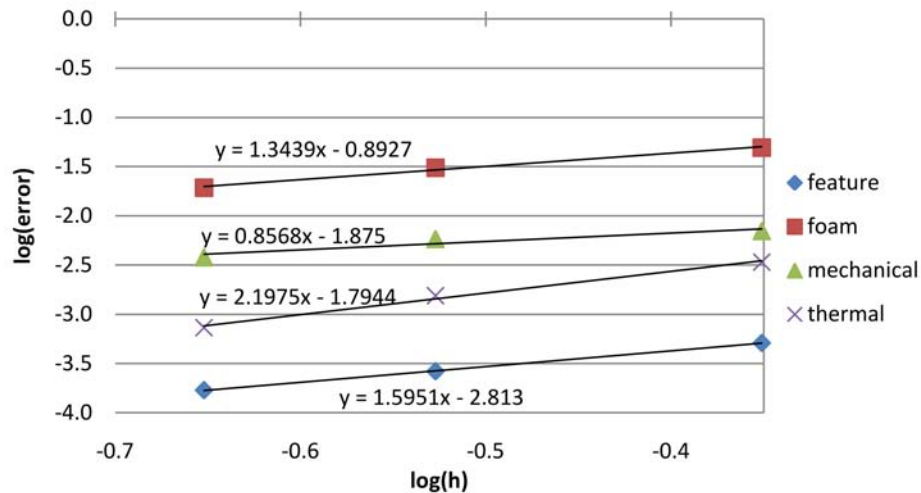


Figure 3.13. A convergence analysis of each metric on a series of mesh refinements.

all higher than the error norms in this chart, which gives us confidence that we have sufficiently resolved the mesh for the purposes of this work.

It should be noted that a **unified 16x** run was attempted, but ultimately did not run due to insufficient memory in calculating the internal enclosure radiation view factors. The sideset for this enclosure contained around 612,652 element faces, and the run was attempted using 2048 processors, with each equipped with 1.5GB of RAM. We felt a factor of 2 was necessary between the “exact” mesh and the comparison mesh to yield a reasonable error estimate, thus **unified 8x** was used as the finest. The **unified 2x** mesh was chosen as the most coarse refinement as we felt it necessary to have at least two elements through the thickness.

Methods comparisons

We will compare the following methods of dealing with mesh dissimilarities:

- **dissimilar** - This is the standard run which uses a hex mesh for the mechanical model (without the foam pieces), and a tet mesh for the thermal model (without the external feature). The temperature in the external feature is extrapolated using the `Nodes Outside Region = TRUNCATE` option in the `Transfer` block. The displacement in the internal foam is extrapolated in the same manner. All other methods are the same as this one except where noted.
- **project** - This method uses the recently implemented Framework Transfer option `Nodes Outside Region = PROJECT` to extrapolate temperatures and displacements.
- **tets** - This method uses a tet mesh for both the mechanical and the thermal solution. Since tet meshes are straightforward to generate, this method, if sufficiently accurate, could be used to obtain a solution.
- **foam tets** - This method uses a tet mesh for the internal foam parts, while using a hex mesh on the other pieces. The interface between parts is connected using a tied contact. This method would be appropriate for a complex model if the only difference between meshes is the absence of foam in the mechanical model.
- **encompassing** - On the mechanical mesh, foam blocks were constructed which encompass the thermal foam blocks. The reasoning behind this is that while the exact foam geometry might be hard to model, it is relatively easy to create a block which entirely surround the foam. By doing so, the displacement of the foam in the thermal model is interpolated, not extrapolated, which is the preferred method.
- **no transfer** - This is the same as the **unified 3x** run in the convergence study except that no transfer of displacement or temperature takes place. Thus, it is as if the two simulations are run separately. This will be used to see the effect of the coupling.
- **aria mechanical** - The displacements for the shell and door are transferred from Adagio, and Aria is used to calculate displacements for the interior foam.

The results of these runs were used and the relative L2 difference norms were calculated using the **unified 8x** results as the true solution. The results are presented in Table 3.2. Examining the **no transfer** results, one can see that transferring displacements has very little effect on the thermal

run	thermal	mechanical	feature	foam
dissimilar	0.010037	0.013670	0.040565	0.664070
project	0.010037	0.013651	0.040259	0.664074
tets	0.001190	0.030532	0.000158	0.052696
foam tets	0.010077	0.013690	0.040576	0.065381
encompassing	0.010075	0.013753	0.040574	0.084534
no transfer	0.002179	0.880793	N/A	N/A

Table 3.2. The relative L2 difference norms for the various methods of dealing with mesh dissimilarities.

solution. The relative L2 difference norm went from 0.15% to 0.22% in this case, suggesting that including displacement in the thermal problem changes the solution by less than 0.1%. For the mechanical response, the relative L2 difference norm went from 0.6% to 88%. This large increase is mostly due to thermal expansion affects. Obviously, if temperature is not transferred then parts do not expand. However, thermal expansion, if done slowly enough, does not significantly affect the stress field, but unfortunately a measure of this change was not able to be obtained for the reasons outlined earlier. Thus, the 88% change in the displacement field suggests things are worse than they may actually be. Changes in temperature also affect the material properties, which is potentially a significant factor in the results.

In looking the results for the **tets**, one can see the error for the thermal and feature norms is minimal (as expected, since tets perform well in thermal analyses). More surprisingly, the mechanical error is around 3%, and the foam error around 5%. Compare these to the errors of 0.5% and 3% for the **unified 3x** run. Although the errors are higher, they are of the same order of magnitude, which suggest that tet elements may perform sufficiently well for the mechanical analysis. There are other reasons why hex elements are preferred, including tet elements having too stiff of a response in many cases. Also, if modeling failure is an objective, more work has been done using hex elements, and thus material properties have been calibrated for those. Nevertheless, this demonstrates tet elements may be sufficient in certain cases.

It should not come as a surprise then that the **foam tets** run had a low error in the foam displacement norm of about 6.5% while maintaining the mechanical error of 1.4%. This shows using tet elements in the foam is a good solution. The **encompassing** run performed similarly, suggesting it may be sufficient to have some foam component in the foam model even if it does not conform to the exact dimensions of the foam in the thermal model. The error in the foam displacement for these two methods is shown in Figure 3.14. The resulting error fields are very similar, with similar hotspots and trends. The **encompassing** has slightly higher error magnitudes.

The foam displacement error obtained through the projection schemes (i.e. the **dissimilar** and **project** runs), was about 66%, significantly off from that of the true solution. Clearly, any attempt at modeling the internal foam is better than using an extrapolation technique. The tradeoff in computational time is worth the increased accuracy. Also, by modeling the foam, we nearly eliminate the possibilities of inverted elements in Aria crashing the simulation.

There was no clear winner in extrapolating temperatures to the external feature. The baseline **TRUNCATE** option obtained an error of around 4.06%, while the **PROJECT** option obtained an error of

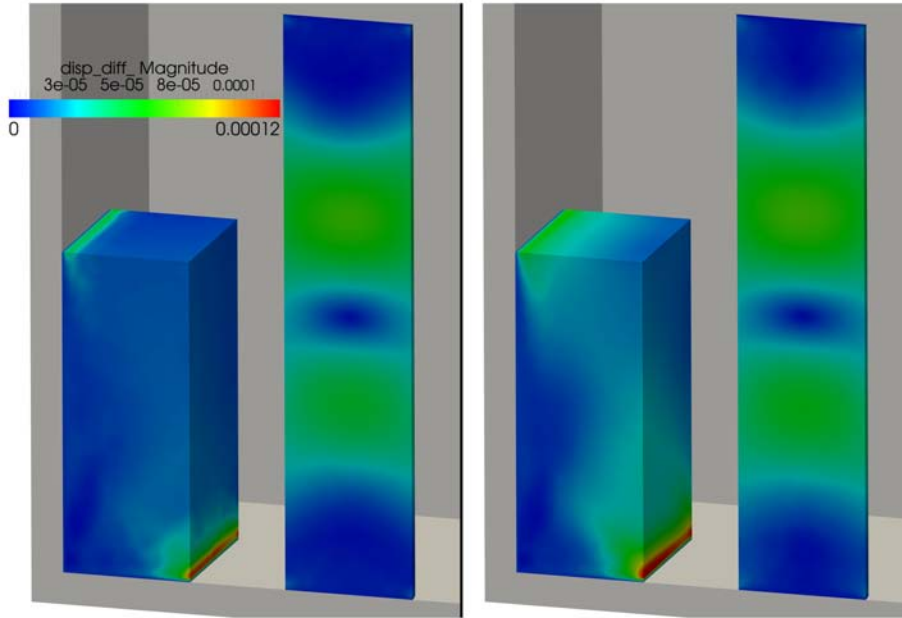


Figure 3.14. The error in foam displacement for the tets run (left) and the encompassing run (right). Displacement difference is in meters.

around 4.03%. The error in temperature for both of these runs is plotted in Figure 3.15. Although the errors are about the same, one can see the project option produces a smoother field which one would expect produce less non-physical residual stresses within the material. For this reason, the **PROJECT** option is preferred.

Computational cost of coupled simulations

A few words are needed on the computational cost of doing coupled analyses. In general, the heat diffusion equation is much easier to solve than the balance of linear momentum, and the computational cost of Aria and Adagio reflects this. While exact timings were not obtained, each Aria solution step took much less time than each Adagio solution step. If contact were included on the Adagio side, the disparity in computation cost can be up to several orders of magnitude. Because of this, one could take much smaller timesteps in Aria than in Adagio without significantly increasing the computational cost. However, if most of the error is brought about due to the mechanical solution, this change does not affect the overall error in the solution. In this analysis, for simplicity, we have chosen to use equal timesteps in Aria and Adagio. While the default timestep size of 2.4 seconds might have been relatively large, we have shown through the convergence analysis that this approaches the converged solution to within 1%.

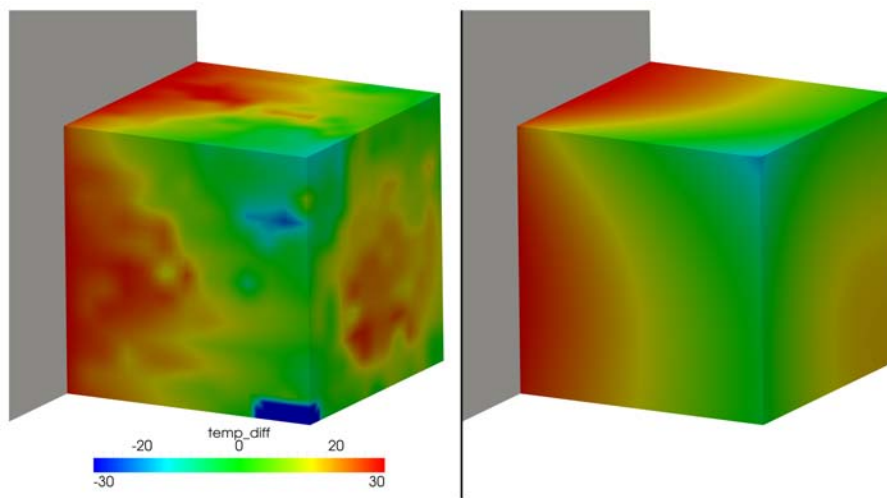


Figure 3.15. The error in external feature temperature for the dissimilar run (left) and the project run (right). Temperature difference is in Kelvin.

Chapter 4

Conclusions

This work examined several options for loose thermomechanical coupling within the SIERRA framework for models with different thermal and mechanical meshes and geometries. Given the integration of the various codes in SIERRA, including Aria and Adagio, data transfers between them already existed. However, the existing suite of techniques proved inadequate for coupled simulations because transferred fields contained nonphysical artifacts due to extrapolation over long ranges using coordinates not grounded in physical space. A partial remedy was obtained by the new PROJECT extrapolation operator requested in SIERRA as part of this work to perform physical space-based extrapolation. Specifically, this aided the mechanical solution in that small parts included in it but removed from the thermal model were able to be given appropriate thermal fields.

Extrapolation of displacements from the mechanical to thermal solution was more complex given the added geometric elements in the thermal model. Several approaches were tested and quantitatively compared. The best solutions involved including the thermal parts in the mechanical model in some fashion, either with tet-based meshes on the thermal-only components tied with the existing mechanical model or by creating simple encompassing shapes that enabled interpolation rather than extrapolation to be used. It was shown that, if possible, using a fully hex-based mesh for the mechanical model yielded the least amount of error. Finally, elasticity solutions using Aria on thermal-only parts were also used with great promise. As of this writing, this method is incompatible with the foam models being used but it is anticipated that on-going SIERRA development will remove this problem.

Appendix A

Sample Arpeggio input deck

In this appendix we present a sample input deck for an Arpeggio simulation using dissimilar geometries (i.e. a separate thermal and mechanical mesh file).

```
1 # input file for the simplified model
  Begin Sierra simplified_arpeggio_model
    # we are using base SI units (m, kg, s, K, A, cd)
    # except for amount which is (kmol)
5   # name the simulation
    Title Simplified Arpeggio Model
    # solver for a system of linear equations
  Begin Aztec Equation Solver aztec_solver
    Preconditioning Method = Jacobi
10   Residual Norm Scaling = Anorm
    Residual Norm Tolerance = 1e-10
    Solution Method = CG
  End
  # used as a full tangent preconditioner for Adagio's nonlinear CG solver
15  Begin Feti Equation Solver feti_solver
    Residual Norm Tolerance = 1e-10
  End
  # define global constants
  Begin Global Constants
20   Stefan Boltzmann Constant = 5.670400e-8 # kg / s^3 K^4
    Ideal Gas Constant = 8314.472 # J / kmol K
  End
  # pmdi foam material (without decomposition model)
  Begin Aria Material pmdi_foam
25   Density = Constant Rho = 319.19 # kg / m^3
    Emissivity = Constant E = 0.8
    Specific Heat = user_function x = temperature Name = pmdi_foam_specific_heat
    Heat Conduction = Basic
    Thermal Conductivity = user_function x = temperature Name = \
      pmdi_foam_thermal_conductivity
30  End
  # structural foam properties for use in Adagio
  Begin Property Specification For Material pmdi_foam
    Density = 319.19 # kg / m^3
    Thermal Strain Function = pmdi_foam_thermal_expansion
35  # estimates
  Begin Parameters For Model elastic_plastic
    Beta = 1.0
    Hardening Modulus = 0.07875e6 # Pa
    Poissons Ratio = 0.05 # estimate
40   Yield Stress = 0.08e6 # Pa
    Youngs Modulus = 0.1e9 # Pa
  End
  End
  # (linear) thermal expansion of foam (estimated)
45  Begin Definition For Function pmdi_foam_thermal_expansion
    Type = Piecewise Linear
    Y Scale = 10e-6 # K^-1 (this is the CTE)
  Begin Values
```

```

-1e4 -1e4
50 298 0
1e4 1e4
End
End
# thermal conductivity for the foam
55 Begin Definition For Function pmdi_foam_thermal_conductivity
Type = Piecewise Linear
Begin Values
303 0.0486 # W / m K
523 0.0706
60 End
End
# specific heat for the foam
Begin Definition For Function pmdi_foam_specific_heat
Type = Piecewise Linear
65 Begin Values
0 1269 # J / kg K
296 1269
323 1356
373 1497
70 423 1843
473 1900
523 2203
999 2203
End
End
75 # material description for 304L stainless steel
Begin Aria Material stainless_steel_304l
Density = Constant Rho = 7900 # kg / m^3
Emissivity = Constant E = 0.5
80 Specific Heat = user_function x = temperature Name = stainless_steel_304l_specific_heat
Heat Conduction = Basic
Thermal Conductivity = user_function x = temperature Name = \
stainless_steel_304l_thermal_conductivity
End
# define 304L stainless steel for use in Adagio
85 Begin Property Specification For Material stainless_steel_304l
Density = 7900 # kg / m^3
Thermal Strain Function = stainless_steel_304l_thermal_expansion
Begin Parameters For Model elastic_plastic
Beta = 1.0
90 Hardening Modulus = 787.5e6 # Pa
Poissons Ratio = 0.3
Yield Stress = 170e6 # Pa
Youngs Modulus = 200e9 # Pa
End
End
95 # specific heat of 304L stainless steel
Begin Definition For Function stainless_steel_304l_specific_heat
Type = Piecewise Linear
Begin Values
100 100.0 502.416 # J / kg K
273.15 502.416
673.15 565.218
3500.0 565.218
End
End
105 # thermal conductivity of 304L stainless steel
Begin Definition For Function stainless_steel_304l_thermal_conductivity
Type = Piecewise Linear
Begin Values
110 173.15 10.8857 # W / m K
273.15 13.3978
373.15 16.3285
773.15 21.7714
973.15 25.9582
115 3500 25.9582

```

```

    End
End
# (linear) thermal expansion of 304L Stainless Steel
Begin Definition For Function stainless_steel_304l_thermal_expansion
120   Type = Piecewise Linear
      Y Scale = 17.3e-6 # K^-1 (this is the CTE)
      Begin Values
        -1e4  -1e4
        0      0
125     1e4   1e4
      End
End
# temperature of the environment
# this is hot because it's in a fire
130   Begin Definition For Function environment_temperature
      Type = Piecewise Linear
      Begin Values
        0  1033.15 # K
      End
135   End
# internal pressure loading
Begin Definition For Function internal_pressure
140   Type = Piecewise Linear
      Y Scale = 6894.75729 # Pa / psi
      Begin Values
        0      0 # psi
        360    600
      End
End
145   # mechanical finite element model
Begin Finite Element Model fem_simple_mechanical
      # mesh file location and type
      Database Name = mesh_simple_mechanical.g
      Database Type = ExodusII
150   # shell, door, external feature
      Begin Parameters For Block block_1 block_2 block_4
        Material stainless_steel_304l
        Solid Mechanics Use Model elastic_plastic
      End
155   End
# thermal finite element model
Begin Finite Element Model fem_simple_thermal
      # mesh file location and type
      Database Name = mesh_simple_thermal.g
160   Database Type = ExodusII
      # shell, door
      Begin Parameters For Block block_1 block_2
        Material stainless_steel_304l
      End
165   # foam
      Begin Parameters For Block block_3
        Material pmdi_foam
      End
End
170   # this defines how to do the coupling
Begin Procedure coupling_procedure
      Begin Solution Control Description
        # send the initialized variables around
        # important to get the initial temperature correct
175     Begin Initialize init
          Advance aria_region
          Transfer aria0_to_adagio
          Advance adagio_region
          Transfer adagio_to_aria
180     End
      Use System Main
      Begin System Main
        Simulation Start Time = 0

```

```

Simulation Termination Time = 360 # s
185 Use Initialize init
# set up the timestep procedure
Begin Transient time_period
  Advance aria_region
  Transfer aria_to_adagio
190 Advance adagio_region
  Transfer adagio_to_aria
End
End
Begin Parameters For Transient time_period
195 Start Time = 0
Termination Time = 360 # s
Begin Parameters For Aria Region aria_region
  Initial Time Step Size = 9 # s
  Minimum Time Step Size = 9 # s
200 Time Step Variation = Fixed
End
Begin Parameters For Adagio Region adagio_region
  Time Increment = 9 # s
End
205 End
End
# initialization transfer
Begin Transfer aria0_to_adagio
  Interpolate Volume Nodes From aria_region To adagio_region
210 Send Field solution->temperature State New To temperature State Old
  Nodes Outside Region = truncate
End
# transfer temperature from Aria to Adagio
Begin Transfer aria_to_adagio
215 Interpolate Volume Nodes From aria_region To adagio_region
  Send Field solution->temperature State New To temperature State New
  Nodes Outside Region = truncate
End
# transfer displacements from Adagio to Aria
220 Begin Transfer adagio_to_aria
  Interpolate Volume Nodes From adagio_region To aria_region
  Send Field displacement State New To solution->mesh_displacements State New
  Nodes Outside Region = truncate
End
225 # thermal region
Begin Aria Region aria_region
  # define the model
  Use Finite Element Model fem_simple_thermal
  # define the linear solver
230 Use Linear Solver aztec_solver
  # solve the energy equation everywhere
  EQ Energy For temperature On all_blocks Using Q1 With mass diff
  # "solve" for displacements with transfer
  EQ Mesh For mesh_displacements On all_blocks Using Q1 With xfer
235 # nonlinear solver parameters
  Nonlinear Solution Strategy = Newton
  Nonlinear Residual Tolerance = 1e-09
  Nonlinear Correction Tolerance = 0 # prevent this from causing termination
  Maximum Nonlinear Iterations = 100 # default is 20 and sometimes not enough
240 # Set initial temperature to 298 K on all blocks
Begin Initial Condition set_initial_temperature
  All Volumes
  Temperature = 294.2611111 # K
End
245 # radiation to the outside (this ignores self-radiation)
Begin Radiative Flux Boundary Condition outside_to_atmosphere
  Add Surface surface_2
  Radiation Form Factor = 1
  Reference Temperature Time Function = environment_temperature
250 End
# viewfactor calculation parameters

```

```

Begin Viewfactor Calculation viewfactor_calculation
  Compute Rule Is hemicube
End
255 # viewfactor smoothing (required to conserve energy)
Begin Viewfactor Smoothing viewfactor_smoothing
  Method = least-squares
End
# radiosity solver parameters
260 Begin Radiosity Solver radiosity_solver
End
# internal radiation enclosure
Begin Enclosure Definition interior_enclosure
  Add Surface surface_3
265 Blocking Surfaces
  Use Viewfactor Calculation viewfactor_calculation
  Use Viewfactor Smoothing viewfactor_smoothing
  Use Radiosity Solver radiosity_solver
  Database Name Is interior_enclosure.e In Binary Format
270 Rowsum Database Name Is interior_rowsum.e
End
# calculate heat flux
Postprocess heat_flux On all_blocks
# define the output
275 Begin Results Output aria_output
  Database Name Is output_simple_thermal.e
  Database Type Is ExodusII
  Global Variables = current_time As time
  Nodal Variables = solution->temperature As temp
280 Nodal Variables = solution->mesh_displacements As disp
  Nodal Variables = pp->heat_flux As flux
  At Time 0 Interval = 36
End
End Aria Region aria_region
285 # mechanical region
Begin Adagio Region adagio_region
  # define the model to use
  Use Finite Element Model fem_simple_mechanical
  # nonlinear solver parameters
290 Begin Solver
  # how to predict the next solution step
  Begin Loadstep Predictor
    Type = Secant
  End
295 # define the nonlinear CG method
  Begin CG
    # set up convergence criteria
    Reference = Internal
    Target Relative Residual = 1e-09
300 Maximum Iterations = 1000
    Iteration Reset = 500
    Line Search Secant
    # use full tangent preconditioner
  Begin Full Tangent Preconditioner
305 Linear Solver = feti_solver
    # do X nodal preconditioner steps before using feti
    Number of Smoothing Iterations = 100
    # stop after X tries (10 is usually enough)
    Maximum Iterations for ModelProblem = 20
310 # update every X mp iterations (instead of only once at the beginning)
    Iteration Update = 21
    # try not to trigger stagnation condition
    Minimum Convergence Rate = 1e-16
  End
315 End CG
End Solver
# get rid of rigid body motions
Begin Fixed Displacement
  Node Set = nodelist_1 # 2 nodes in here

```

```

320     Component = XZ
      End Fixed Displacement
      Begin Fixed Displacement
        Node Set = nodelist_2 # 1 node in here
        Component = XY
325     End Fixed Displacement
        # apply pressure to interior surface (ignoring foam)
        # foam is initially closed cell, but becomes open cell
        # at elevated temperatures
      Begin Pressure internal_pressure
330     Surface = surface_4
        Function = internal_pressure
      End
        # get resultant force from the pressure condition
        # note that since surface_4 is a closed surface, the resultant
335     # force should be exactly zero
      Begin User Output resultant_internal
        Include All Blocks
        Compute Global resultant_internal As Sum Of Nodal force_internal
      End
340     Begin User Output resultant_external
        Include All Blocks
        Compute Global resultant_external As Sum Of Nodal force_external
      End
        # define the output
345     Begin Results Output adagio_output
        Database Name = output_simple_mechanical.e
        At Time 0 Interval = 36
        Node Variables = reaction
        Node Variables = displacement As disp
350     Node Variables = temperature As temp
        Node Variables = force_external As f_ext
        Node Variables = force_internal As f_int
        Node Variables = residual As resid
        Global Variables = resultant_internal
355     Global Variables = resultant_external
        Element Variables = effective_log_strain As log_strain
        Element Variables = von_mises As vm_stress
        Element Variables = eqps As eqps
      End
360     End Adagio Region adagio_region
      End Procedure coupling_procedure
End

```


Appendix B

Sample Encore input deck

In this section, we provide the basic Encore deck which was used to determine the error in the displacements. In order to do a quantitative comparison between different meshes, the comparison field must be transferred onto the mesh of the exact solution. Then, one can perform the calculation on the exact mesh. It is not possible to directly compare fields on different meshes. A sample file for calculating the relative L2 difference norm of the “disp” field between the output files `compare_mechanical.e` and `exact_mechanical.e` is shown below.

```
1 # file to compare solutions for the simple model
  Begin Sierra Encore
    # name the output file
    Begin Postprocessor Output Control postprocessor_control
5     Write To File encore_mechanical.txt
      # print as much precision as we can expect
      Floating Point Format Is Scientific
      Floating Point Precision Is 16
    End
10    # "exact" mechanical mesh
    Begin Finite Element Model fem_exact_mechanical
      Database Name = exact_mechanical.e
      Database Type = ExodusII
      Omit Block block_3
15    End
    # comparison mechanical mesh
    Begin Finite Element Model fem_compare_mechanical
      Database Name = compare_mechanical.e
      Database Type = ExodusII
20    Omit Block block_3
    End
    # create a field to hold stored displacement
    Begin Field Function function_transferred_nodal_compare_displacement
      Use Nodal Field transferred_nodal_compare_displacement
25    End
    Begin Field Function function_nodal_exact_displacement
      Use Nodal Field nodal_exact_displacement
    End
    # define a function for the difference
30    Begin Difference Function function_displacement_difference
      Difference Is function_transferred_nodal_compare_displacement - \
        function_nodal_exact_displacement
    End
    # define the post processing procedure
    Begin Encore Procedure encore_procedure
35    Begin Solution Control Description
      Use System main
      Begin System main
        Begin Transient encore_transient
          Advance compare_region
40          Transfer displacement_transfer
          Advance exact_region
        End
      Simulation Start Time = 0
```

```

Simulation Termination Time = 1e21 # Arbitrarily large
45 Simulation Max Global Iterations = 1000 # Arbitrarily large
End
End
# transfer displacement from the compare mesh to the exact mesh
Begin Transfer displacement_transfer
50 Interpolate Volume Nodes From compare_region To exact_region
Send Field nodal_compare_displacement State New To \
transferred_nodal_compare_displacement State New
End
# describe the compare region
Begin Encore Region compare_region
55 # load up the finite element model
Use Finite Element Model fem_compare_mechanical
# load the displacement field for use
Import Field disp as Nodal Field nodal_compare_displacement
End
# describe the exact region
60 Begin Encore Region exact_region
# load up the finite element model
Use Finite Element Model fem_exact_mechanical
# load the displacement field
65 Import Field disp As Nodal Field nodal_exact_displacement
# create a field to hold the transferred displacement
Create Nodal Field transferred_nodal_compare_displacement Of Type Vector_3D And \
Dimension 1
# compute relative L2 difference norm using function_nodal_exact_displacement as the \
exact
Compute Difference Relative L2 Of function_nodal_exact_displacement \
function_transferred_nodal_compare_displacement
70 # put the difference into a nodal field for visualization
Interpolate Function Value Of function_displacement_difference Into Nodal Field \
nodal_displacement_difference
# define the results output
Begin Results Output Label encore_results
# define the database file name and type
75 Database Name = output_encore_mechanical.e
Database Type = ExodusII
# output every step
At Step 0 Increment = 1
# which nodal variables to output
80 Nodal Variables = nodal_exact_displacement As disp_exact
Nodal Variables = transferred_nodal_compare_displacement As disp_compare
Nodal Variables = nodal_displacement_difference As disp_diff
End
End
85 End
End

```

References

- [1] Kevin D. Copps and Brian R. Carnes. Encore user guide. Technical report, Sandia National Laboratories, 2009. Document in progress.
- [2] H.C. Edwards. SIERRA Framework for massively parallel adaptive multiphysics applications. Technical Report SAND2004-6277C, Sandia National Laboratories, Albuquerque, NM 87185, 2005.
- [3] Ken Erickson and Amanda Dodd. Foam decomposition and container pressurization: Comparison of experimental and modeling results. Presented at the VNIIA-SNL Technical Exchange, August 2010.
- [4] Patrick K. Notz, Samuel R. Subia, Matthew M. Hopkins, Harry K. Moffat, and David R. Noble. Aria 1.5: User manual. Technical Report SAND2007-2734, Sandia National Laboratories, Albuquerque, NM 87185, April 2007.
- [5] V. L. Porter. Guide to coupled electrostatic-structural analysis with arpeggio. Technical Report SAND2006-5654, Sandia National Laboratories, Albuquerque, NM 87185, 2006.
- [6] J. R. Stewart. SIERRA Framework version 3: Transfer services design and use. Technical Report SAND2003-0249, Sandia National Laboratories, Albuquerque, NM 87185, 2003.
- [7] Cubit Development Team. Cubit 12.2 on-line user's manual (september 2010), September 2010. <http://cubit.sandia.gov/help-version12.2/cubithelp.htm>.
- [8] SIERRA Solid Mechanics Team. Adagio 4.16 user's guide. Technical Report SAND2010-3111, Sandia National Laboratories, Albuquerque, NM 87185, May 2010.

DISTRIBUTION:

1 MS 0346 Dean Dobranich, 1514 (electronic)
1 MS 0386 Amanda Dodd, 1514 (electronic)
1 MS 0836 Nicholas Francis, 1514 (electronic)
1 MS 0836 Roy Hogan, 1514 (electronic)
1 MS 0828 Martin Pilch, 1514 (electronic)
1 MS 0836 Tre' Shelton, 1514 (electronic)
1 MS 0836 Stefan Domino, 1541 (electronic)
1 MS 0836 Toluope Okusanya, 1541 (electronic)
1 MS 0836 Samuel Subia, 1541 (electronic)
1 MS 0828 Kenneth Alvin, 1544 (electronic)
1 MS 0828 Amalia Black, 1544 (electronic)
1 MS 0828 Brian Carnes, 1544 (electronic)
1 MS 0828 Kevin Dowding, 1544 (electronic)
1 MS 0828 Vicente Romero, 1544 (electronic)
1 MS 9154 Mary Gonzales, 8240 (electronic)
1 MS 9042 Michael Chiesa, 8249 (electronic)
1 MS 9042 Jay Dike, 8249 (electronic)
1 MS 9042 Bruce Kistler, 8249 (electronic)
1 MS 9042 Timothy Kostka, 8249 (electronic)
1 MS 9042 Alex Lindblad, 8249 (electronic)
1 MS 9042 Arthur Ortega, 8249 (electronic)
1 MS 9409 Neal Fornaciari, 8365 (electronic)
1 MS 9409 Jeremy Templeton, 8365 (electronic)
1 MS 9409 Gregory Wagner, 8365 (electronic)
1 MS 0823 Robert Baca, 9326 (electronic)
1 MS 0899 Technical Library, 8944 (electronic)

