SAND2017-0474C

Verification Examples from ASC Integrated Codes at Sandia

Brian Carnes and Walt Witkowski

Sandia National Laboratories
Validation, Verification, Uncertainty Quantification and Credibility Processes Dept.
bcarnes@sandia.gov

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 Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Credibility Assessment

Sandia National Laboratories

- The Predictive Capability Maturity Model (PCMM) is a communication tool for informing stakeholders of the level of maturity of an application-specific simulation capability
  - It is a multidimensional, qualitative metric
    - Determine readiness for stockpile issues
    - Identify gaps in credibility of application
    - Measure progress of integrated simulation effort

PCMM Elements

1. RGF: Representation or Geometric Fidelity
   Are representation errors corrupting simulation results?
2. PMMF: Physics and Material Model fidelity
   How accurate are the physics and material models?
3. CVER: Code Verification (inc. SQE)
   Are software errors or algorithm deficiencies corrupting simulations?
4. SVER: Solution Verification
   Are human procedural errors or numerical solution errors corrupting simulation results?
5. VAL: Model Validation
   How accurate are the integrated physics and material models?
6. UQ: Uncertainty Quantification/Sensitivity Analysis
   What is the impact of variabilities and uncertainties on system performance and margins?

PCMM allows to qualitatively measure our CompSim "due diligence"

PCMM is intended to be a communication and a planning tool

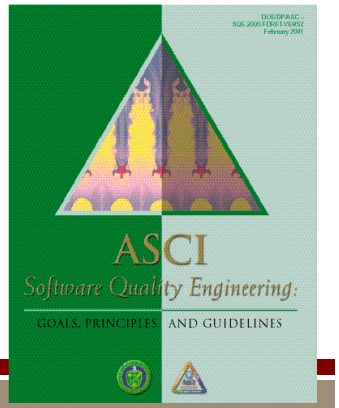
It is not intended to be a report card

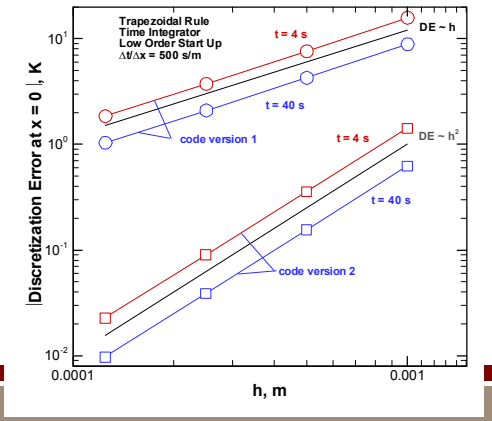
Verification Activities

"Are we solving the equations correctly?"

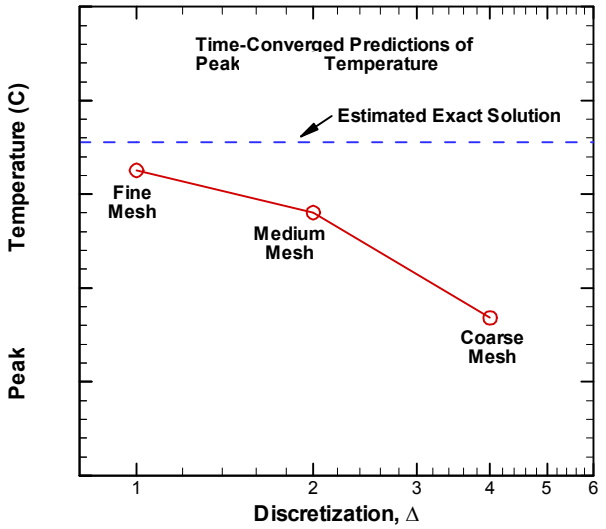
- Code verification: Correctness of implemented mathematical algorithms.
- Solution verification: Convergence to the correct answer, at the correct rate, as model is refined.

Code: Software quality practices & accuracy checks on test problems





Solution: Convergence checks on engineering application



Code and Solution Verification

Sandia National Laboratories

Code Verification is the activity of ensuring that the code correctly implements the numerical model.

- Errors in computer models are called code defects or bugs
- The code developers/testers have primary responsibility for identifying and eliminating code bugs
- Analysts should check that the features used are appropriate and adequately tested (for the intended application)

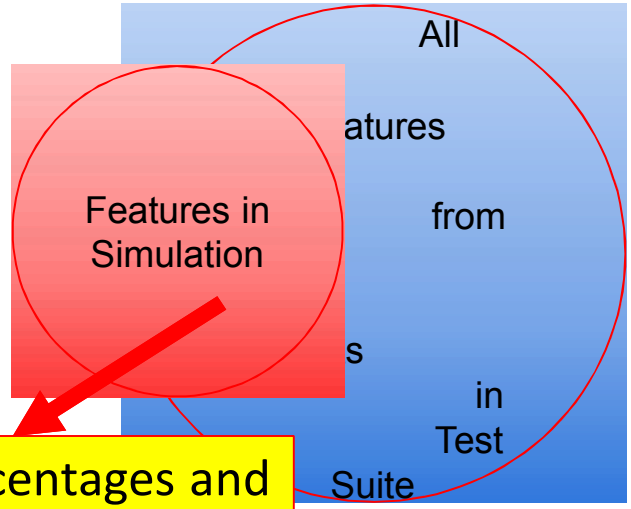
Solution Verification is the quantification and reduction of numerical error.

- Done in the context of the overall uncertainty budget.
- Error may or may not need to be reduced.
- Primarily the responsibility of analysts

Feature Coverage Tool (FCT)

Sandia National Laboratories

Intersects the set of features used in a simulation with the set from the full suite of nightly and verification tests.



Analyst's simulation (input file)

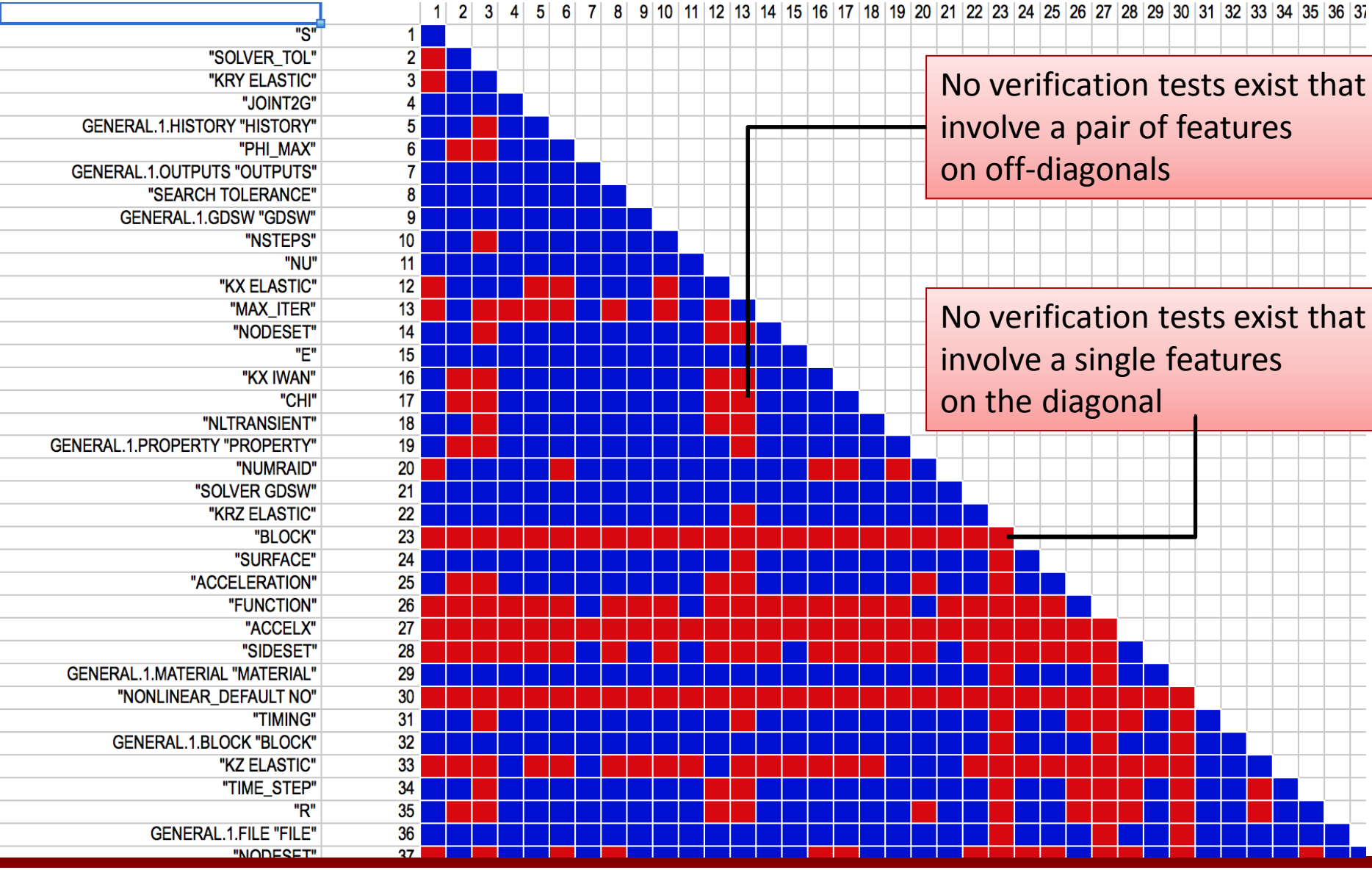
Annotated by the FCT

```
begin sierra myJob 1214 +
  # Define gmres linear system solver, this should not need to be changed.
  begin trilinos equation solver gmres 113 +
    param-int AZ_kspace value 1000 4 +
    param-real AZ_ilut_fill value 2 55 + # changed from 5.0
    maximum iterations = 1000 55 +
    solution method = gmres 29 +
    preconditioning method = dd-ilut 7 +
    #matrix scaling = row-sum # added
    preconditioner subdomain overlap = 2 67 +
    residual norm tolerance = 1.0e-9 56 +

  end   begin trilinos equation solver direct_solver 113 +
                solution method = amesos-umfpack 10 +

        end  # Define the mesh and which mesh blocks correspond to which elements.
  Begin Finite Element Model rect 1214 +
    database Name = single_stack_3D.g 1213 +
    coordinate system is cartesian 79 +
    decomposition method = rcb 173 +
    omit block block_4 8 +
    use material cathode_collector for surface_2 33 +
    use material cathode   for block_3 33 +
    use material separator for block_2 33 +
    use material anode     for block_1 33 +

  End Finite Element Model rect  Begin Output Scheduler forResults 18 +
    At step 0, Increment = 1 16 +
```

Report percentages and annotate the simulation input on intersection

```
verified

  * one-way: 58%
  * two-way: 27%

tested

  * one-way: 71%

untested

  * ignored
```

FCT Pairwise Feature Coverage



What is a Verification Test?

Sandia National Laboratories

A high quality test of the code, evaluating the accuracy and precision of the approximation, preferably on a problem with a known analytic solution.

Aiming for a higher level of rigor as opposed to:

- Code-to-code comparisons
- Benchmarks
- Comparing to the solution of a nearby simpler model (beam/shell theory)

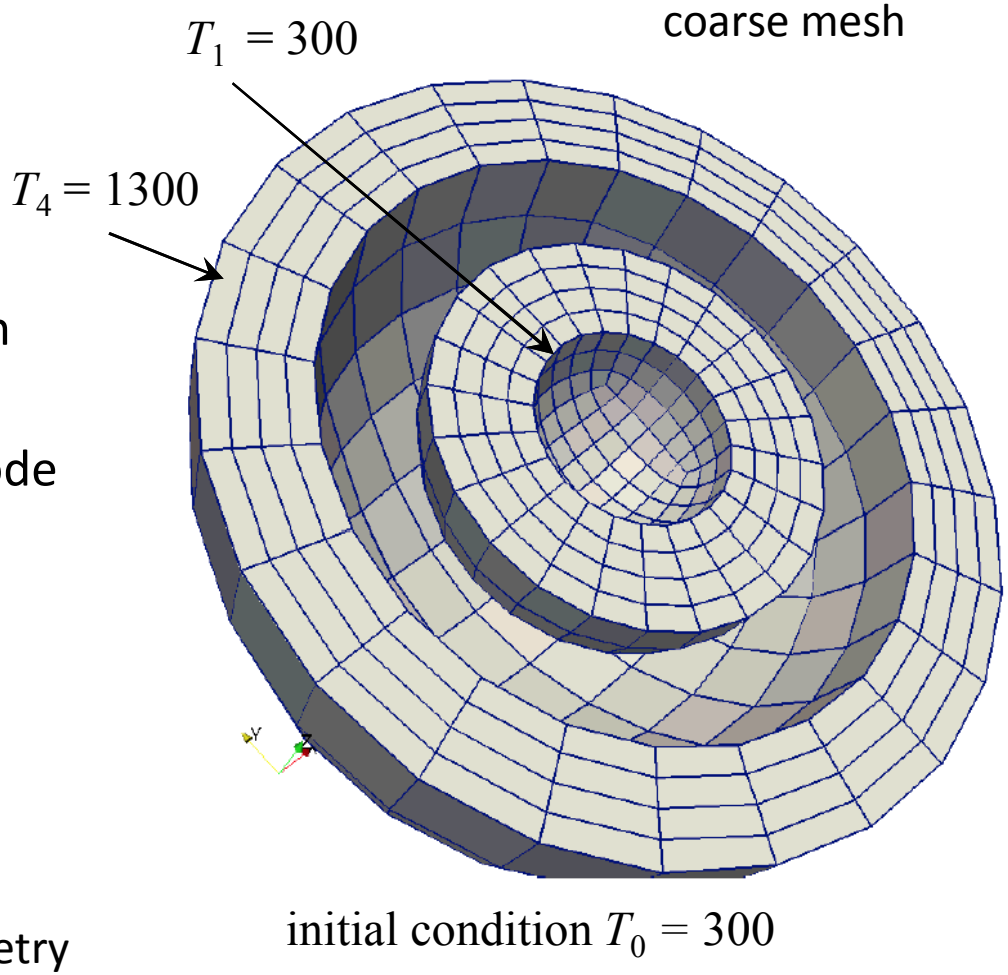
Procedure:

1. Construct an analytic solution to a problem
   a) Find exact solution on a simple domain, or
   b) Prescribe the solution and reverse engineer boundary conditions, source terms
2. Compute errors in suitable metrics, the difference between they analytic and approximate solution produced by the code
3. Examine convergence behavior as mesh size becomes uniformly smaller

Verification Test Example (Thermal)

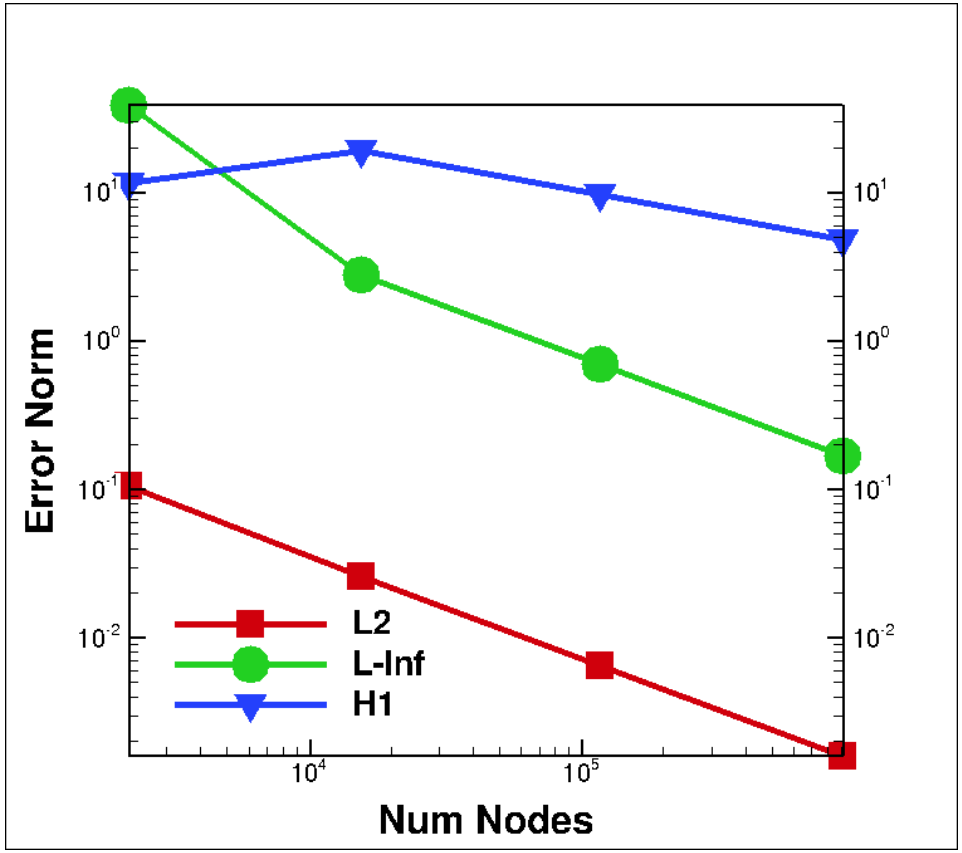
Goal: test whether code achieves expected order of convergence

- Transient heat conduction and enclosure radiation
- Formulated in such a way that an analytic solution can be derived
- Exact solution provided to the code
- Strengths:
  - Realistic range of temperature, gradients
  - Realistic material properties, time/length scales
- Weaknesses:
  - Problem possesses radial symmetry – cannot assess fully 3D behavior
  - Current version only looks at norms, not relevant QoIs



Passing the Test Requires Rigor

- The problem is run by the code on a sequence of four meshes using uniform refinement.
- Each subsequence mesh has 8X the elements as the previous.
- From theory, the expected rates of convergence in the L^2, L^∞, and H^1 norms are 2, 2, and 1, respectively.
- These are tested against the actual slopes using a log-log plot



Code Verification Documentation

- Sierra delivers manuals to document most of the verification test suite
- Specific to each code
  - SM, SD, Thermal, Aero, Fuego
- These are approved for unlimited release (UUR)

7.2.5 Verification of Solution

The manufactured solution is

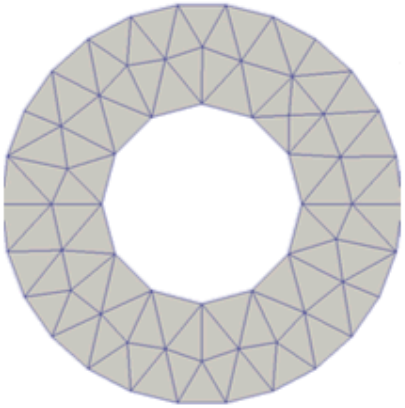
$$J(\theta) = k_1 + k_2\sqrt{\varepsilon}\cos\left(\frac{\sqrt{\varepsilon}\theta}{2}\right) / \sin\left(\sqrt{\frac{\varepsilon}{\pi}}2\right),$$

$$H(\theta) = k_1 + k_2(\frac{\sqrt{\varepsilon}}{1-\varepsilon})\left(\cos\left(\frac{\sqrt{\varepsilon}\theta}{2}\right) / \sin\left(\frac{\sqrt{\varepsilon}\pi}{2}\right) - \sqrt{\varepsilon}\cos\left(\frac{\theta}{2}\right)\right),$$

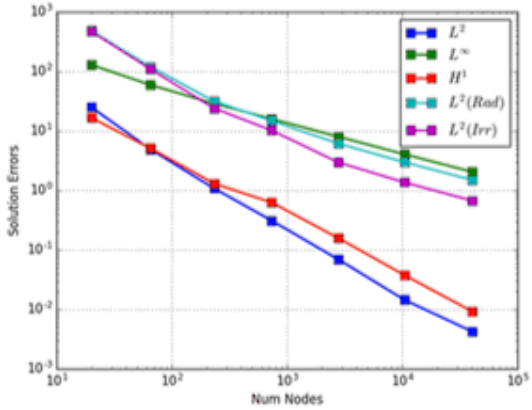
$$q(\theta) = J(\theta) - H(\theta),$$

$$\beta(\theta) = \left(\frac{k_1 + k_2\cos\left(\frac{\theta}{2}\right)}{\sigma}\right)^{1/4},$$

$$T(r,\theta) = r\beta(\theta) + (r - r_{\text{cyl}})\left(\frac{q(\theta)}{\kappa} - \beta(\theta)\right),$$



Coarse Mesh



Error Norms

Figure 7.2. 2D Full Enclosure Radiation

Single Code Verification

Sandia National Laboratories

- Aero – normal environment (free fall/captive carry/reentry)
- Solid mechanics – normal/abnormal environment (impact/crash/drop)
- Thermal – normal/abnormal environment (environmental/fire)
- Structural dynamics – normal/abnormal (vibration/blast)

Automatic Differentiation (AD) for MMS Source Terms

Sandia National Laboratories

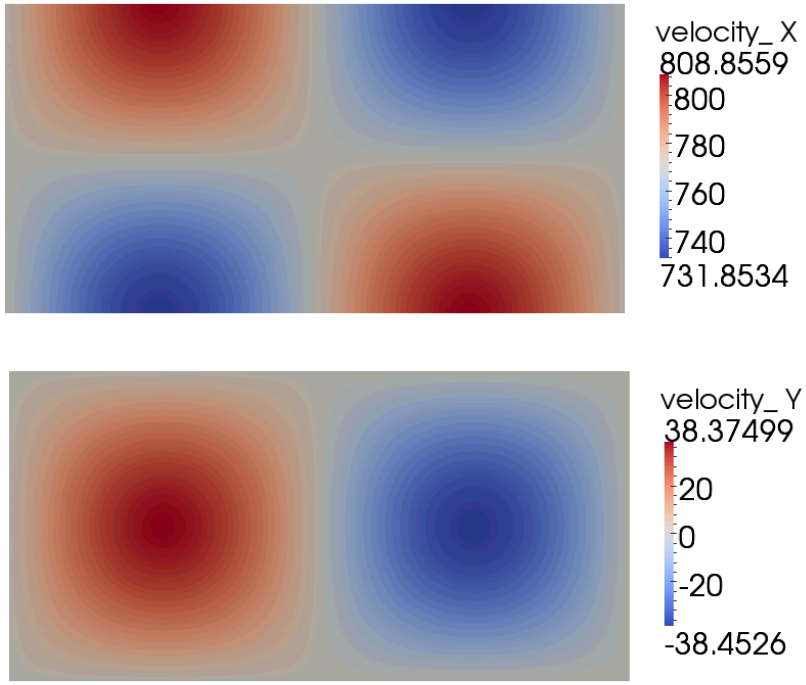
- Method of Manufactured Solutions (MMS): substitute the solution into the differential equations to get source terms
- Typical approach: use Mathematica/SymPy to generate explicit code for source terms
- New approach: Use Automatic Differentiation (AD) to build the sources directly in the code

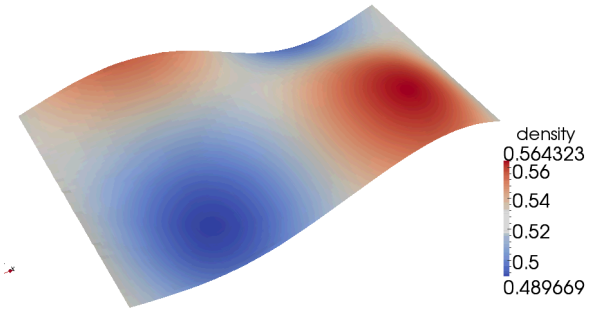
```
trace_e = diff(u, x) + diff(v, y) + diff(w, z)

p = (lambda + 2*mu/3)*trace_e

exx = diff(u, x) - trace_e/3
exz = (diff(u, z) + diff(w, x))/2
```

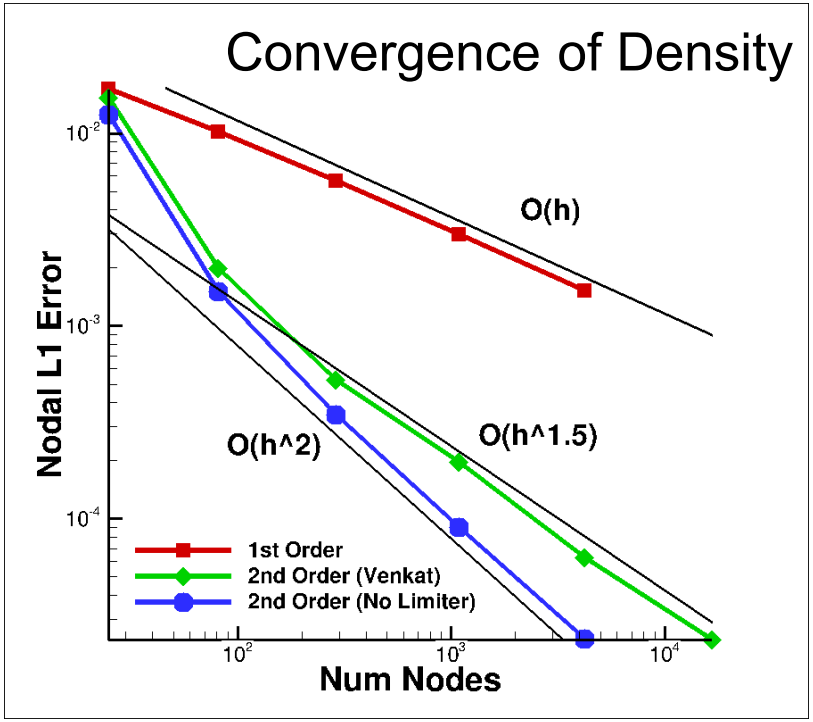
```
sxx = 2*mu*exx + p
sxy = 2*mu*exy

F_x = diff(sxx, x)
    + diff(sxy, y)
    + diff(sxz, z)
```

Code Verification of Sierra/Aero Code

Sandia National Laboratories







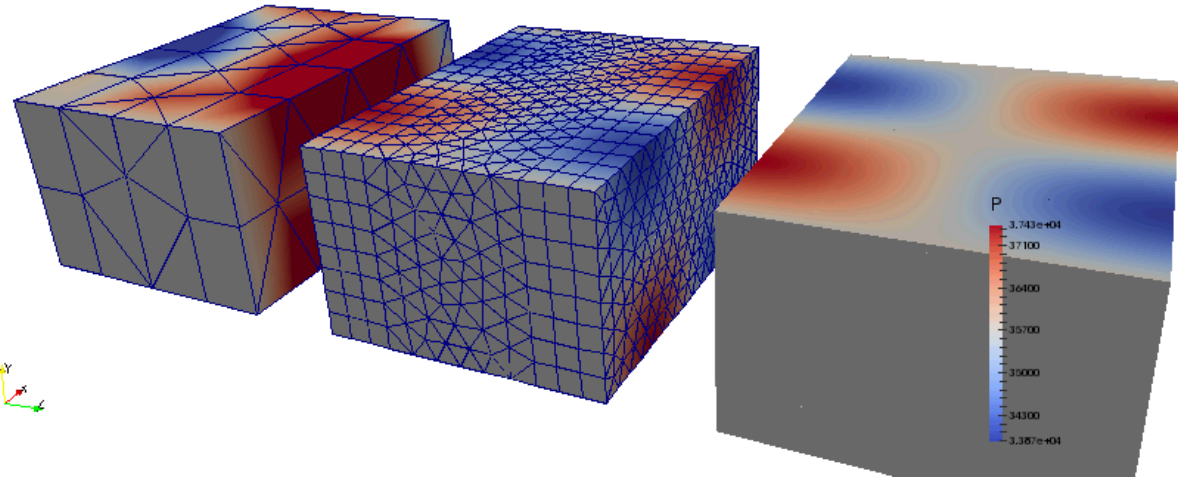
Inviscid flow: perturbation of uniform supersonic channel flow (parallel to walls): Y-velocity is 5% of X-velocity

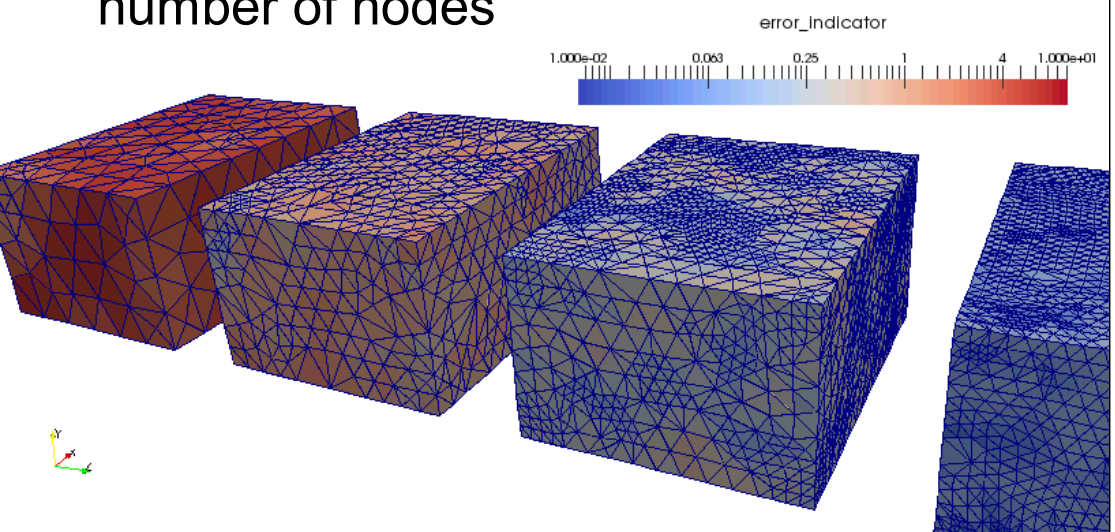
Code Verification of Sierra/Aero Code

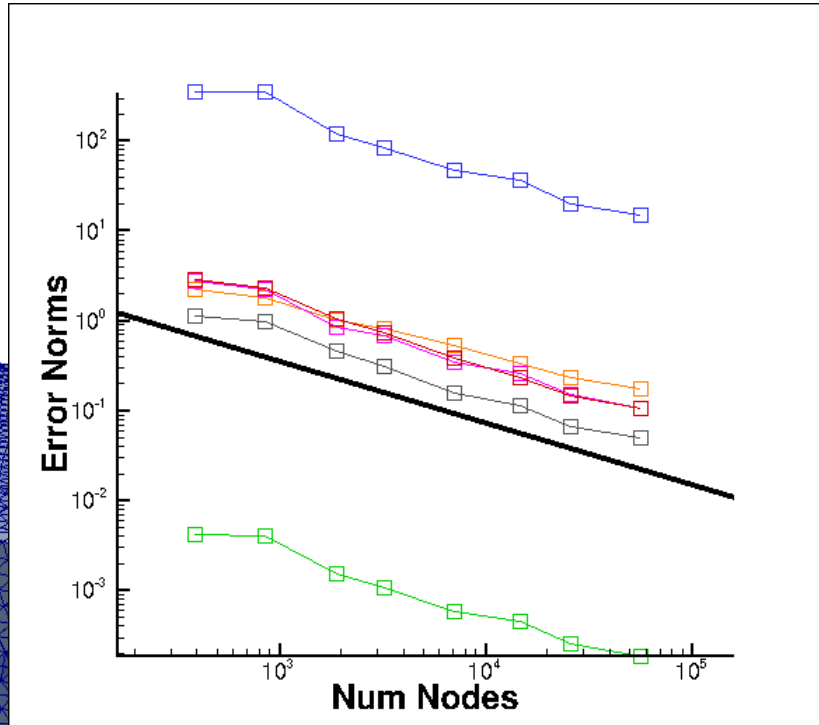
Sandia National Laboratories

Hybrid mesh verification using Euler MMS test, using transfer for ICs

Offline adaptivity for tet meshes also using Euler MMS; We see optimal convergence w.r.t. number of nodes







Sierra/SM Code Verification

Sandia National Laboratories

- Elements and Materials
  - Large deformation patch tests (sanity type test for distorted elements in uniform stress)
  - Small and large deformation convergence tests
  - Use of MMS for hyper-elastic material models
  - Convergence tests over a range of aspect ratios
- Contact
  - Classical (linear elastic) contact problem convergence tests
- Temporal Integration (Dynamics)
  - Convergence tests for spring-mass system
  - Convergence tests for linear elastic wave propagation in a rod
- Other key capability areas that lack sufficient testing
  - Inelastic material models
  - Material failure models
  - Coupled thermo-elasticity
  - Joint models

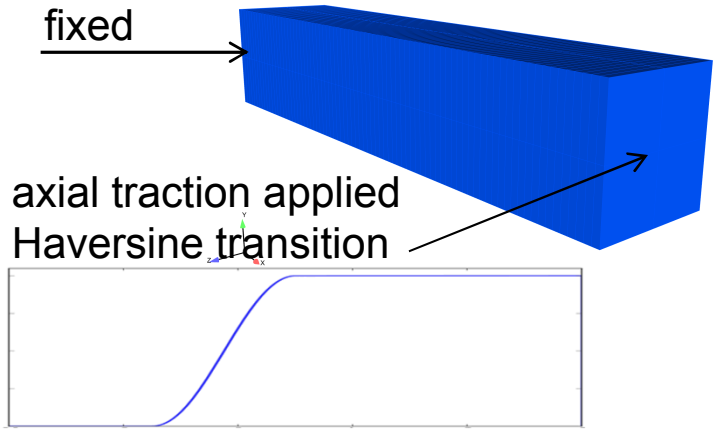
SM Verification Testing

Sandia National Laboratories

Parallel consistency questioned by analysts

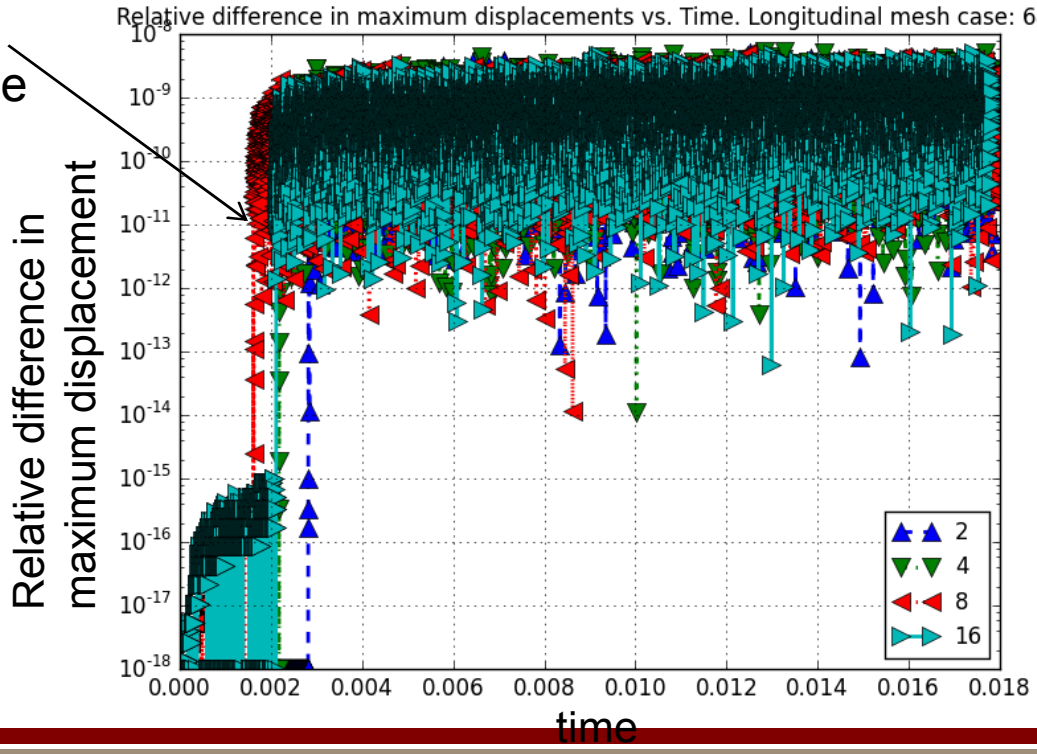
Tested two key physics of analyses:

- wave propagation and contact



Adapted wave propagation tests to examine differences in single-multiple processor results

- Jump in differences was a concern
- Round-off error analyses with perturbed mesh showed same issue
- Traced to form of hyper-elastic models – magnifies round-off
- Parallel consistency OK



Sphere contact problem showed parallel consistency OK for contact

Caveat: two tests do not imply parallel consistency can not occur.

SM Verification Testing

Another analyst priority for testing: Contact

Existing contact verification tests are Hertz type solutions, which assume: linear elastic response and small contact area.

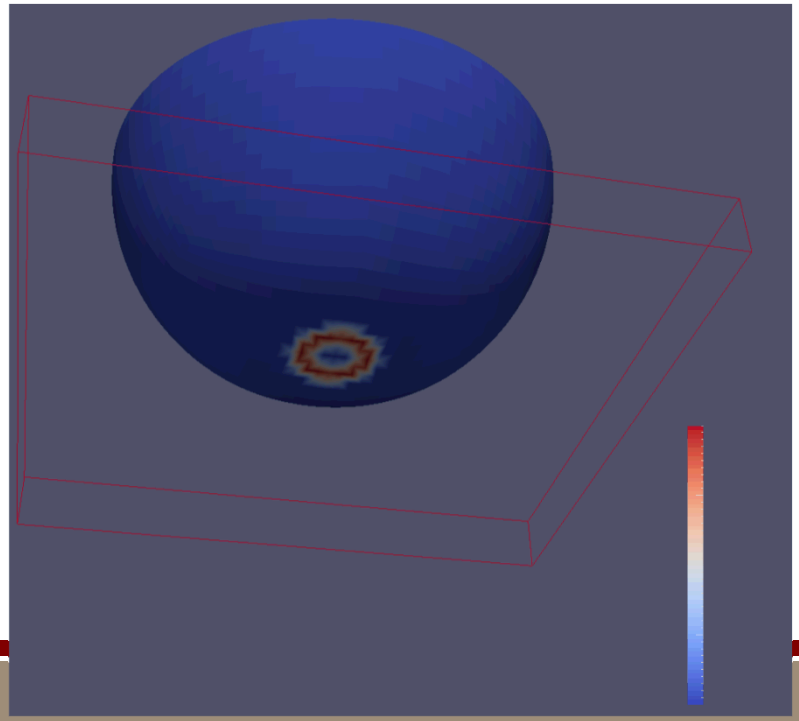
Objective: top-down contact convergence tests that are much closer to the application space.

top-down => weaker tests where we do not have the exact solution, but examine tendency to converge.

Incrementally increasing complexity:

- Large deformations & contact area
- Explicit analysis
- Inelastic (elastoplastic) material behavior
- Friction

Large elastic deformation case exhibits convergence but requires significantly finer mesh than typical system models.

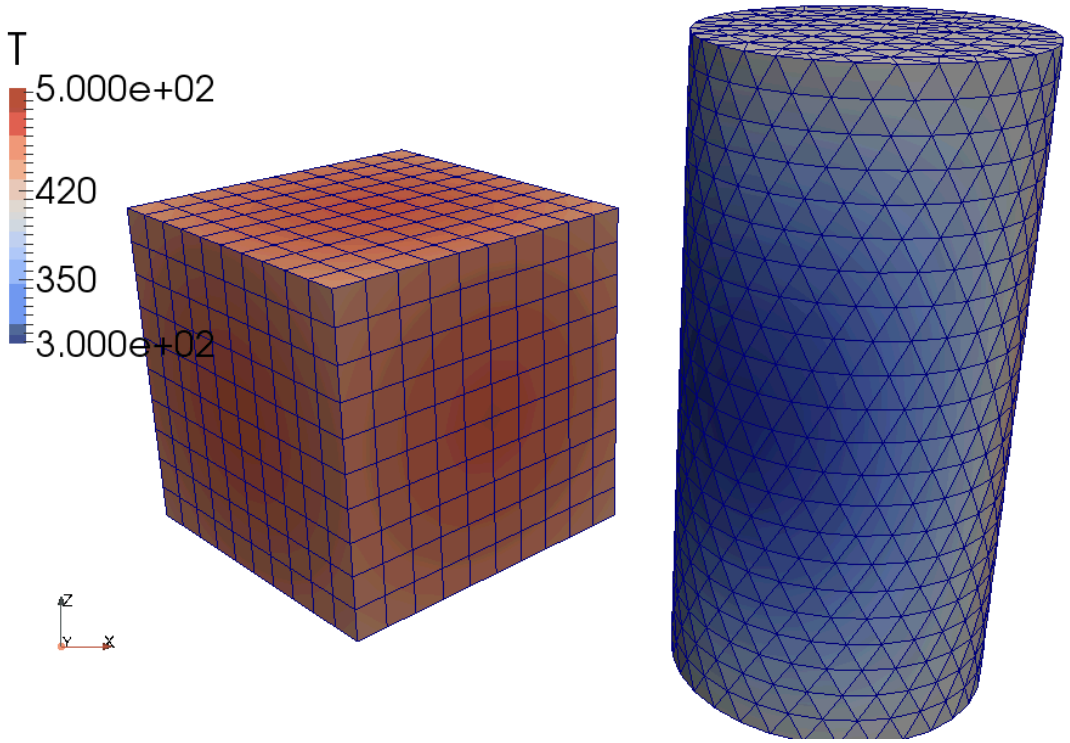


Thermal: Local Coordinates

- **Verification gap**: local cylindrical coordinates in material
- Users can specify anisotropic conductivity using local coordinates
  - Cartesian, with rotated coordinate frame
  - Anistropic conductivity (constant values)
- We generated MMS tests for these two cases

Both tests use the same rotated coordinate system

Both demonstrate 2nd order convergence



Coupled Code Verification

Sandia National Laboratories

- Fluid-structure – normal environment (captive carry/vibrational loading)
- Thermal-mechanical – abnormal environment (crash/burn)

FSI Example: Acoustic Piston

- **Motivation**: produce a solution with zero initial condition and smooth ramp up in time
- Initial displacement, velocity, and acceleration are zero
- Solution is linear displacement in time plus damped spring solution

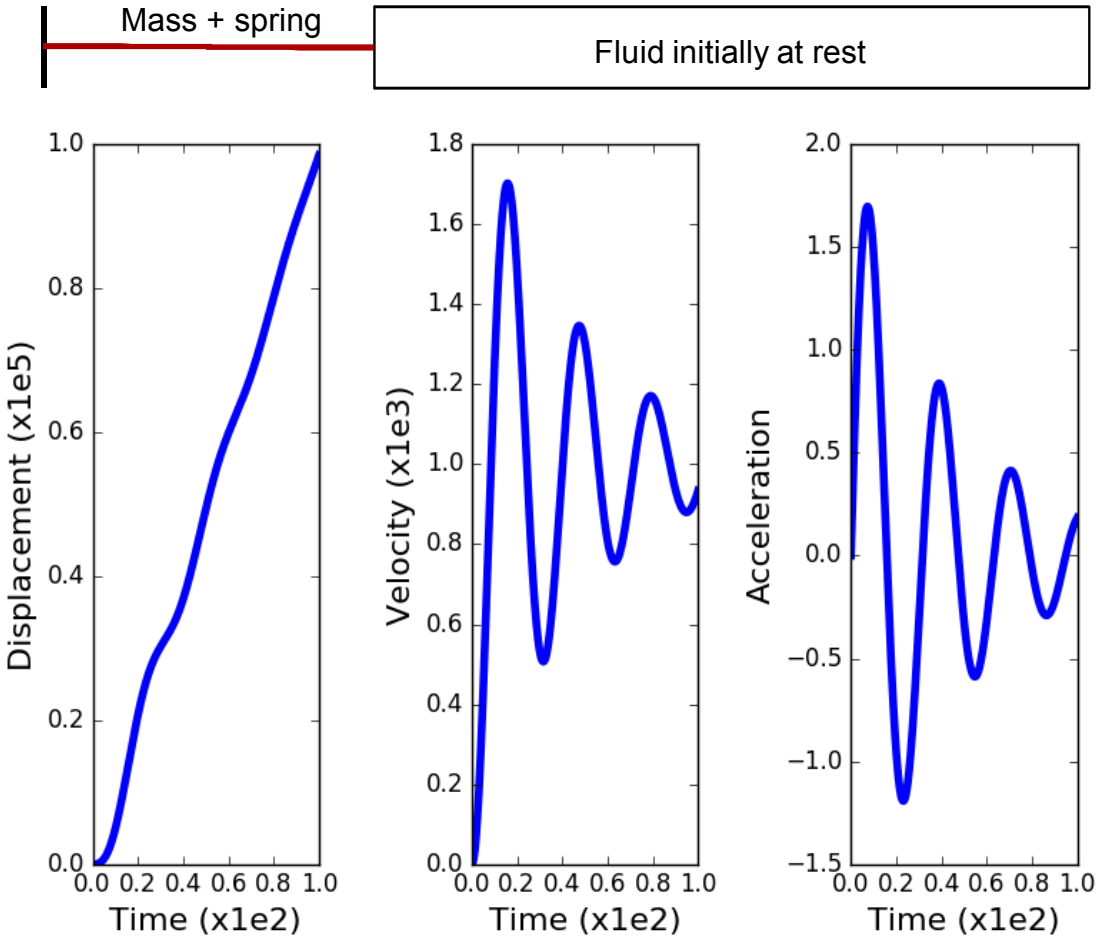
$$d = \frac{c}{2m}, \omega = \sqrt{\frac{k}{m} - d^2}$$

$$\beta = \frac{c}{k}, a = \bar{v}\beta, b = \frac{(ad - \bar{v})}{\beta}$$

$$ma + cv + ku = \bar{v}kt$$

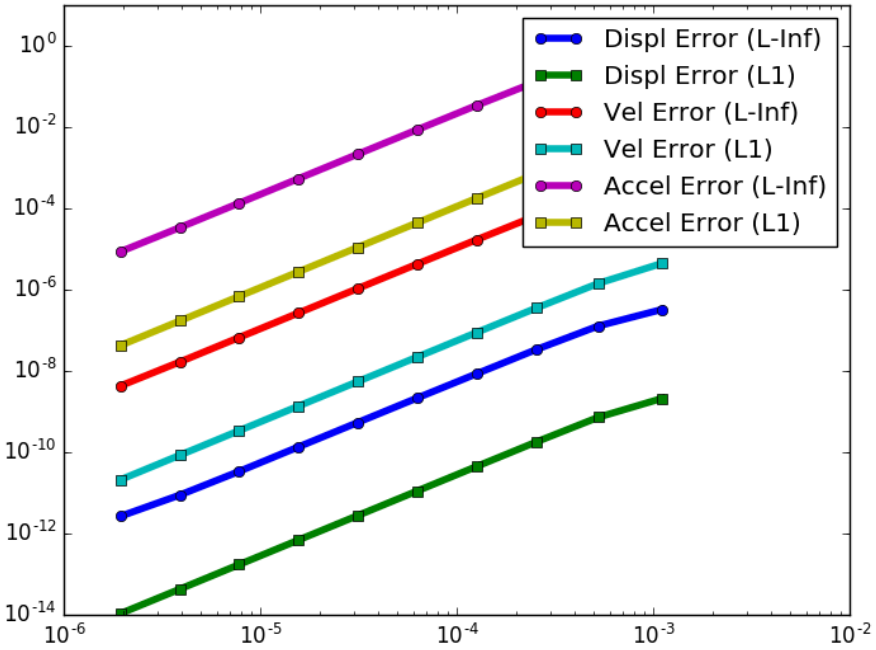
$$v(0) = u(0) = 0$$

$$u(t) = e^{-dt}\left(a\cos(\omega t) + b\sin(\omega t)\right) + \bar{v}(t - \beta)$$



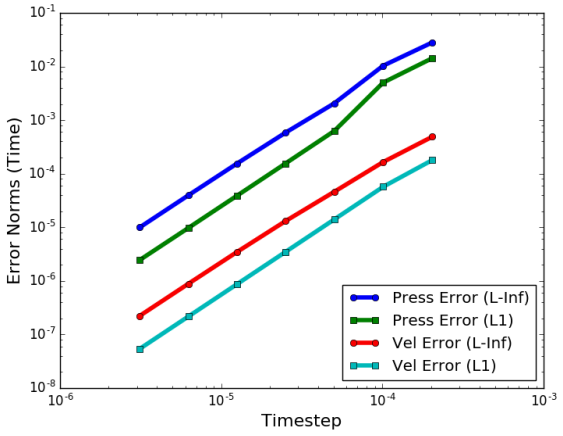
Piston: Structural Dynamics only

- One shell element connected to one spring for each node
- Damping applied based on constant damping coefficient
- Body force using time dependent load function on the shell
- Newmark time integrator
- All variables converge second order

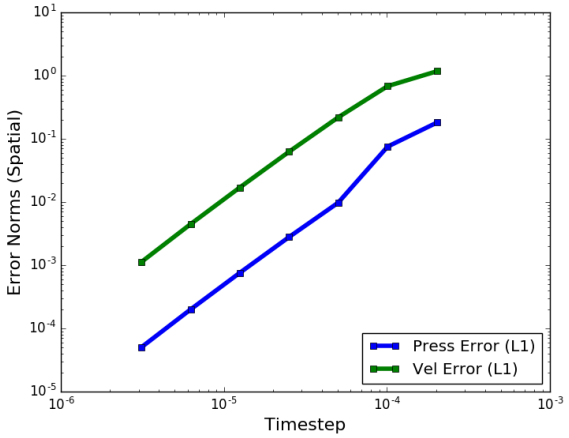


Piston: Aero Only

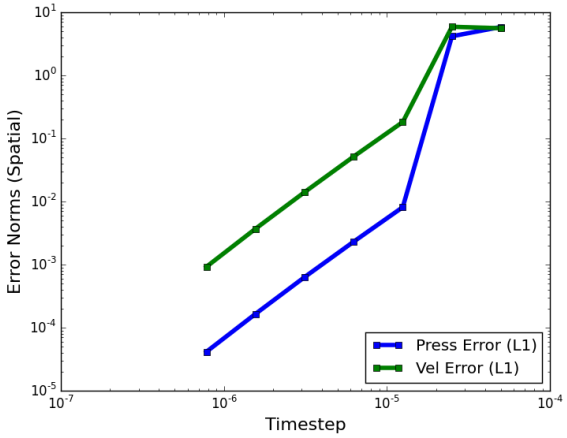
- Piston solution applied as displacement BC on fluid
  - Time integrators: BDF2 and AB2
  - Meshes with 50-3200 elements
- Verify fluid pressure, velocity, and displacement at piston surface
- Verify fluid pressure/velocity using spatial norms



Temporal error at piston surface (BDF2)



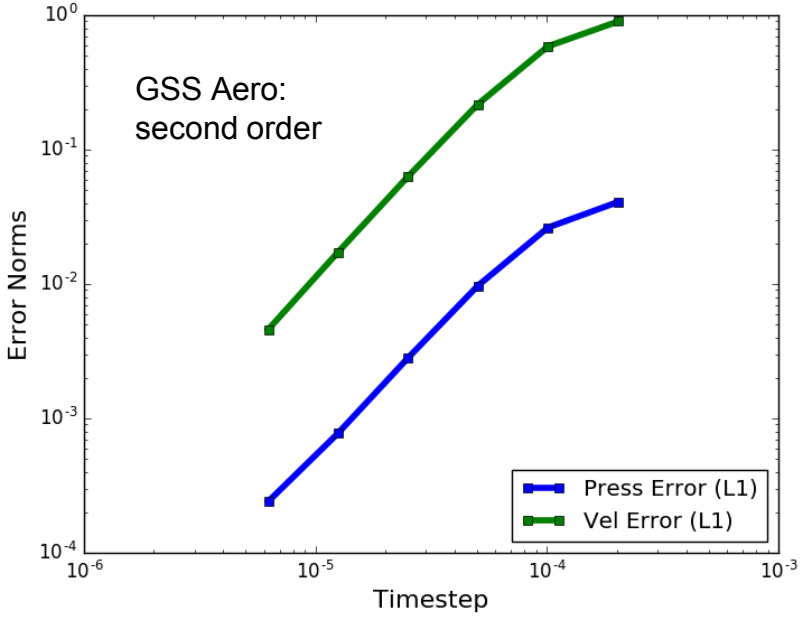
L1 spatial error (BDF2)

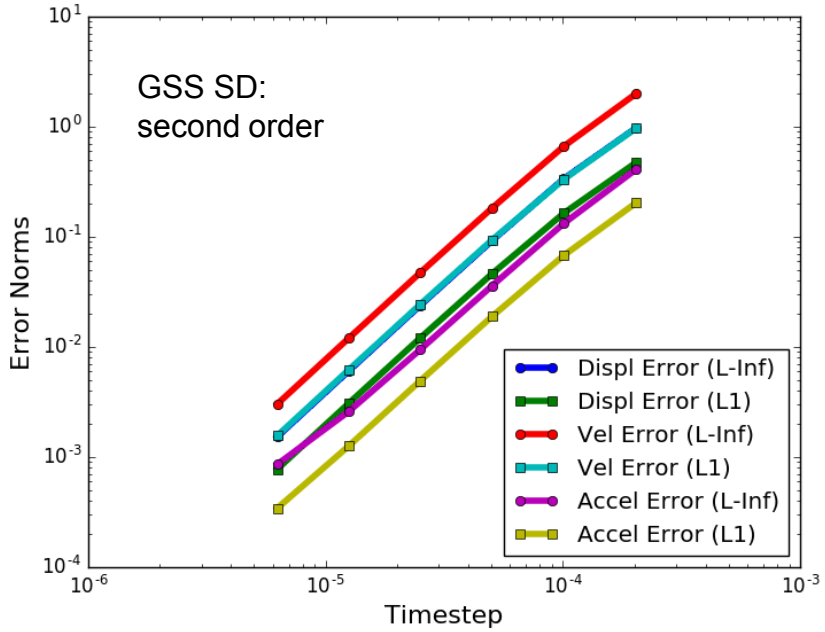


L1 spatial error (AB2)

Piston: Coupled Convergence

- Test now exercises two-way coupling
  - Piston damping comes from applied fluid force
  - Fluid displacement comes from solid
- Verify coupling accuracy
  - CSS should be first order
  - GSS should be second order



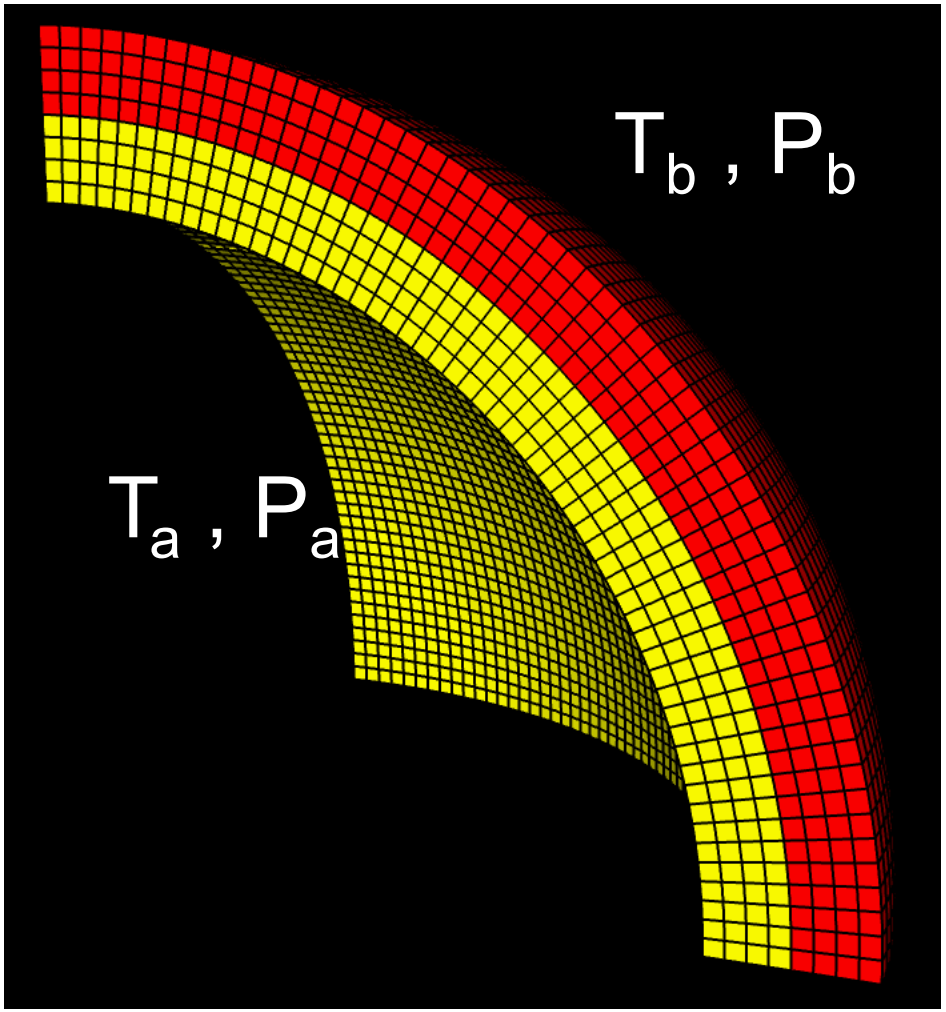


Thermal-Mechanical Coupling Pressurized Sphere

Sandia National Laboratories

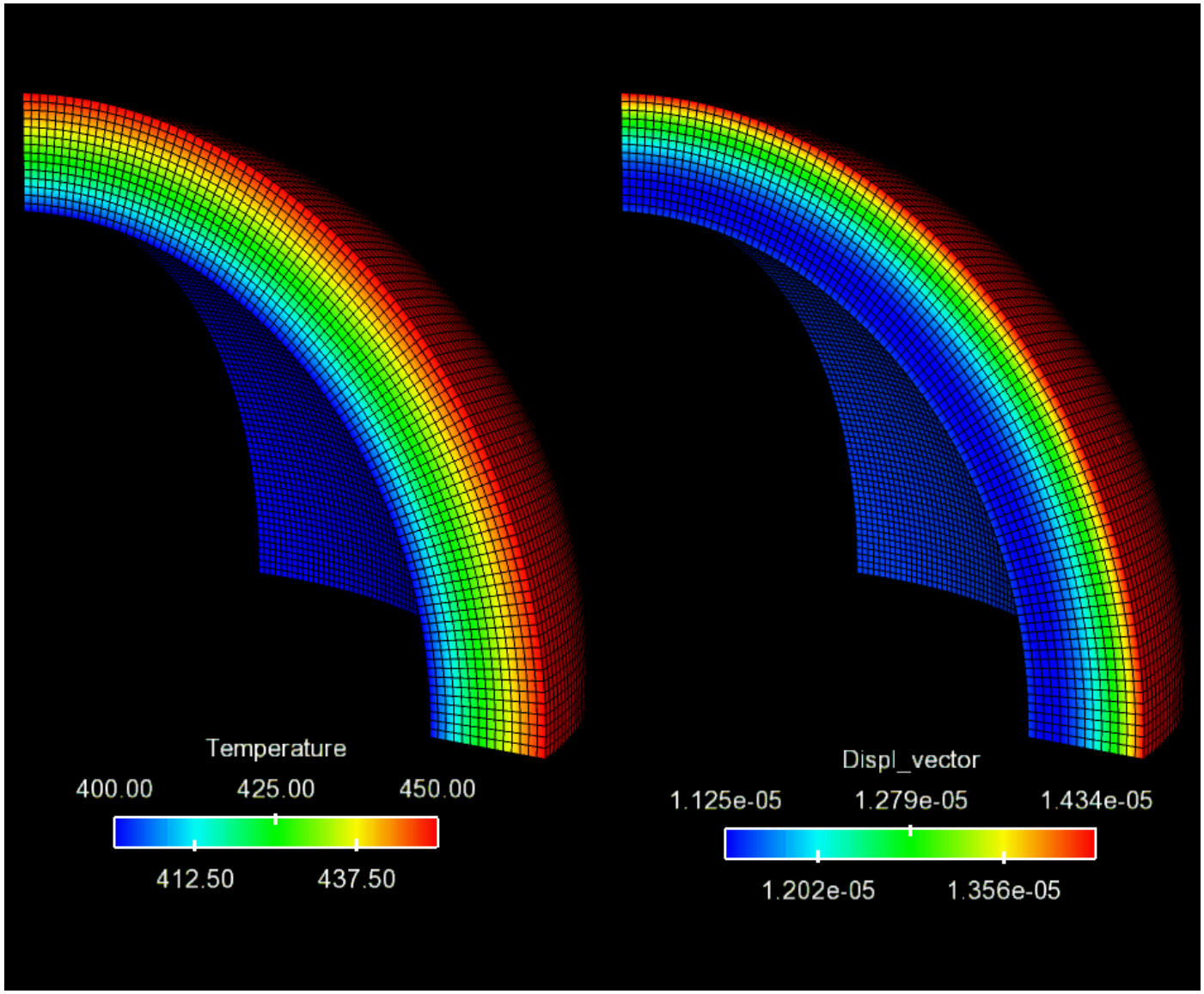
Features tested
- Thermal Expansion
- Pressurization
- Contact

Exact solution assumes linear mechanics

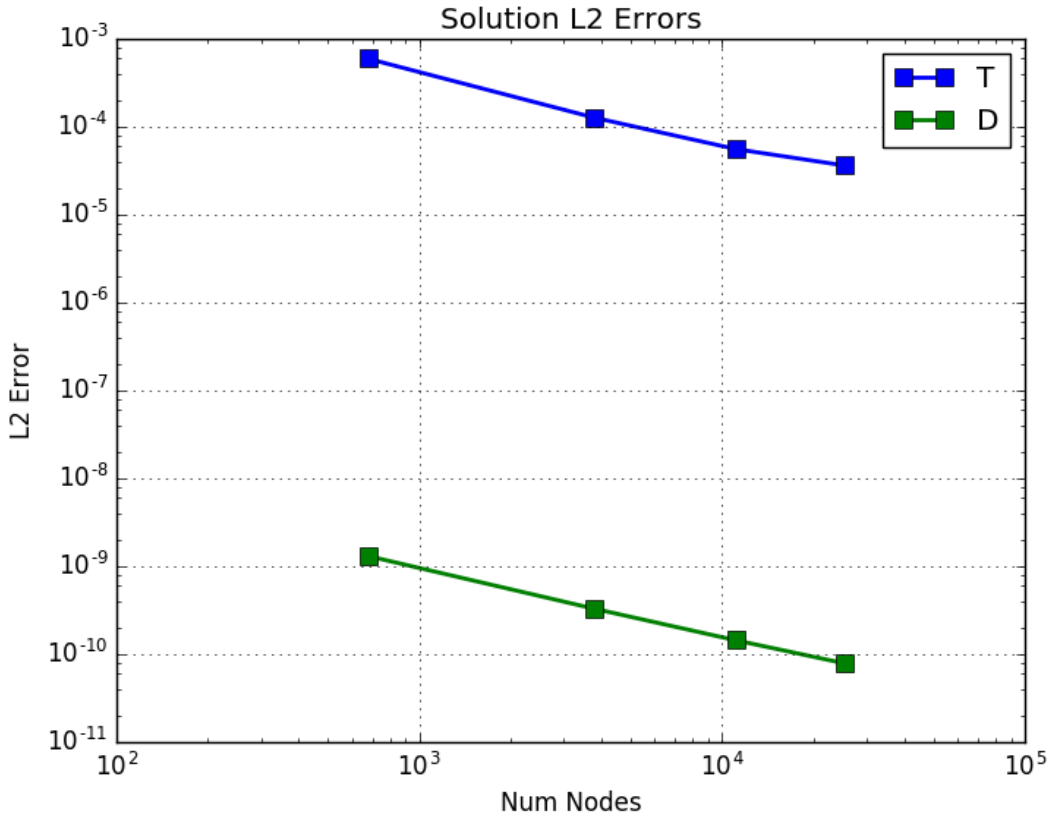


Coupled Problem Solution

Sandia National Laboratories



Thermal-Mechanical Solution Error



Solution Verification Tools

Sandia National Laboratories

- Percept mesh adaptivity
- Compare tool
- Mesh scaling

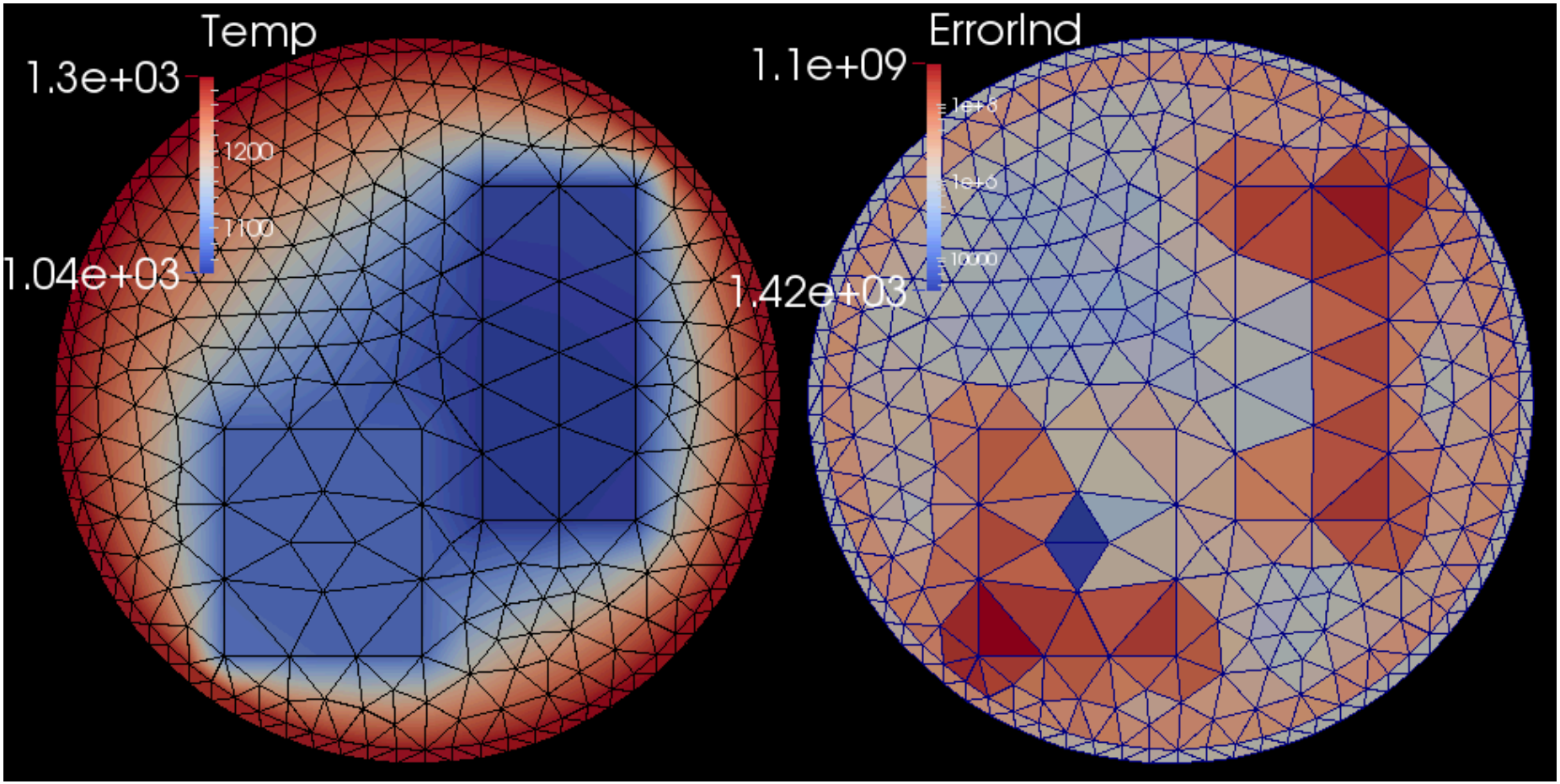
Percept Mesh Adapt Tool

Sandia National Laboratories

- https://github.com/PerceptTools/percept
- Refine almost any mesh (uniform mesh refinement = UMR)
- Enable offline or inline adaptive mesh refinement (AMR)
- Scalable, parallel operation
- Support for geometry (CAD or mesh-based)
- Advanced features:
  - Mesh smoothing
  - Convert low order elements to high order
  - Respect mesh spacing on refined mesh

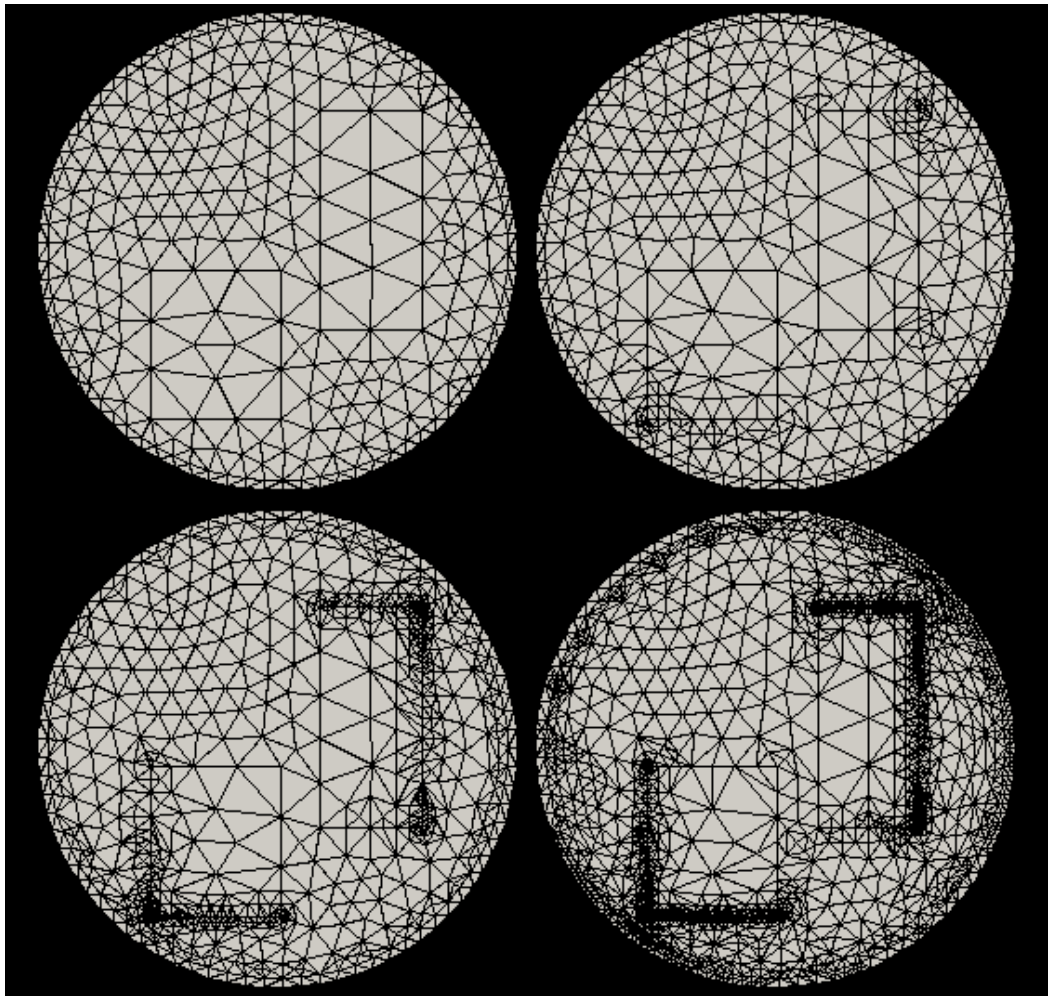
Error Indicator (Thermal)

- Gradient recovery indicator (error in thermal gradients)



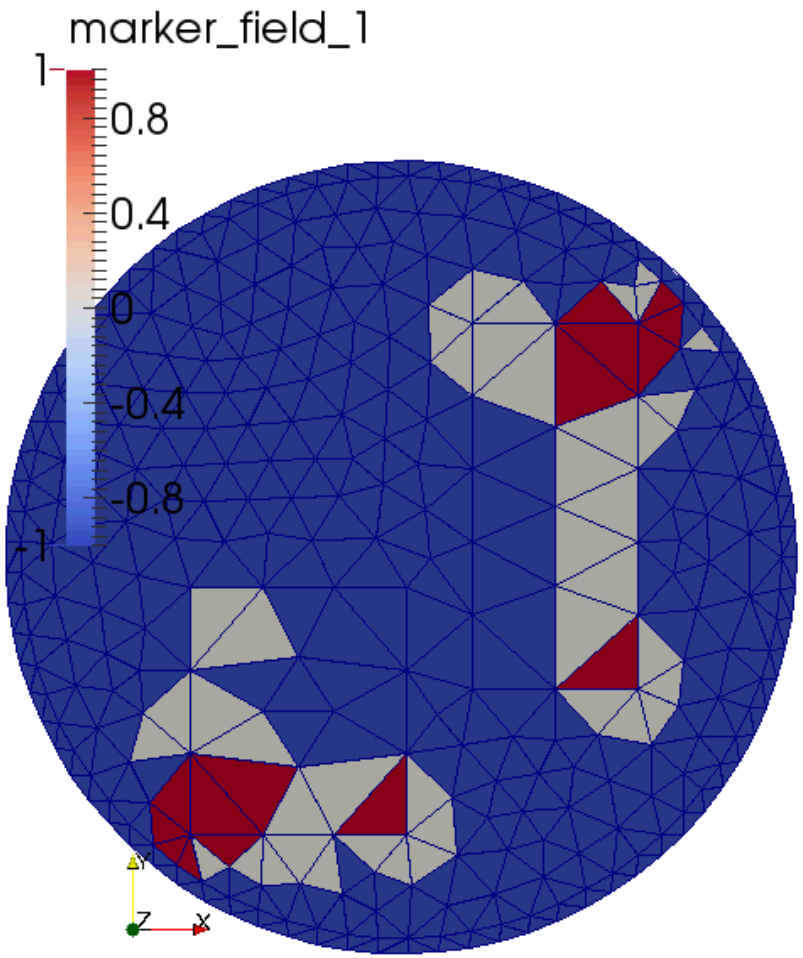
Locally Adapted Meshes

- Approach here was to generate new adapted meshes for each transient run
- Like UMR but with locally refined meshes
- With mesh size widely varying, we focus on number of nodes or elements



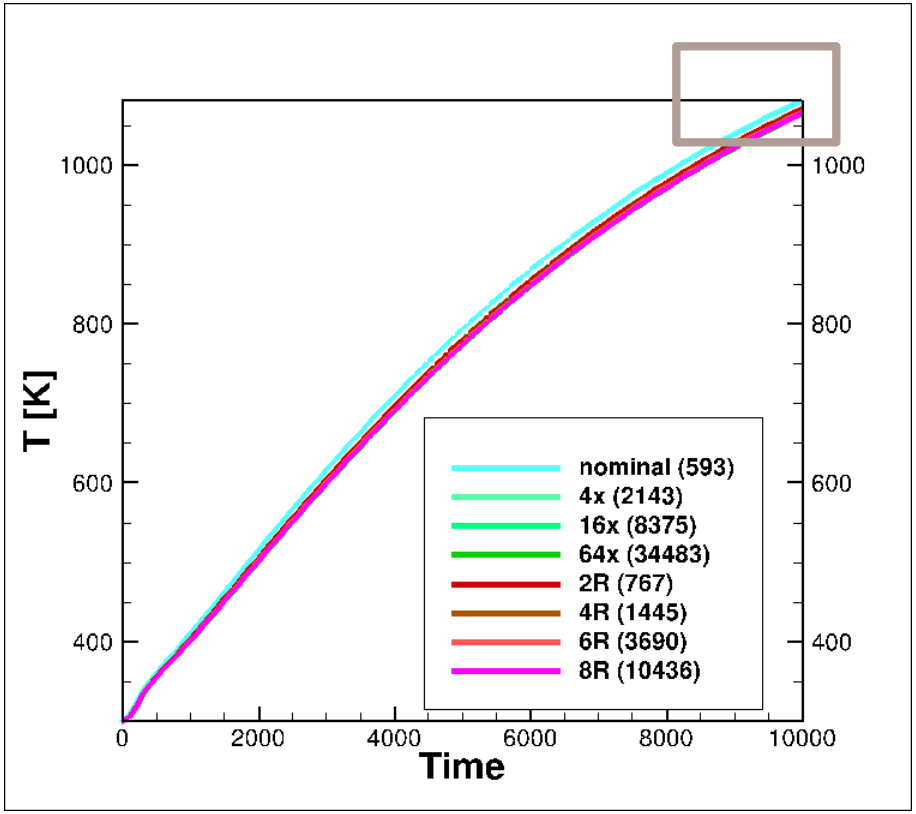
Adaptivity: Marking and Refinement

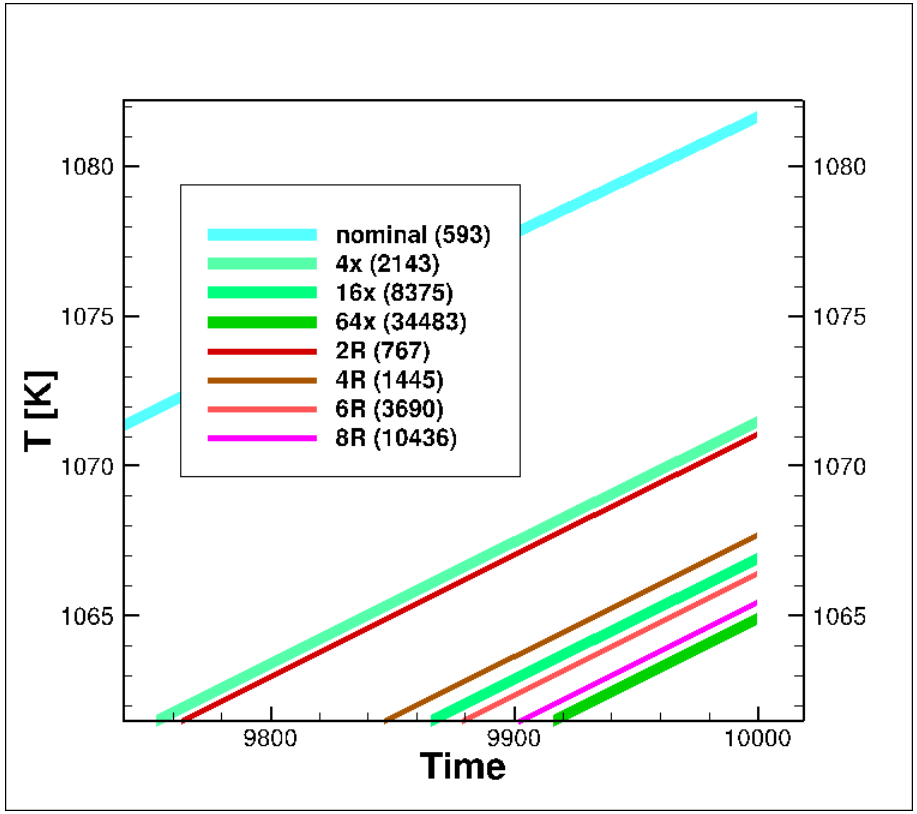
- Once we have an indicator of local error contributions, we then
  - Mark for refine/coarsen
  - Adapt the mesh
- The resulting adapted mesh:
  - Has a nearly uniform distribution of error indicators
  - Is conforming (no irregular nodes)
  - Has smooth change in element size
  - Respects the original geometry



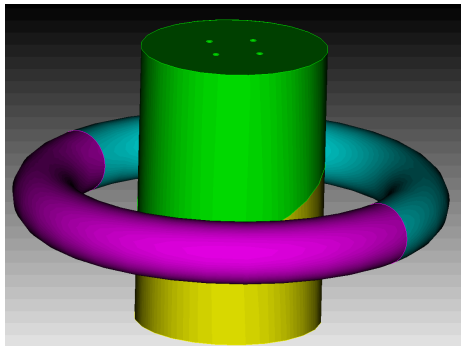
Comparing UMR and AMR

- Solution from AMR (thin lines) is as accurate as UMR solution (thick lines), but using fewer nodes

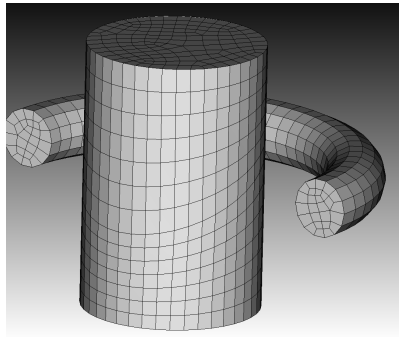




Compare tool (Cubit)

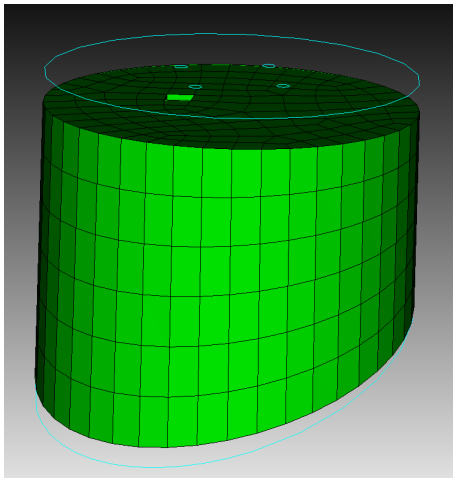


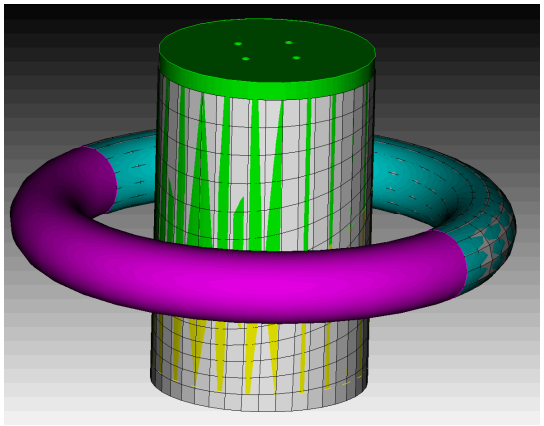
CAD



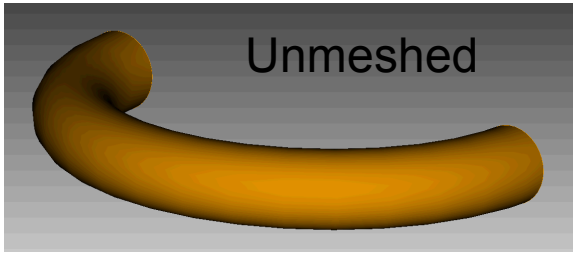
Mesh

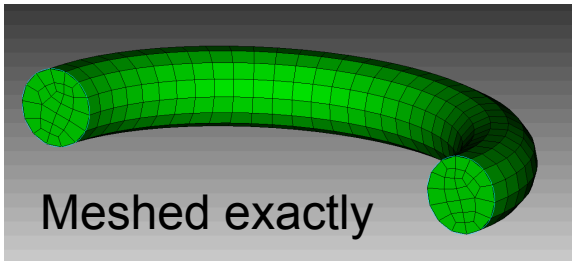
Partial match

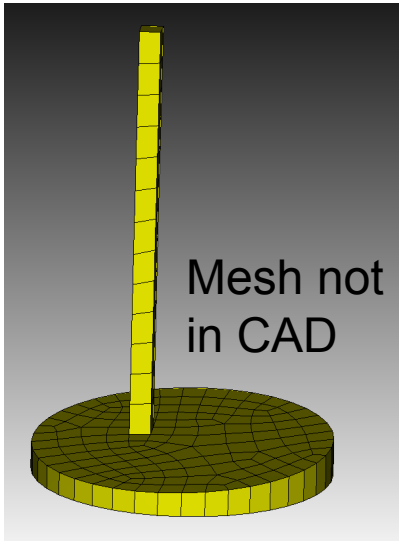




Is the mesh consistent with the CAD?

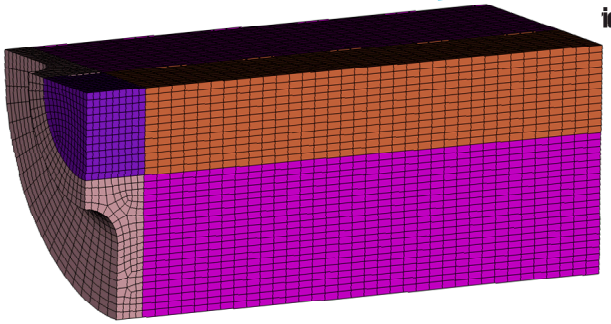






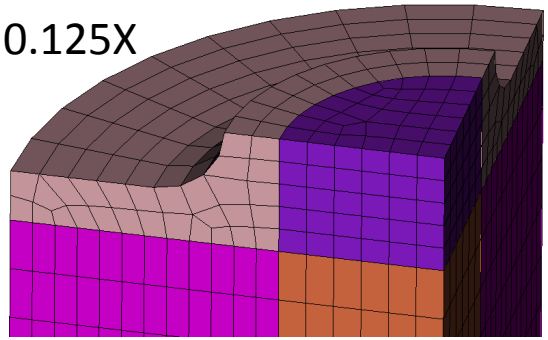
Affordable Solution Verification through Mesh Scaling

- Available in Cubit and Sierra
- Notch tension test with 1/8 symmetry
- Elastic-plastic material mode, implicit solver
- QoIs: Reaction force, max equivalent plastic strain
- Mesh scaling is used to coarsen/refine meshes using an arbitrary scale multiplier (not just 8x, 64x)

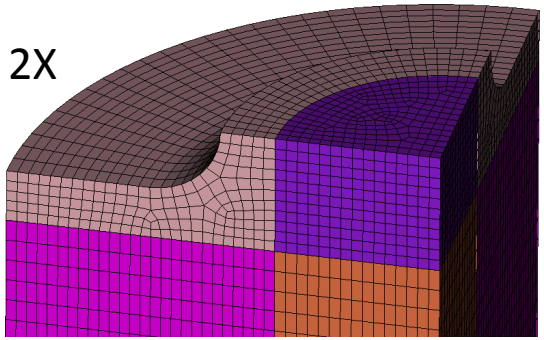


Original Mesh: 22,860 Hexes
1 element blocks
4 nodesets
Has CAD Geometry

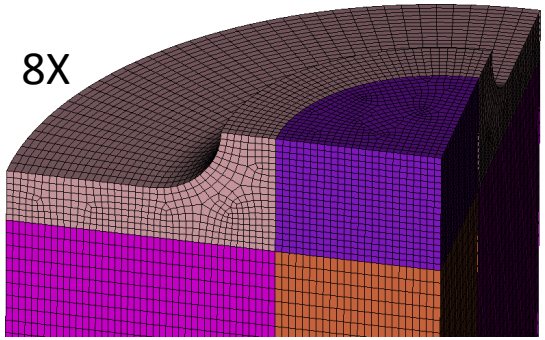
0.125X



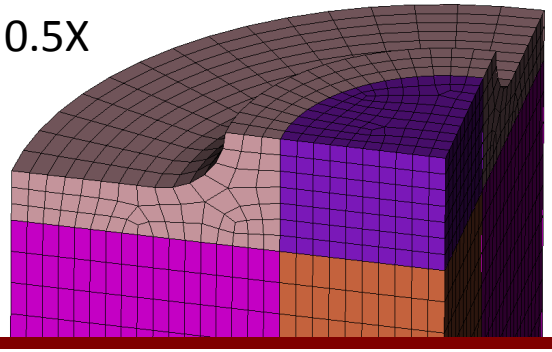
2X



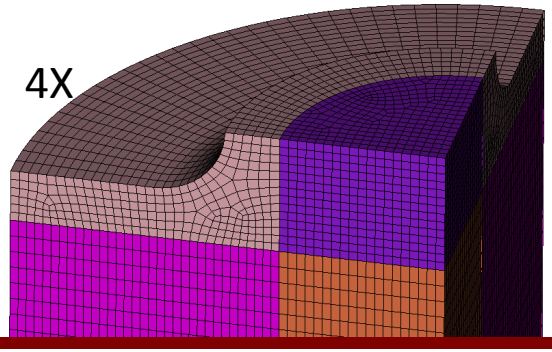
8X



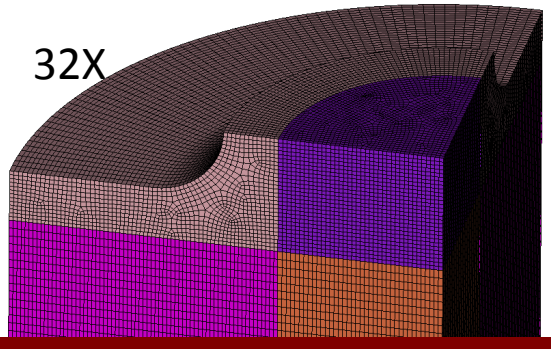
0.5X



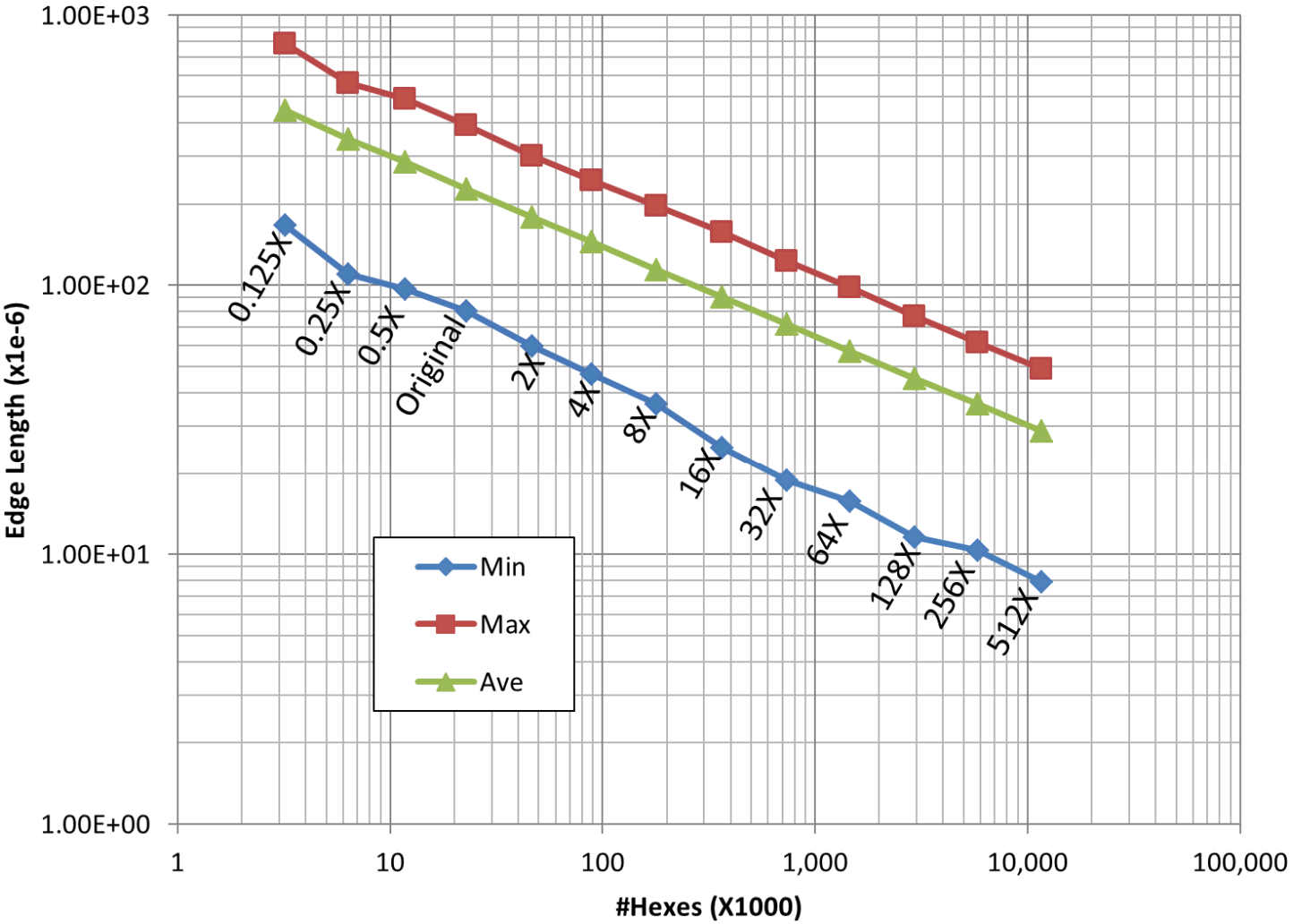
4X



32X



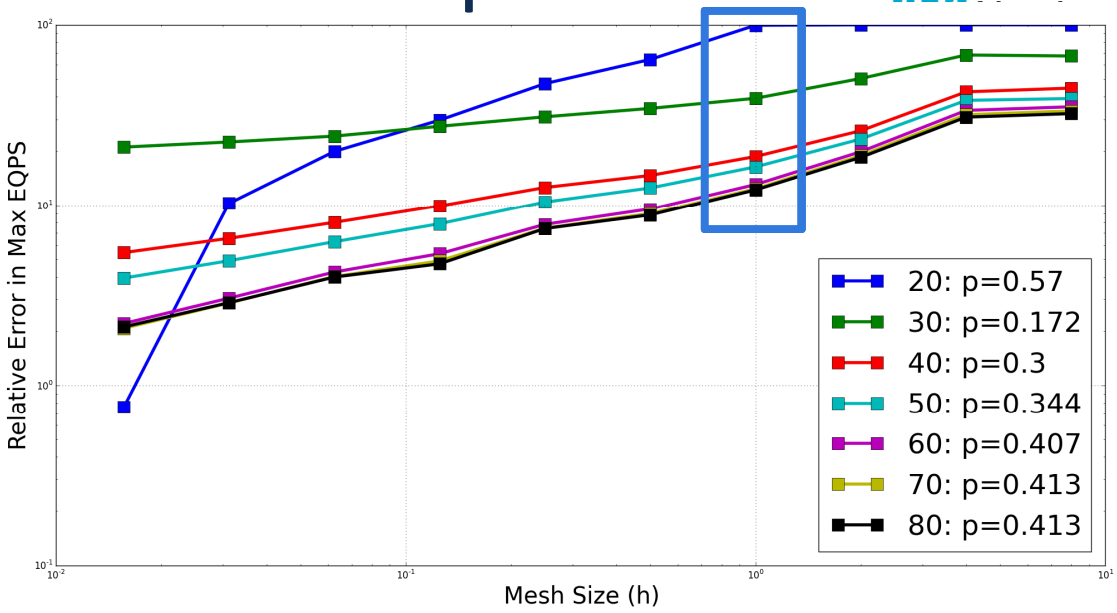
Smooth Variation of Edge Lengths

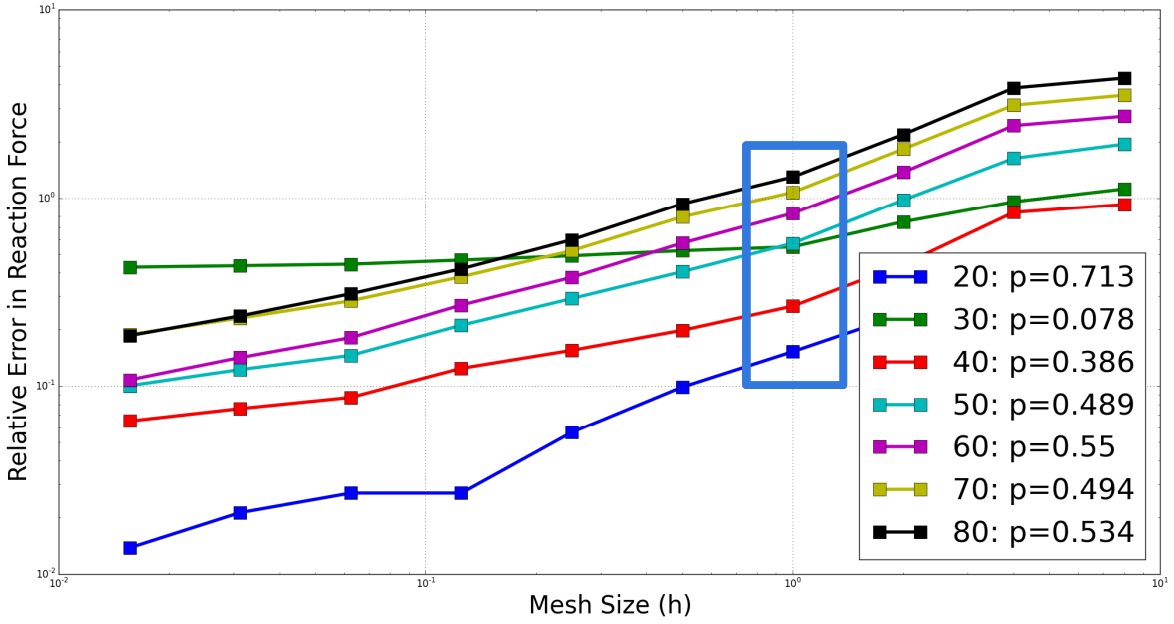


Geometric constraints & surface paver are reasons for variance from exact linear in Min/Max values, especially in coarsening.

Errors Using Multi-Mesh Extrapolation

- Extrapolation using nonlinear least squares approach (using base mesh plus 4 finer meshes)
- Estimated rates for max EQPS approach 0.4 at end of load steps
- Extrapolation provides useful error estimates





- Estimated rates for reaction force approach 0.5 at end of load steps
- Extrapolation provides useful error estimates
- Nominal mesh error: 0.1-2%

Summary

- One of the main reasons for practicing code and solution verification is to increase the confidence in CompSim results
  - PCMM is a infrastructure to communicate this confidence and its supporting evidence
- Code Verification supports credibility by testing the code to independent benchmarks and insuring accuracy
- An extensive suite of verification tests has been developed and documented for the Sierra codes
- Tools for solution verification can accelerate assessment of numerical errors