SAND2017-0960PE

Mesh Adaptation in Albany

Glen Hansen, Andrew Bradley, Brian Granzow, Dan Ibanez, Mario Juha, Mauro Perego, Seegyoung Seols, Mark Shephard

Derived from SAND2015-5946C, SAND2014-0249P

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. 2015-XXXXP

Overview

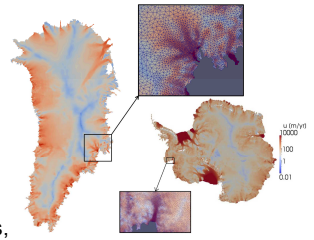
- Albany supports a wide variety of application physics areas including heat transfer, fluid dynamics, structural mechanics, plasticity, quantum device modeling, climate modeling, and many others
- Mesh adaptation is supported using LOCA and the Piro transient time integrators
  - Most application of adaptation is within the LCM project to support large deformation analysis
  - FELIX ice sheet example was presented at ATPESC 2016 https://github.com/gahansen/Albany/wiki/PAALS-Tutorial-2016
- Adaptation is supported using the SCOREC mesh database and tools, integrated into Albany by the FastMATH SciDAC project

Albany ecosystem

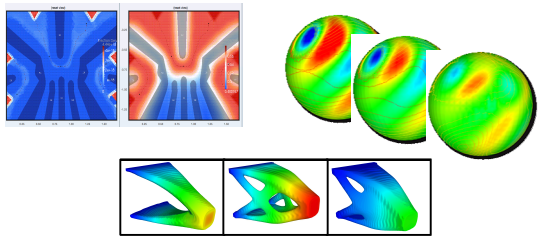
Sandia National Laboratories

Application Impact: Ice Sheets

- Surface flow velocities for Greenland and Antarctic Ice Sheets
- Demonstrates nonlinear solves, linear solves, UQ, adaptivity, and performance portability
- Employs automatic differentiation, discretizations, partitioning, mesh database

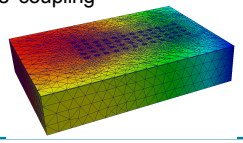


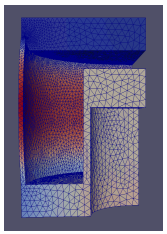
Additional Application Impact



Application Impact: Computational Mechanics

- Largest implicit problem solved in Albany to date: 1.7B degrees of freedom
- Initial capabilities for Schwarz multiscale coupling

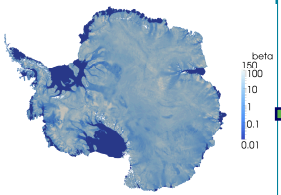




Nonlinear Solvers and Inversion

- Homotopy and Anderson Acceleration in Trilinos::NOX
- The robustness of nonlinear solvers are critical when an application is to be called as a sub-component within a larger application code.
- Uses Automatic Differentiation, Preconditioning, Optimization algorithms from Trilinos

$$\frac{dg}{dp} = \frac{\partial g}{\partial x}^T \frac{\partial f}{\partial x}^{-1} \frac{\partial f}{\partial p} + \frac{\partial g}{\partial p}$$



Application

Albany

DAKOTA

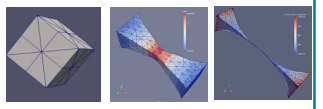
Trilinos

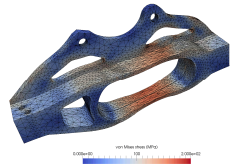
PUMI

Parallel Unstructured Mesh Infrastructure

Mesh Adaptivity

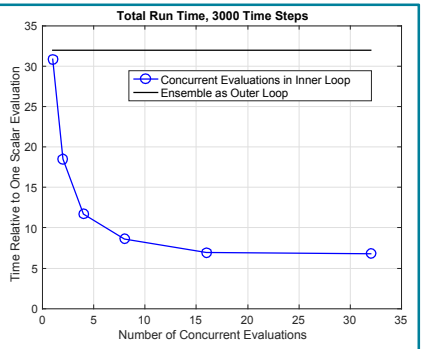
- Mesh adaptation can be essential for efficiency and robustness
- Cube geometry subjected to large deformation (elasticity and J2 plasticity results shown)





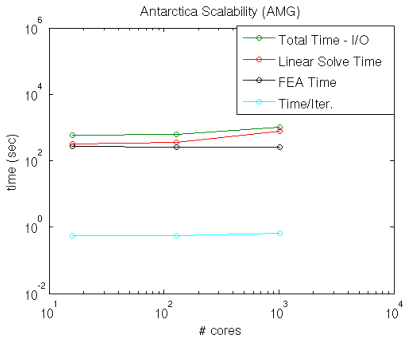
Embedded UQ

- New *Ensemble* data type in Sacado package
- Vectorization of kernels over ensembles
- Contiguous memory access in arrays



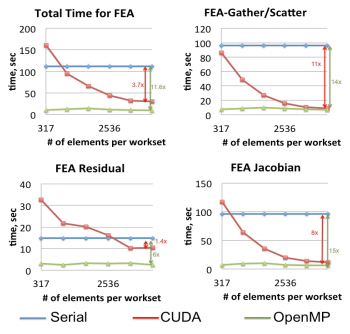
Scalable Linear Algebra

- Scalability of simulations requires effective preconditioning
- Multi-level solves are essential for the largest problems



Performance Portability

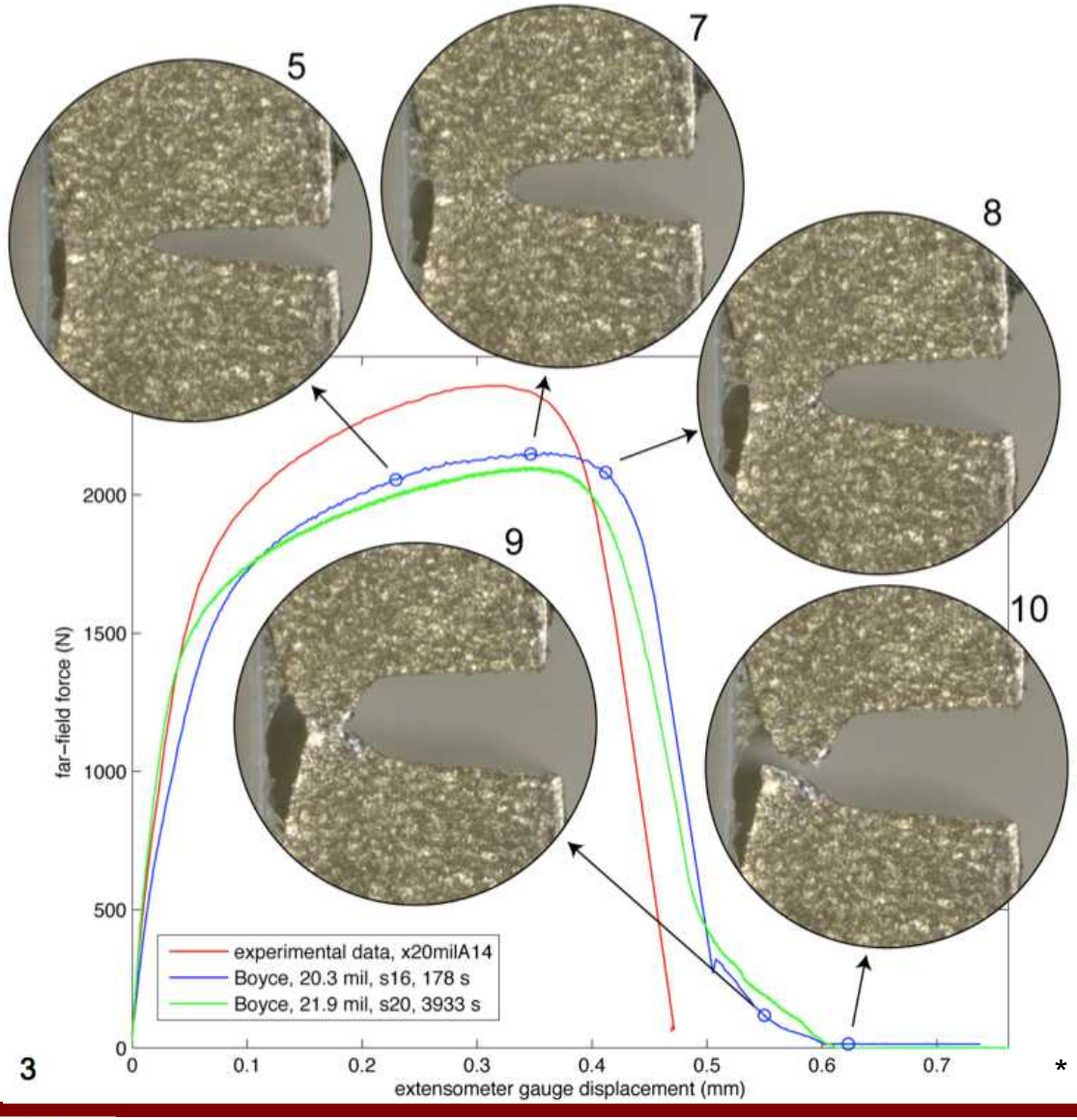
- The Kokkos programming mode supports performance portability of kernels.
- Kokkos' abstraction layer allows code to be tailored for specific devices

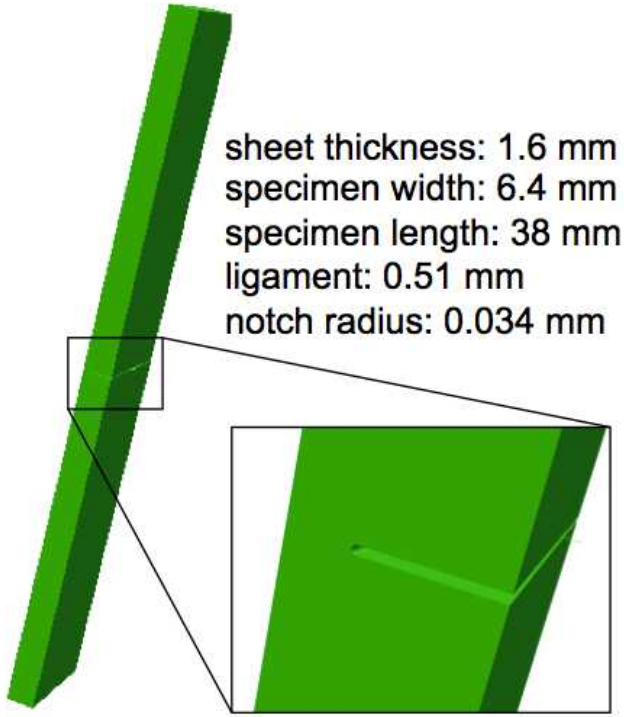


SCOREC

FASTMATH

Weld failure: motivation for the use of mesh adaptation*

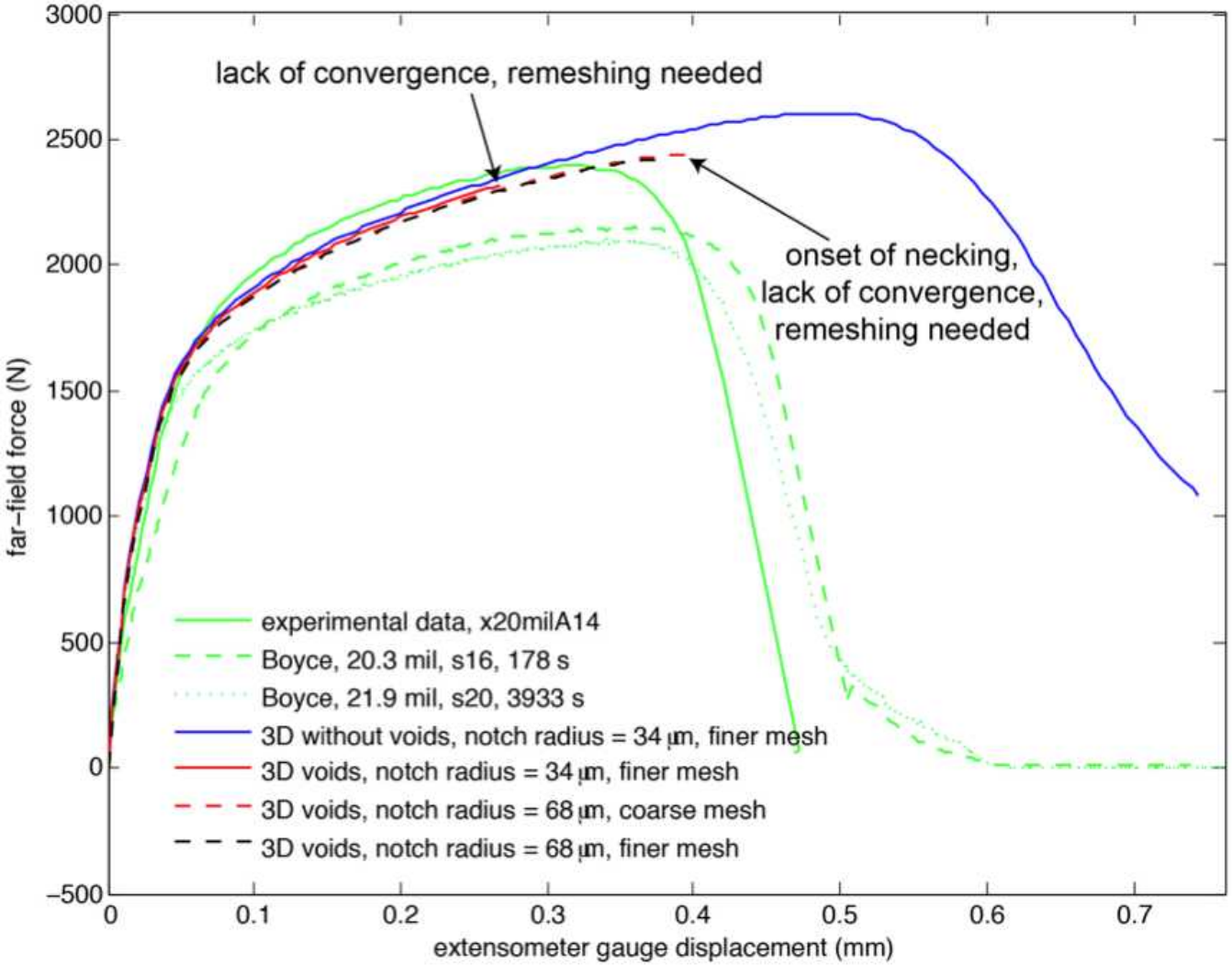




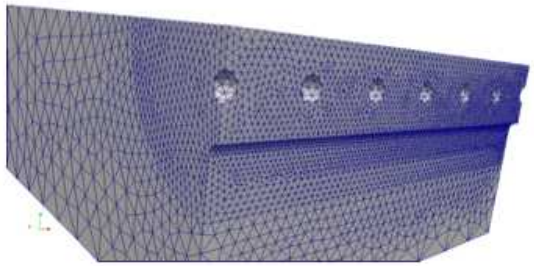
- *Predict the peak load and the rapid drop in the load-bearing capacity of the weld.*

* Jay Foulk and LCM Team

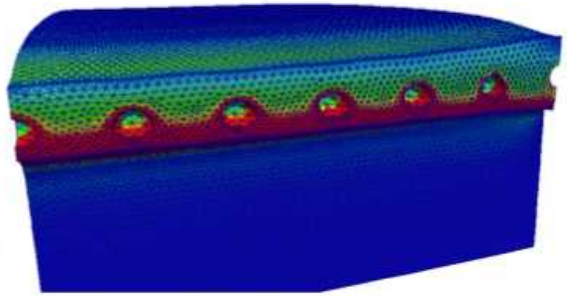
Comparison of the computed load-displacement curve vs. experiment*



notch radius: 68 μm



onset of necking
notch radius = 68 μm
coarse mesh



* Jay Foulk and LCM Team

What adaptation criteria should be used?

Sandia National Laboratories

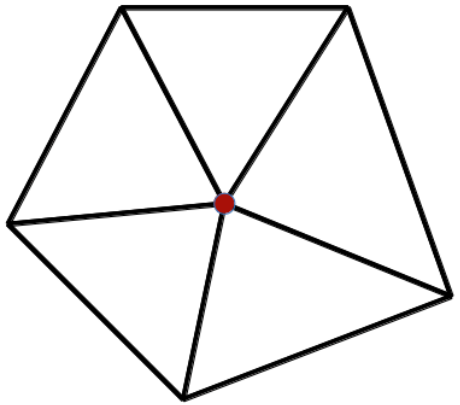
- Depends on the nature of the problem being solved
- Goal: Improve robustness for large deformation problems
  - Observation - the analyst often builds “features” into the initial mesh to resolve important details in the solution
  - Calculate and employ a *mesh size field* to preserve these features through use of adaptation as the domain deforms
- Goal: Manage error for a target number of elements (DOFs)
  - For a given target element count in the simulation, use an *error indicator* to calculate a dynamic *mesh size field* that:
    - refines the mesh in areas of large error
    - coarsens the mesh in areas of small error (or areas of less interest)

Adaptation based on error estimation

Sandia National Laboratories

Given a (p-1) order state variable σ (e.g. Cauchy stress) at integration points

1. Construct an appropriate patch of elements
2. Recover a p-order field σ* via a least squares fit to integration point data σ over the element patch
3. Repeat steps 1 and 2 for all element patches in the mesh
4. Integrate norms of p-order error field of e = σ – σ* over the whole mesh
5. Compute an appropriate mesh size field based on the estimated error



A 2D element patch

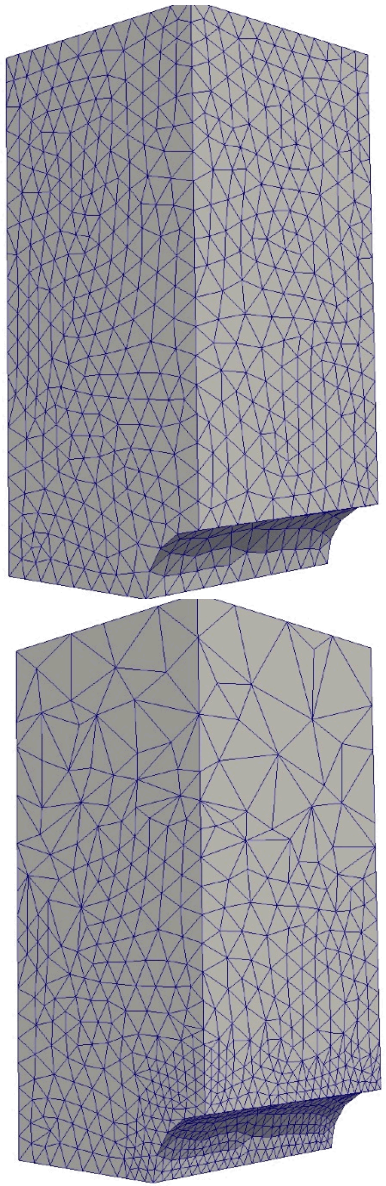
* Brian Granzow, RPI

Mesh size field

- An element or node field that specifies what the local mesh size should be, as a scalar radius (isotropic) or vector (anisotropic)
- Two available approaches
  - calculate size field in template class to concrete **AbstractAdapt** object given solution and mesh fields, or
  - **ElementSizeField** evaluator can calculate size field as an element, QP, or nodal field as a response
- **StateManager** manages the resulting fields
  - **NodalDataBlock** used for cross-workset and interprocessor consistency of nodal data fields
  - **NodalDataBlock** accesses all adaptive discretizations through **AbstractDiscretization** base class

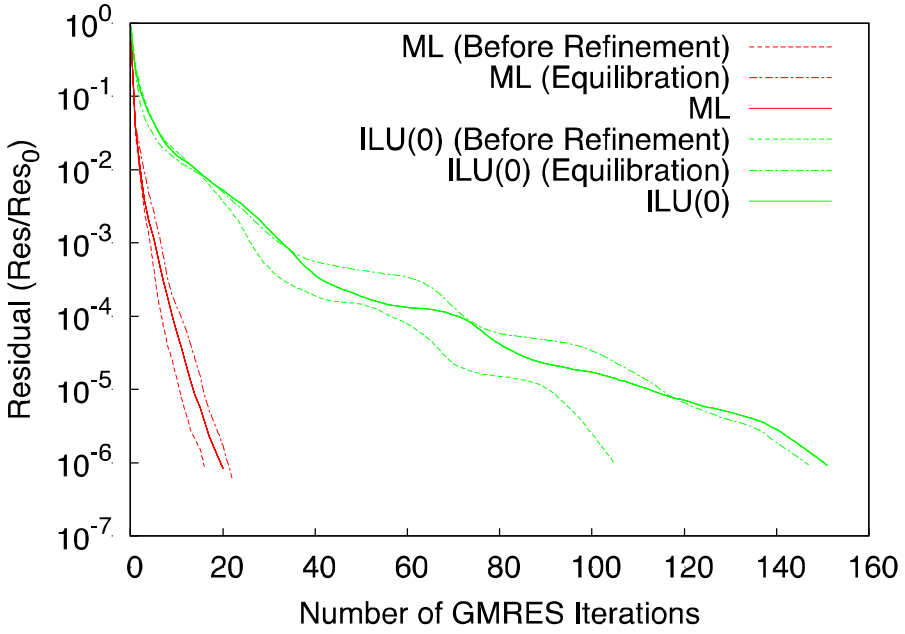
Adaptation process

- PUMI meshadapt modifies the mesh to satisfy the size field
- PUMI then load balances the adapted mesh across the available processors
- Locally, PUMI estimates the nodal solution at the locations where new nodes are added, and passes the modified nodal field data back to Albany
  - **AbstractAdapter** provides virtual *solutionTransfer*() method
- Finally, the concrete version of **AbstractAdapter**::*adaptMesh*() returns control to the LOCA stepper to begin the equilibration step

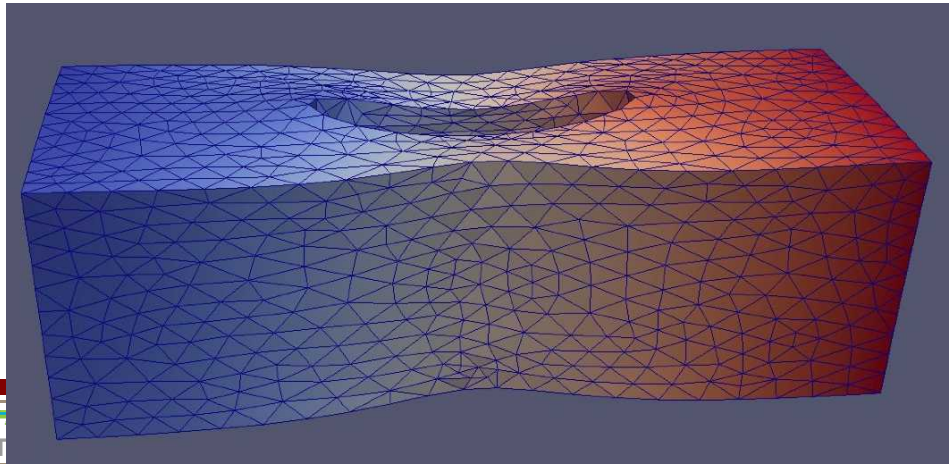


MueLu rigid body modes

- Null space data structures are resized and repopulated prior to LOCA equilibration step, given the new nodal coordinate data from PUMI
- The equilibration step is designed to "adjust" new interpolated values to satisfy equilibrium conditions, without advancing the problem state, to reduce overall error
- LOCA then resumes the stepping process



Bar deformation problem



Solution transfer

- As the mesh adapts, one must "estimate" the values of the nodal fields at the location of new node points, and the values of QP fields at the integration points of new (or modified) elements
  - Some state quantities cannot be transferred using interpolation and must be treated using Lie Algebra Formalism (c.f. Dept. 8343)
- It is usually important to ensure that the estimated field values satisfy the conservation laws being employed (i.e., the transfer method is conservative) without introducing new maxima or minima not present in the original fields
- One may choose to employ an "equilibration" or "relaxation" process to adjust the estimated values to provide equilibrium with the fields on the existing mesh entities
  - LOCA provides an equilibration step after each adaptation operation

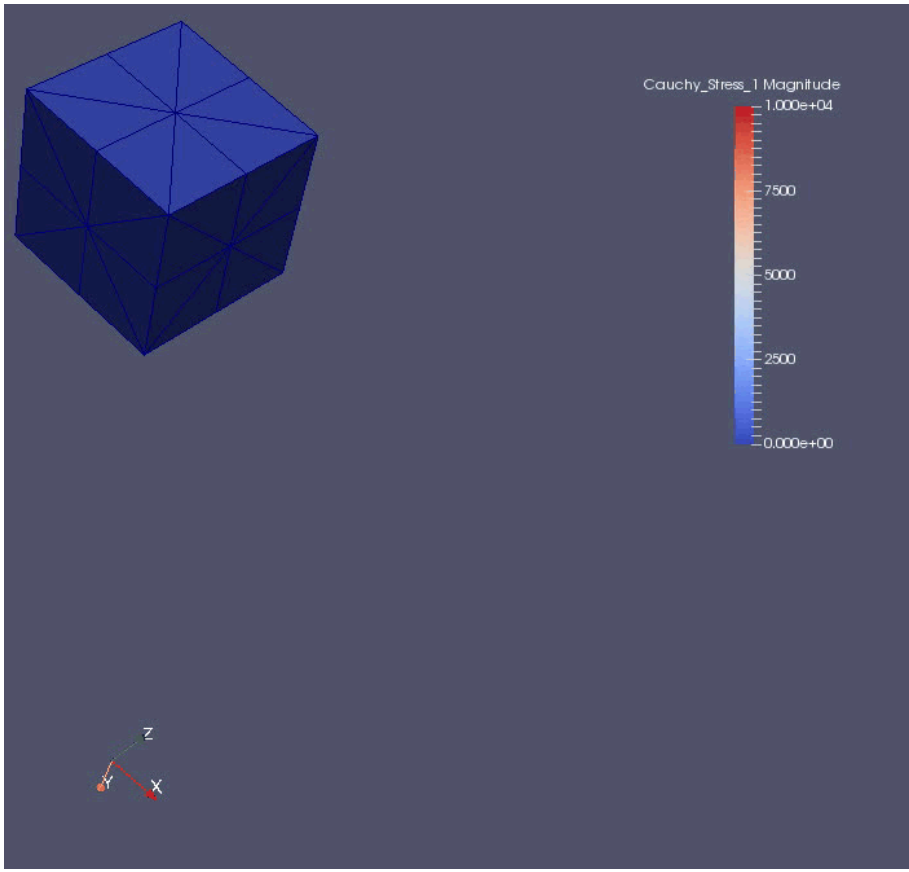
Adaptive meshing workflow

- Generate a *tetrahedral* mesh using the Simmetrix simmodeler tool, typically from a CUBIT .sat file
- Edit the Albany input file to specify that PUMI will serve as the mesh database, and specify parameters to drive the adaptation process and problem load balancing
- Please attend the Albany Adaptive tutorial on Thursday to see the process in full detail

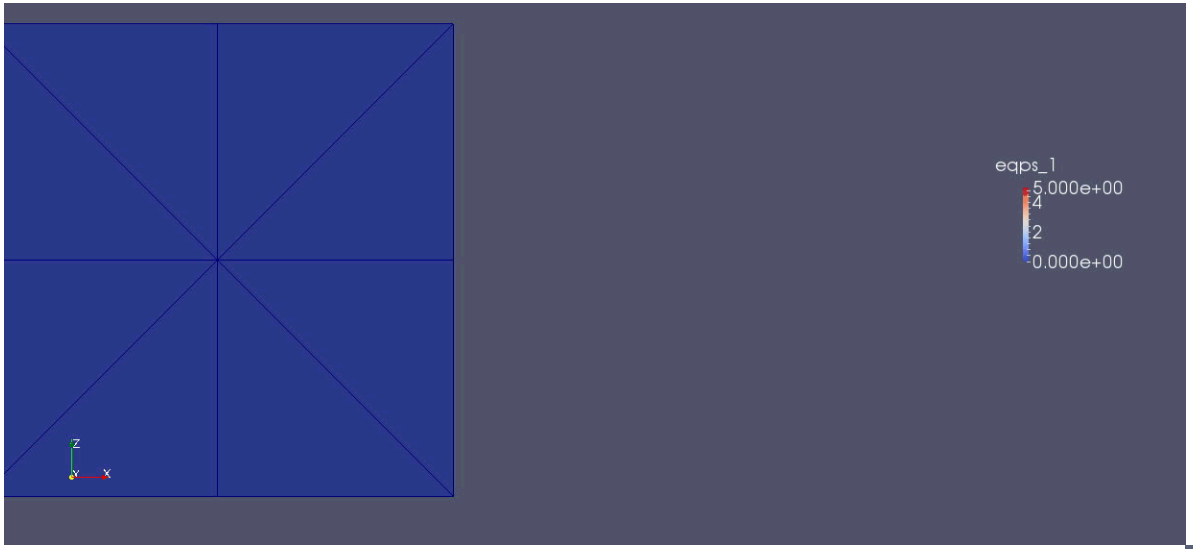
Mesh adaptation for mechanics

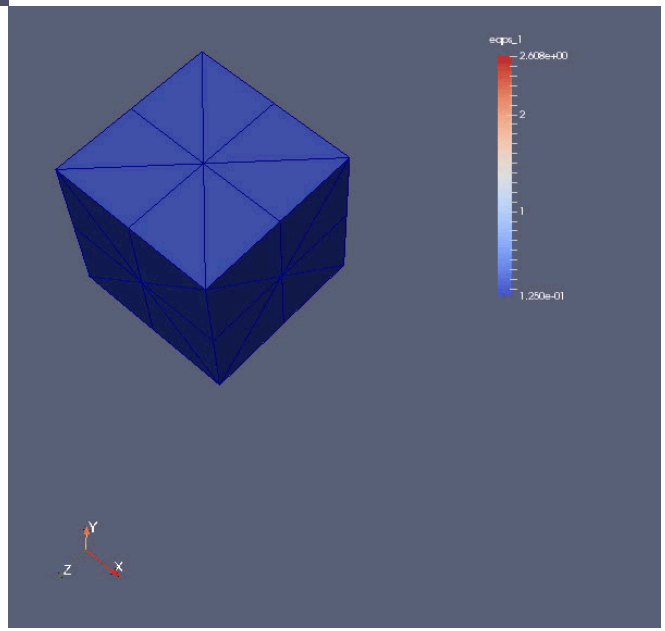
- Goal: Employ evolving mesh topology to maintain element quality in large deformation scenarios
- Challenge: Ensure conservation (mass, momentum, strain energy) as the mesh adapts, maintaining element quality and robustness of the approach during complex deformation states
- Method:
  - Stress model is a function of the deformation gradient tensor **F**.
  - **F** is stored at integration points.
  - In a step, **F** is (a) updated and then (b) transferred to the adapted mesh.
  - Interpolation data transfer is used in the examples to follow.

Elasticity demonstration



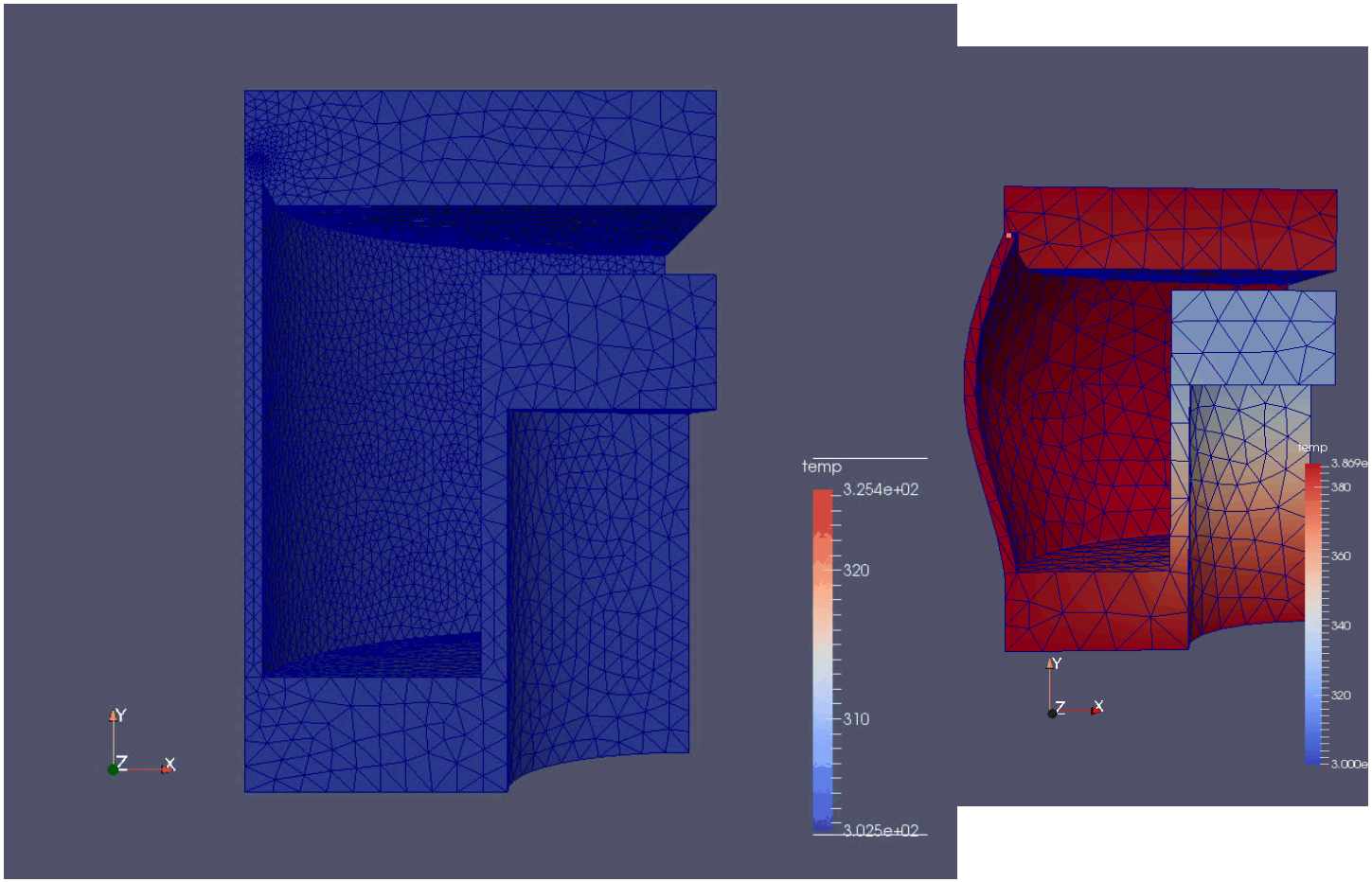
J2 adaptation

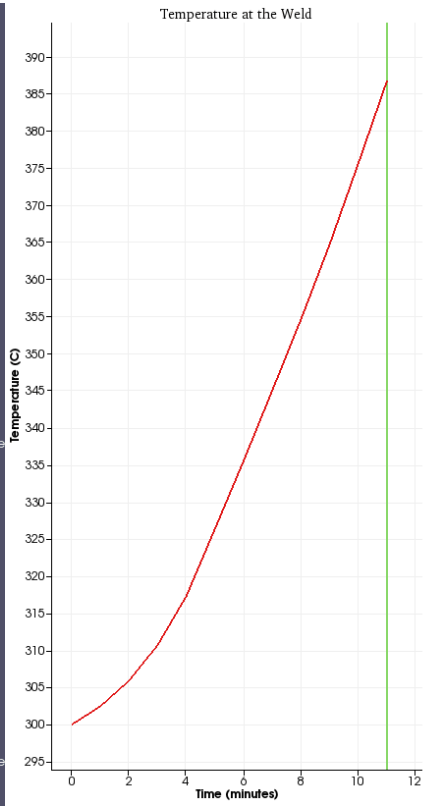




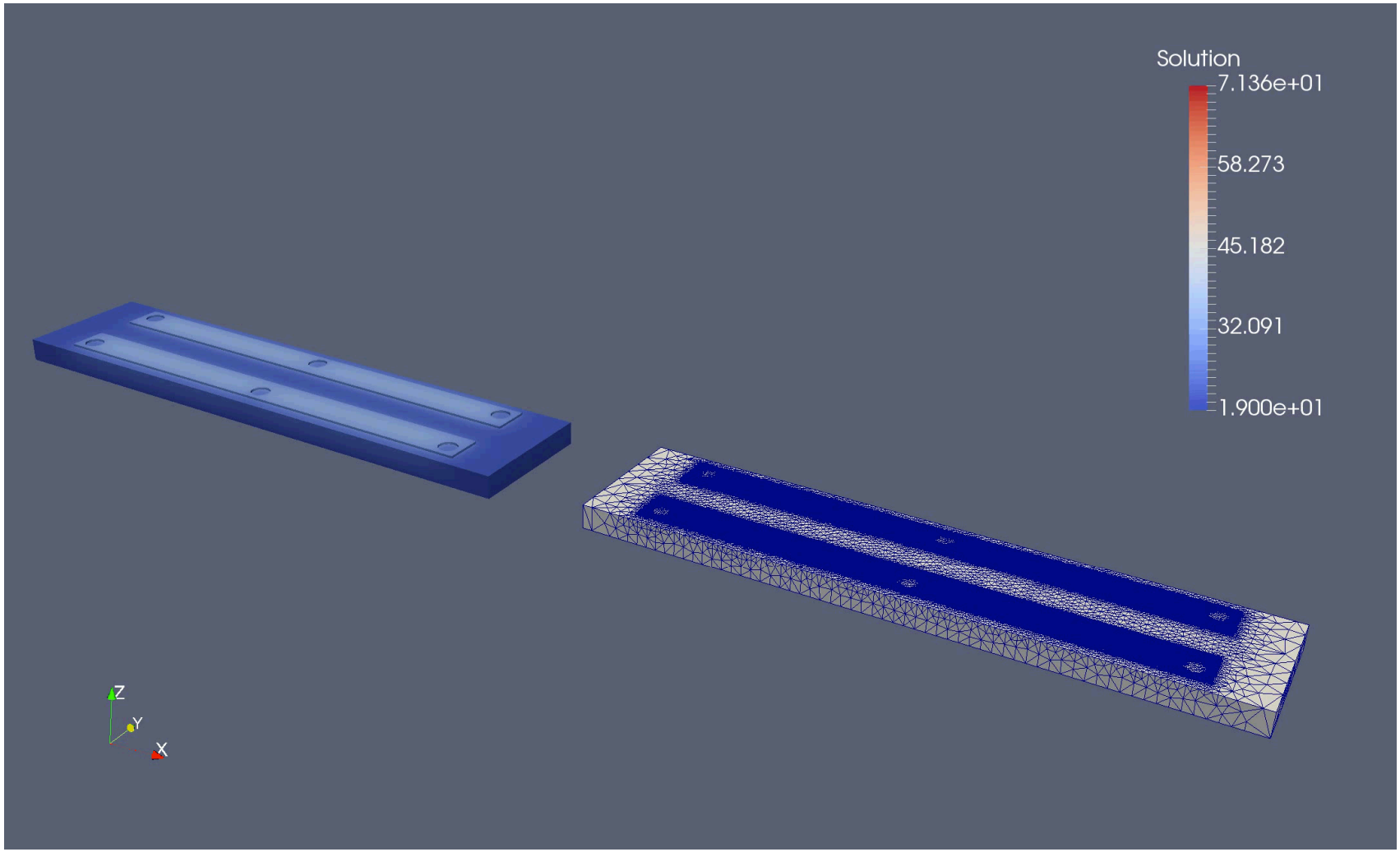
All together now

- Coupled thermomechanics, J2 plasticity, temperature dependent material hardening at the weld





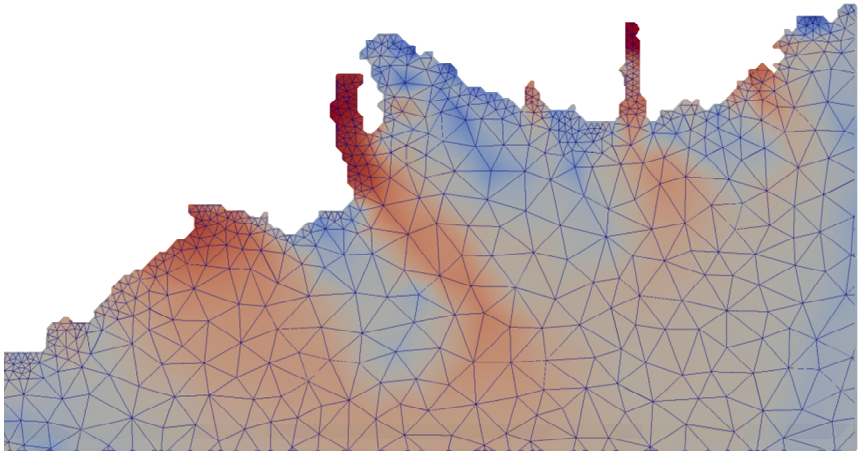
Thermal additive manufacturing*

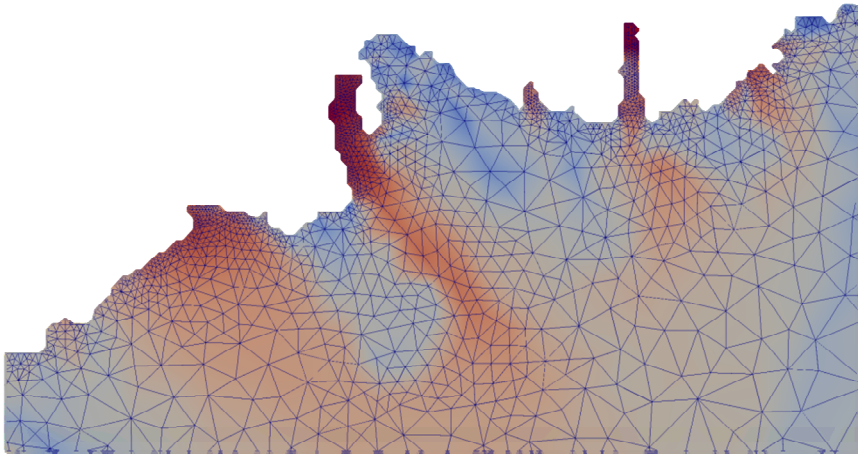


* Mario Juha and SCOREC Team

Felix ice sheet modeling*

- In-memory parallel adaptive loop to analyze ice sheet flow
- Running on 32 cores
- Combining
  - Parallel mesh adaptation
    - Linear prism mesh elements
  - SPR-based error estimation
  - Local solution transfer of state variables
  - Predictive load balancing





* Mauro Perego, Irina Tezaur, Cameron Smith

Summary

Sandia National Laboratories

- **ATPESC adaptation exercises**
  - https://github.com/gahansen/Albany/wiki/PAALS-Tutorial-201x
- **Capabilities:**
  - *Agile Component*-based, massively parallel solution adaptive multiphysics analysis
  - Fully-coupled, in-memory adaptation and solution transfer
  - Parallel mesh infrastructure and services
  - Dynamic load balancing
  - Generalized error estimation drives adaptation
- **Download:**
  - Albany (http://gahansen.github.io/Albany)
  - SCOREC Adaptive Components (https://github.com/SCOREC)
- **Further information:** Mark Shephard [shephard@rpi.edu]
  Glen Hansen [gahanse@sandia.gov]

PUMI Parallel Unstructured Mesh Infrastructure

Albany

ALEGRA SHOCK & MULTIPHYSICS

Adaptation tutorial Thursday

Sandia National Laboratories

Please attend to learn more about the details of adaptation in Albany!