SAND2013-0049C

Verification Overview

Brian Carnes

Validation and Uncertainty Quantification Processes (1544)

AWE/SNL SIERRA Executive Meeting

Jan 7-9, 2013

Sandia National Laboratories

Motivation

- We build codes for modeling/simulation of high-consequence events
- How do we know the codes give correct answers?
  1. Correct for the overall prediction means code can predict physical events with accuracy (e.g, Validation)
  2. Correct for verification means that we have quantified errors from approximate math models (Verification)
  3. Correct may also mean that the code is free from bugs
- We need to provide evidence of this correctness to our customers

What is Verification?

- Verification is the process of quantitatively assessing the accuracy of a code for:
  - Quantification of discretization errors (item 2)
  - Removal of coding errors (item 3)

- Related concepts: approximation error, discretization error, numerical error, error from meshing, etc.
- These errors all arise from approximations of the governing equations (math model), regardless of the predictive capability of the math model.

Verification Activities

- At the code development level:
  - Verification testing (next)
  - Software Quality Engineering* (SQE)
- At the application level:
  - Mesh refinement studies (later in talk)
  - A posteriori error estimation*
  - Mesh adaptivity*

*Not discussed in this talk.

Verification Testing

- Big goal: verify that 100% of the code functions correctly
- Realistic goal: test the most critical code features

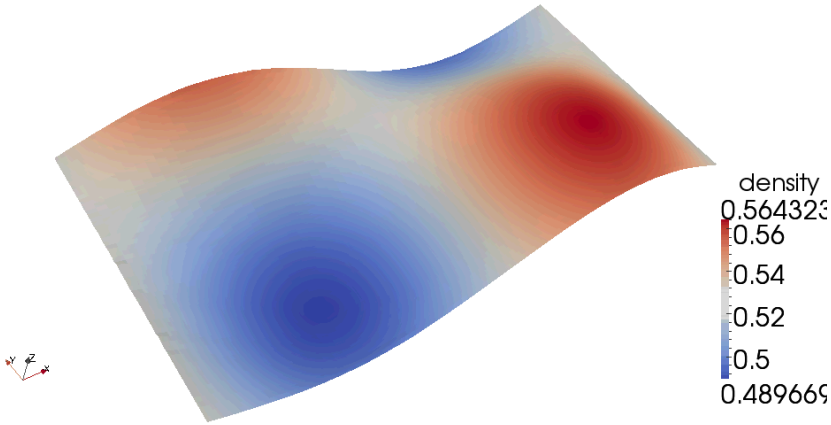
- How do we select critical code features?
  - Using input files / meshes from the user community (customer focused)
- Our tool is called Feature Coverage Tool (FCT)
  - Compares an input file against the verification tests
  - Identifies features not covered (gaps)
  - Identifies coverage of pairs of features

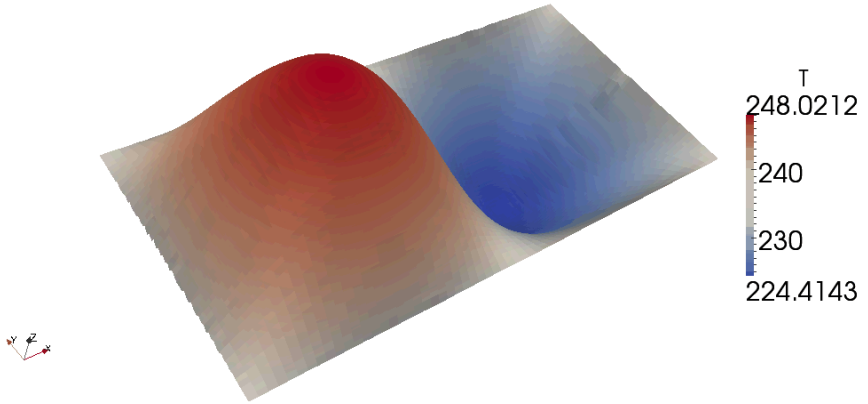
Verification is Quantitative

- Verification tests **must** contain an exact, quantitative criterion based on the underlying math model:
  - Mass/force conservation
  - Symmetry
  - Patch tests
  - Analytic value of a derived quantity
  - Analytic solution for all of the solution variables
- This excludes things like code-to-code comparisons, beam theory, asymptotic solutions, etc.
- The last case is most desirable and enables:
  - Mesh refinement studies and order of accuracy estimation

Manufactured Solutions

- Where do we find analytic solutions?
- We make them up! These “manufactured” solutions required source terms and boundary conditions
- In spite of their non-physical nature, they provide excellent quantitative criteria for verification tests





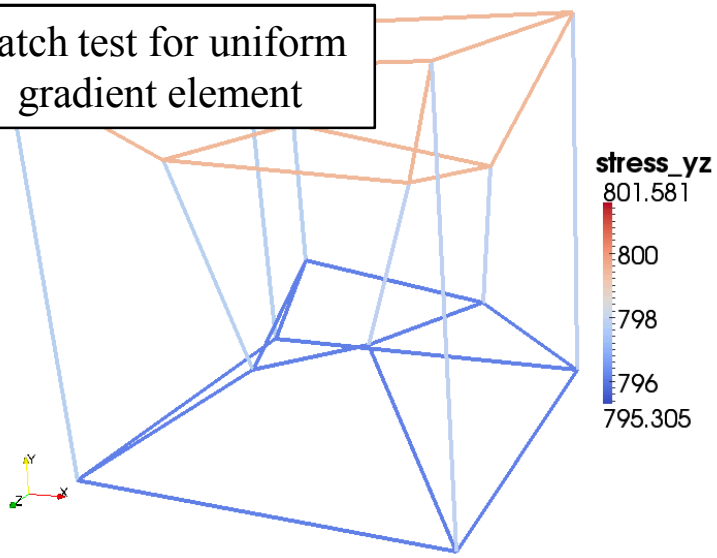
Sandia National Laboratories

SIERRA Verification Tests

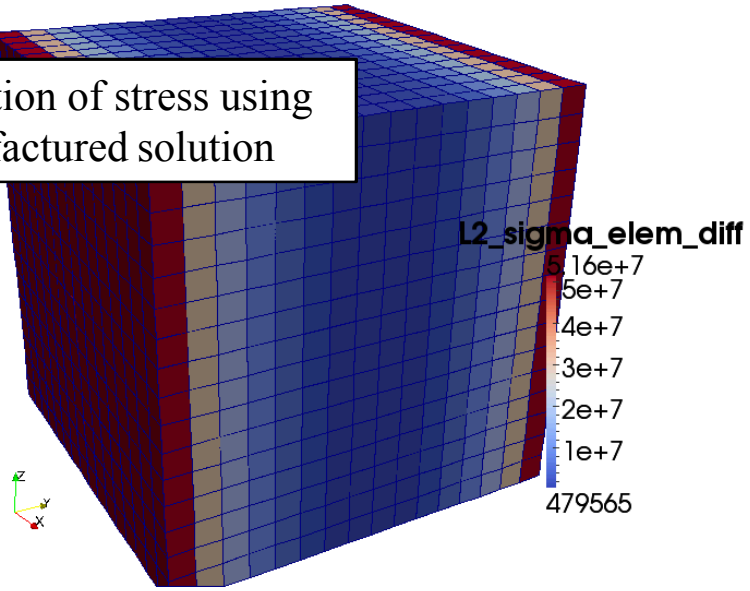
- A search of SIERRA tests with “verification” keyword:
  - SD: **1022** SM: **490** TF: **169**
- Verification tests should contain:
  - Input file(s)
  - Mesh(es)
  - Definition of the quantitative criterion
  - Means to compute error metrics for each mesh
  - Documentation

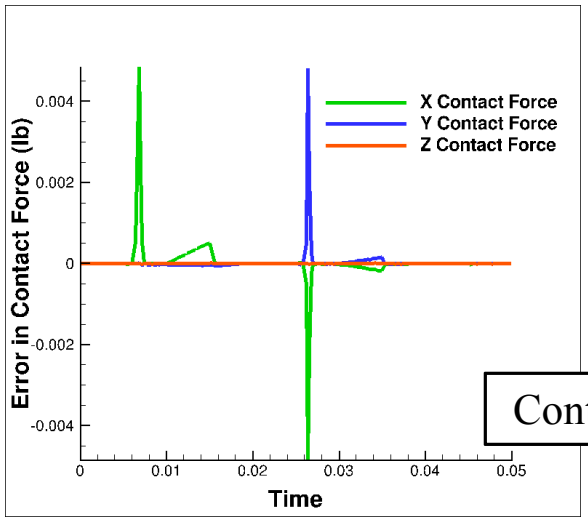
Example: Verification of SIERRA/SM

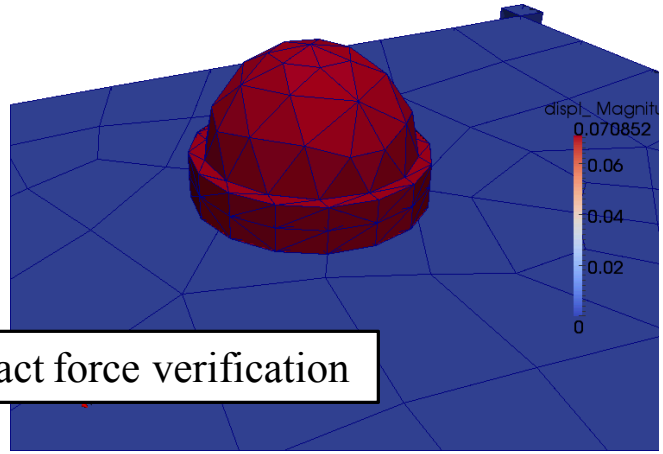
Patch test for uniform gradient element



Verification of stress using manufactured solution





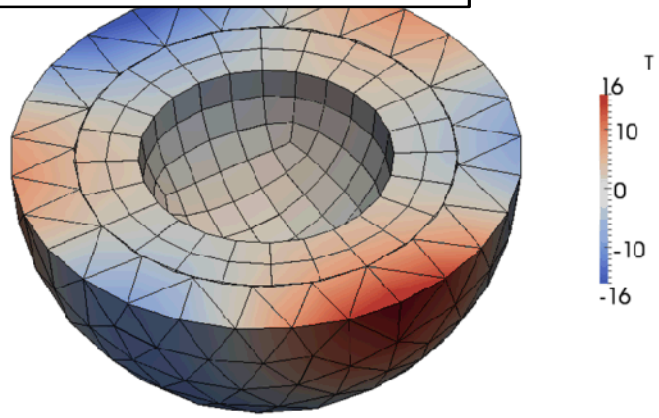


Contact force verification

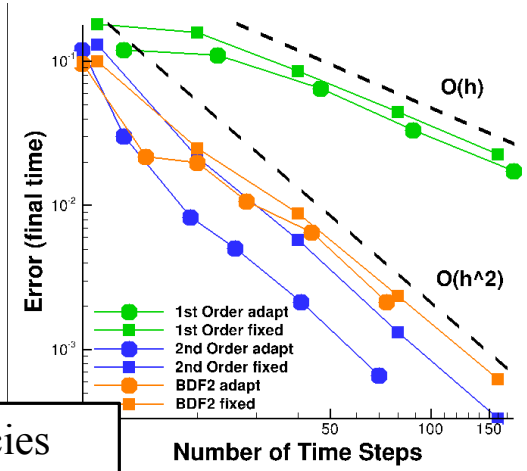
Sandia National Laboratories

Example: Verification of SIERRA/TF

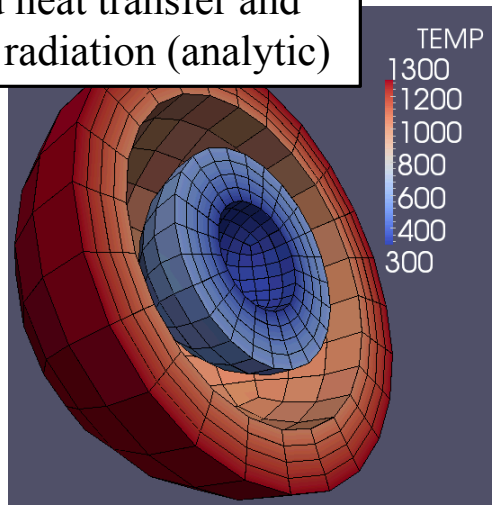
Thermal contact with non-matching grids (manuf. soln.)



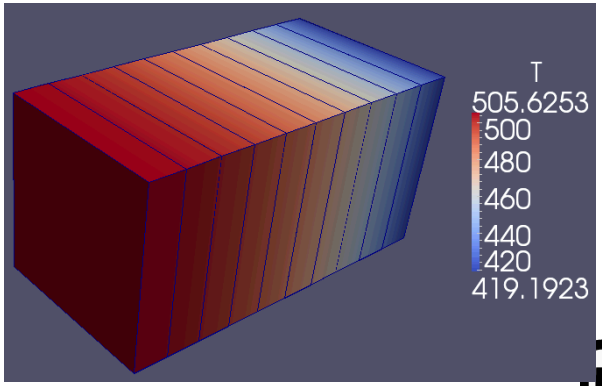
Adaptive time stepping algorithms (manuf. soln.)



Coupled heat transfer and enclosure radiation (analytic)



Chemistry – species evolution and heating



Sandia National Laboratories

Mesh Refinement Studies

- What is the accuracy of a specific simulation?
- Here we assume that the code has been verified
  - Can test using Feature Coverage Tool
- This is called solution/calculation verification
- The main objective: **select a mesh that delivers suitable accuracy for a given cost**, but also:
  - Solver tolerances
  - Time step size / tolerance
  - Contact search tolerances
  - Hourglass stiffness

Extrapolation of Solutions

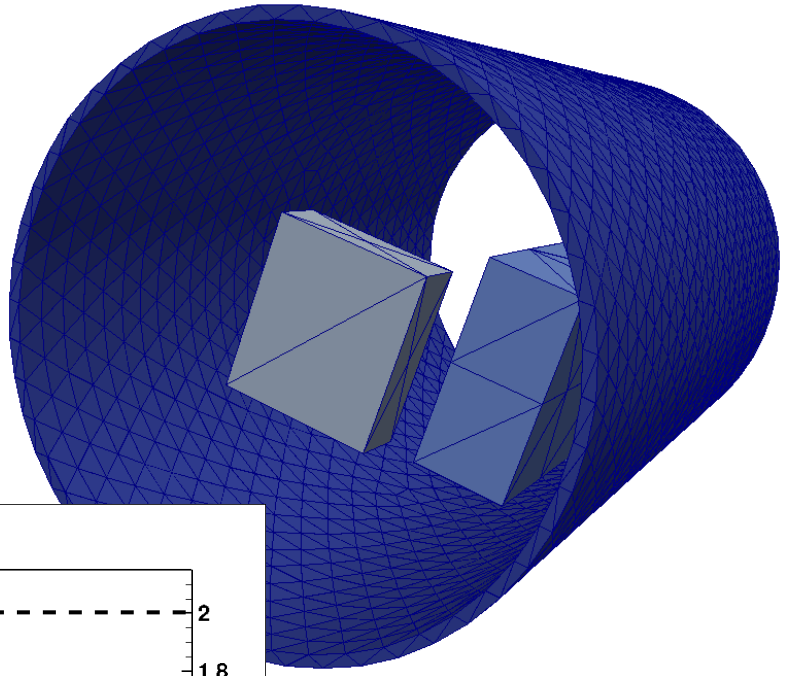
- Solutions (U) are functions of mesh size (h)
  - $U = U(h)$
- Suppose we have a sequence of mesh sizes ($h_1 > h_2 > h_3$) and monotone solutions
  - $U_1 > U_2 > U_3$ or $U_1 < U_2 < U_3$
- Then we can extrapolate the true value of U^e and errors for each mesh
  - $E_i = U^e - U_i$
- More generally we can consider approaches based on curve-fitting and median statistics (Bill Rider)

Generating Meshes

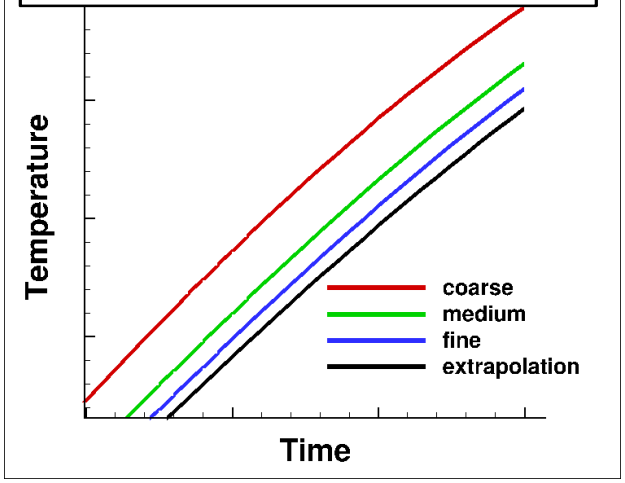
- A sequence of meshes is needed (decreasing h)
- Mesh doubling is common (and we have good tools) but is cost-prohibitive for large problems
- Instead, any sequence of mesh sizes can be used
  - Should respect model geometry under refinement
  - Should be as uniform as possible (also elem. quality)
- For general meshes we estimate mesh size:
  - h=(1/N)^D
  - N=number of nodes / elements
  - D=spatial dimension

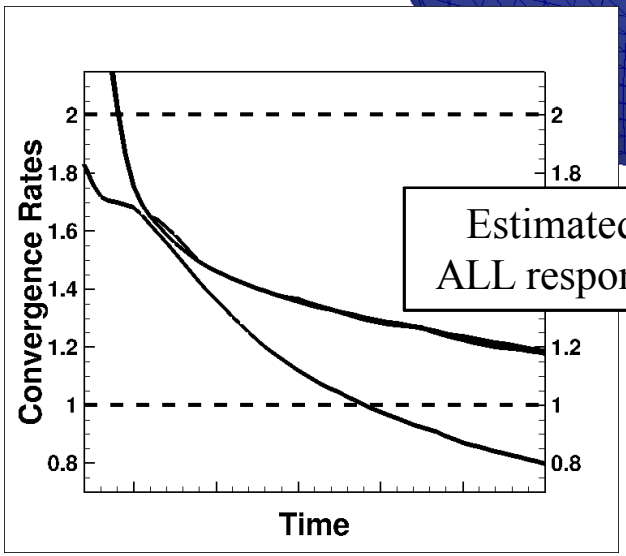
Example: Mock Thermal Analysis

- Note: not actual weapon system
- Applied external heat source
- The derived quantities are max/min temp. on components, temp. at point
- We use three meshes and extrapolate the temperature at each time step



Extrapolated MIN temperature (lower curve in black)



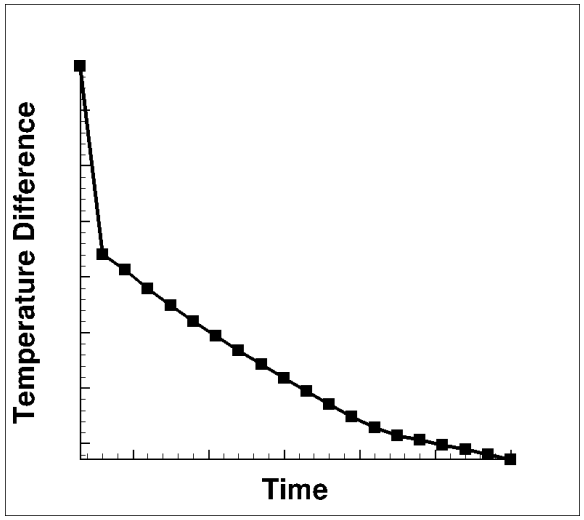


Estimated convergence rate for ALL responses: close to first order

Sandia National Laboratories

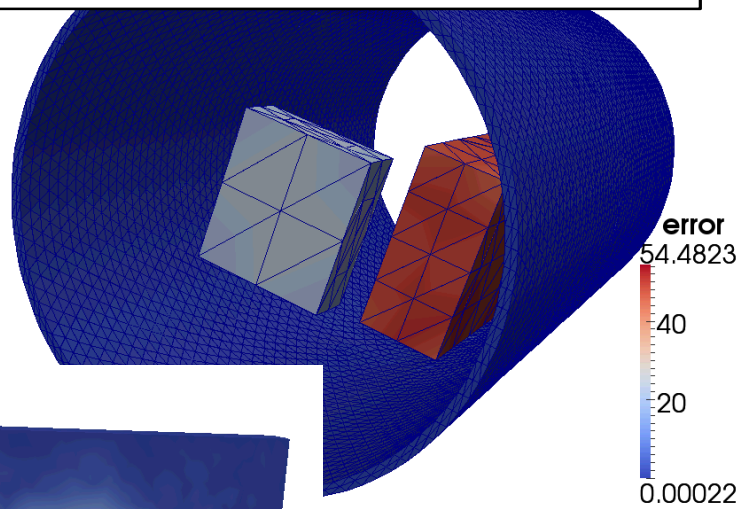
Example: Solution Transfer

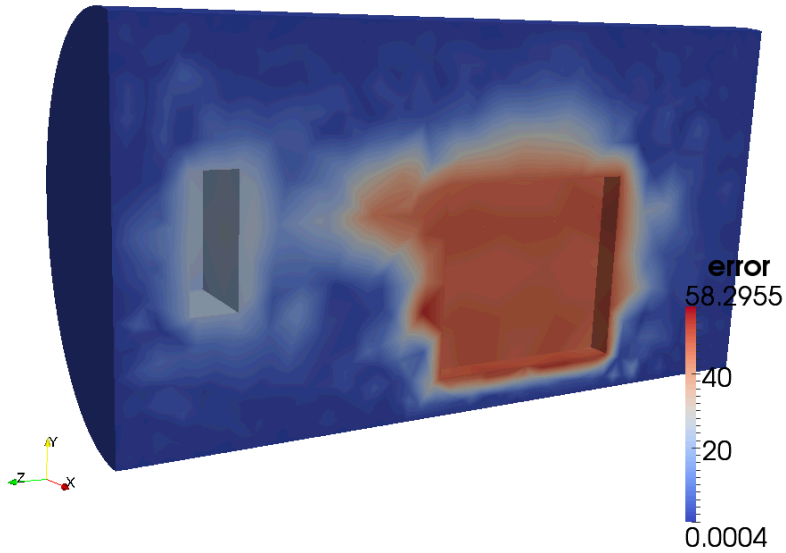
- Note: not actual weapon system
- We can also determine local errors by transferring solutions (coarse -> fine)
- Here dominant errors



Max nodal temperature difference between coarse and medium meshes

Distribution of nodal temperature error between coarse and medium meshes





Sandia National Laboratories

Summary & Discussion

- There are many code verification tests in SIERRA
  - Could improve documentation and quality
- Tools exist for verification
  - Feature Coverage Tool
  - Auto-generate manufactured solution source terms
  - Convergence analysis scripts for mesh refinement
  - Code-specific (embedded) verification tools
- How usable are the tools?
- Are we providing the right assistance?
- How do we resolve cases where mesh refinement fails to exhibit convergence?