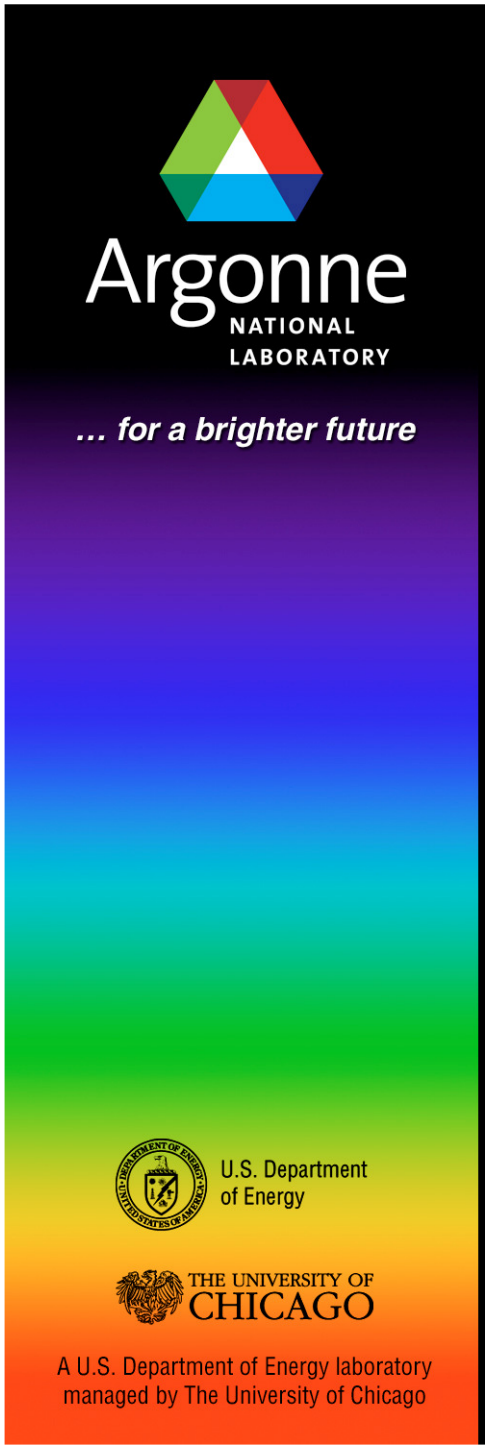


Coupled Multi-Physics Simulation Frameworks for Reactor Simulation: A Bottom-Up Approach

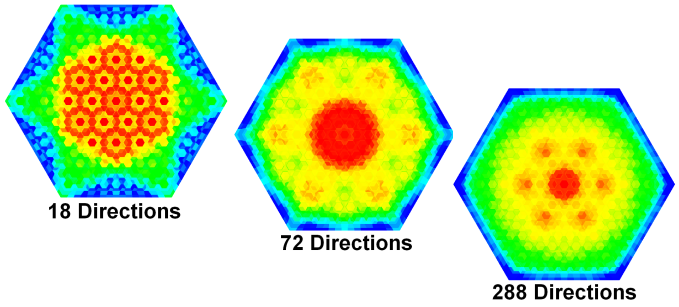
FR09
Kyoto, Japan
December 10, 2009

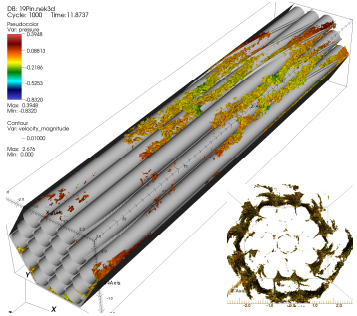
Timothy J. Tautges
Mathematics & Computer Science Division
Argonne National Laboratory

Work performed by ANL for the U.S. DOE in support of the Nuclear Energy Advanced Modeling & Simulation (NEAMS) program

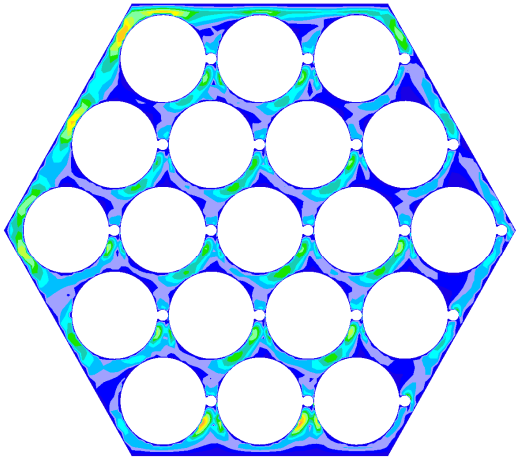
Reactor Simulation is Inherently Multi-Physics, Multi-Model

- Multi-physics
  - Neutronics
  - Thermal/hydraulics
  - Structural mechanics



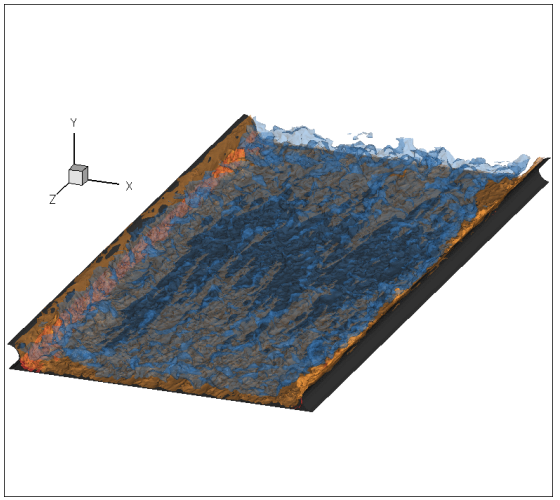


- Multi-model



RANS

LES



DNS

Physics/Models Couple Through Spatial Domain

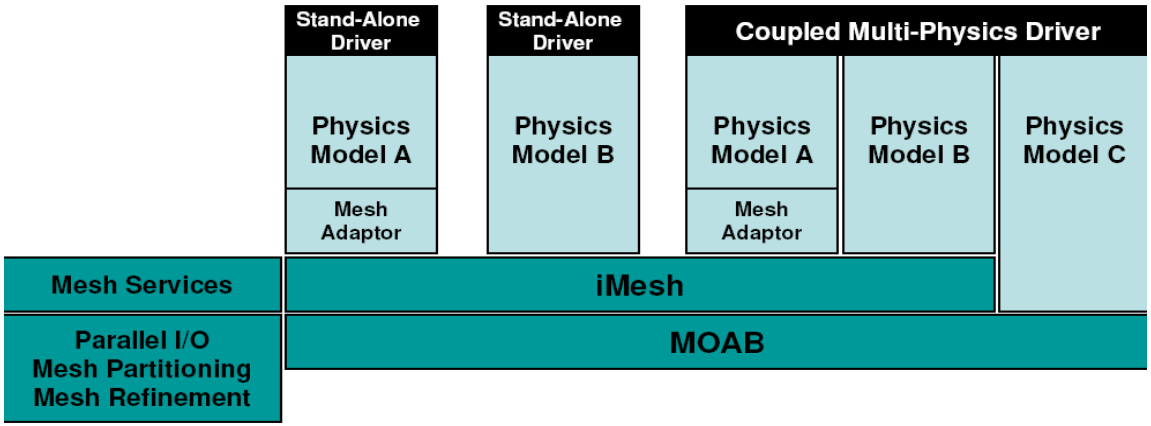
- Spatial model (mesh or geometry/CAD) couples results in both multi-physics and multi-model simulations
- Large-code architecture often organized around handling of the spatial domain (mesh) and fine-grained data on the mesh (fields)
- Geometric model (CAD) closely related
  - Resolution depends on analysis needs
    - *e.g. wire wrap*
  - MPP-enabled resolution should resolve geometric features (where possible & useful?)
- Mesh generation is a bottleneck to the whole process

- *"Bottom-up Framework"* designed with these concepts in mind

Framework Requirements

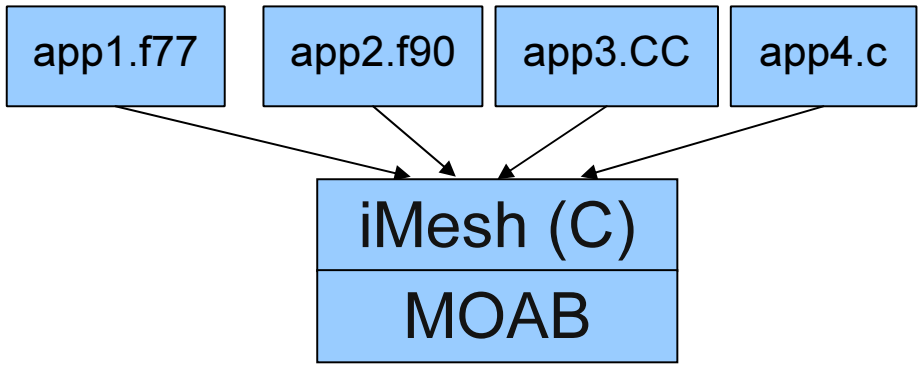
- Act as “data bus” for mesh, data on mesh
- Provide means for accessing common services
  - Parallel decomposition, I/O
  - Mesh generation
  - Solution transfer
- *Simplify* construction of parallel, multi-scale/physics code, not make it more difficult
- Will describe:
  - Mesh interface
  - Services
  - Physics modules already connected to SHARP

The SHARP Framework



The ITAPS Interfaces: iMesh, iGeom, iRel

- ITAPS is designing common functional interfaces to mesh (iMesh), CAD models (iGeom), relations between mesh/geometry (iRel)
  - Simple but powerful interface for accessing mesh, geometry data
  - ANL provides MOAB (iMesh), CGM (iGeom), Lasso (iRel) implementations
- C-based interfaces accessed directly from C, C++, Fortran
- Very simple data model: entity, set, tag, interface
- MOAB:
  - finite element zoo + polygons/polyhedra
  - Memory-efficient mesh storage
  - Parallel mesh access
- CGM
  - CAD geometry access in ACIS, Open.Cascade engines



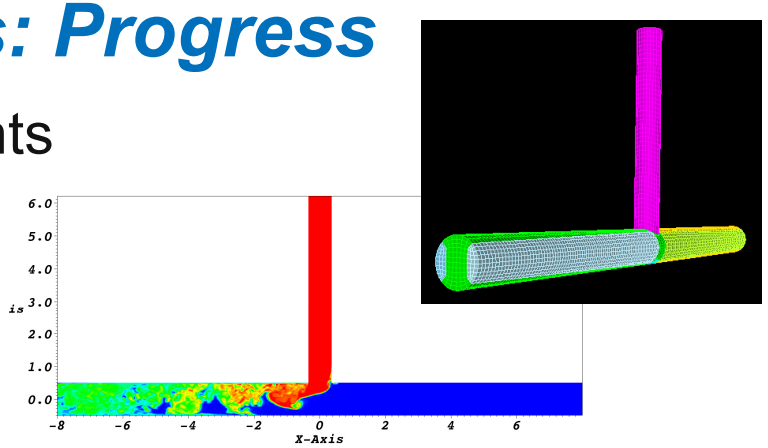
http://trac.mcs.anl.gov/projects/ITAPS/wiki

Connecting Physics Modules in SHARP

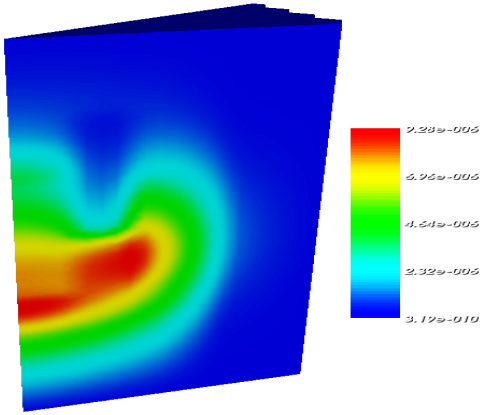
- Value proposition:
  - SHARP provides infrastructure, services that you don't need to (re-) develop yourself (new codes)
  - Allows construction of models, meshes much more detailed than possible before (all)
  - Connecting to SHARP makes it easier to couple to other physics that are already in the framework (all)
- Code modifications
  - Load mesh, access mesh, access metadata, renumber, write solution, save
- Typical simulation workflow
  - Geometry construction, mesh generation, physics input, parallel decomposition, simulation, post-processing

Connecting Physics Modules: Progress

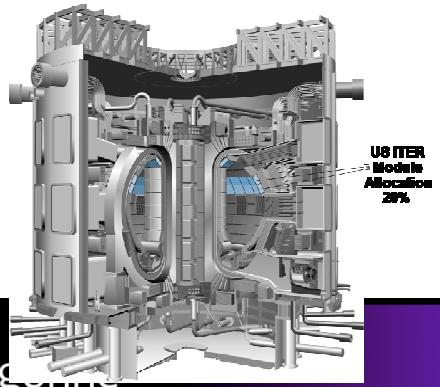
- Nek: LES-based CFD, spectral elements
  - OECD Vettenhall T-Junction benchmark

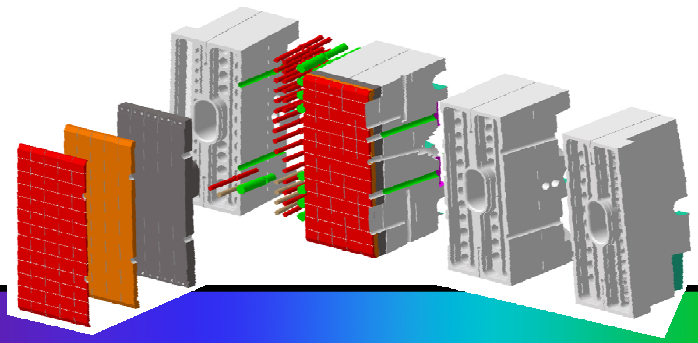


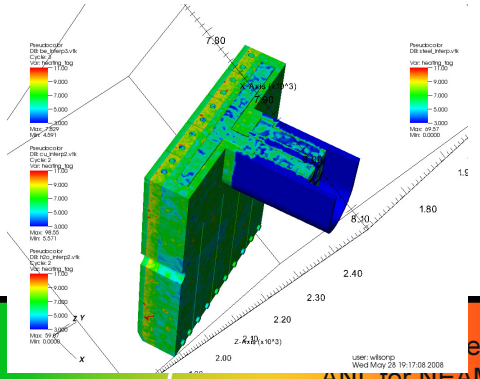
- UNIC: deterministic neutron transport
  - Takeda4 benchmark problem



- MCNP: Monte Carlo neutron transport
  - ITER first wall/shield heating

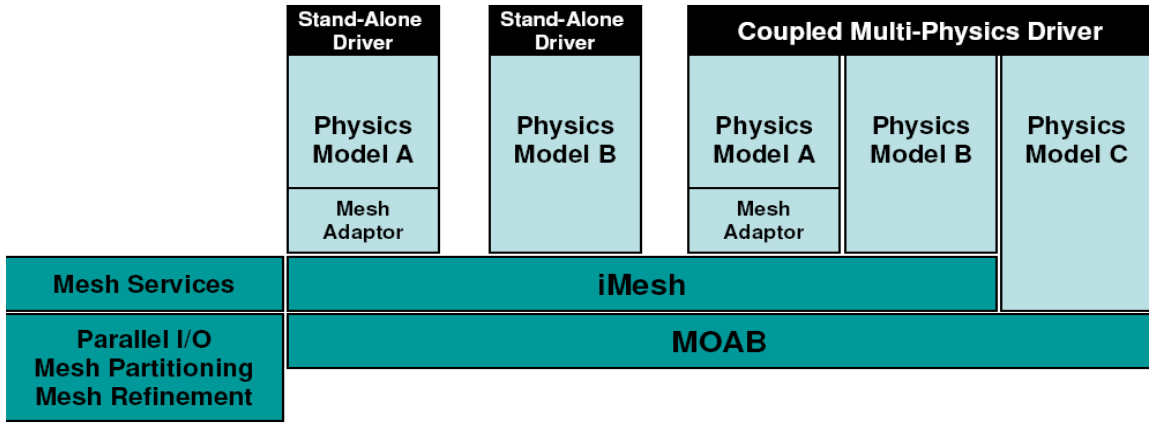






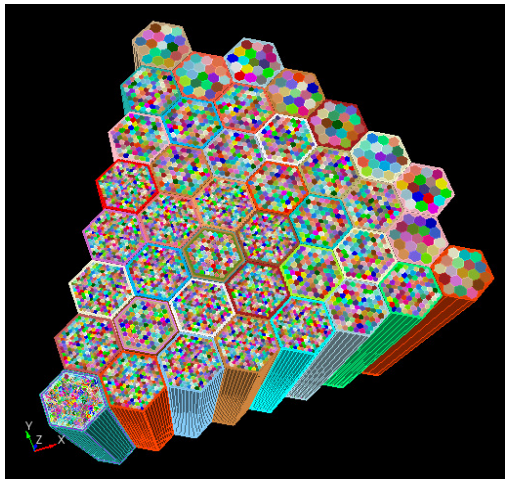
Services

- Part of value proposition is easy access to services
- Examples:
  - Mesh generation
  - Parallel decomposition
  - Solution transfer
  - Visualization



Mesh Generation: Dual Paths

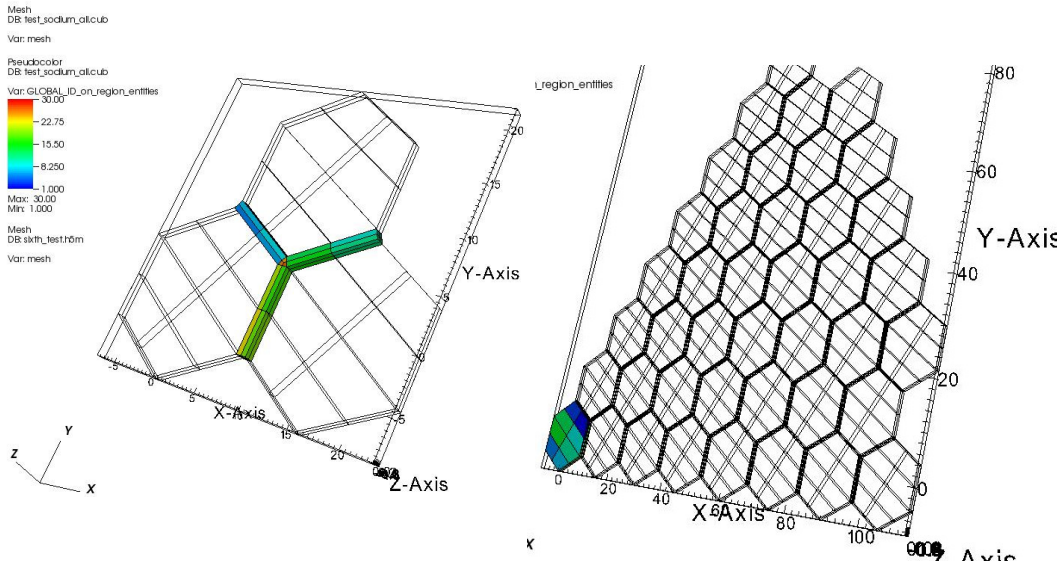
- CUBIT mesh generation toolkit (Sandia Nat'l Labs)
  - Interactive tool



1/6 ABTR core

- 7k volumes (core, ctrl, reflect, shield)
- 43k-5m hex elements

- MeshKit (Argonne Nat'l Lab)
  - Library-based tool

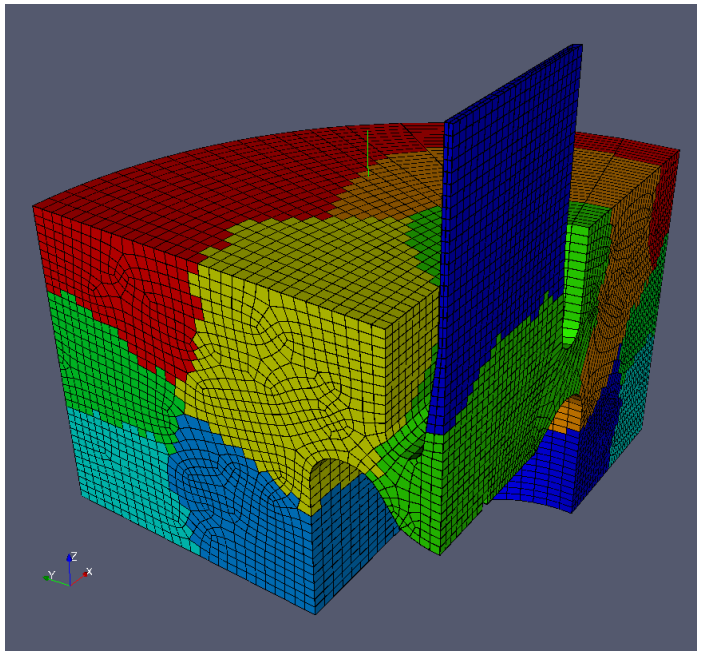


Copy/move/merge

- Copy/move assembly mesh(es)
- Hexagonal or rectangular lattice
- 25x less memory than CUBIT (no geometry, graphics)

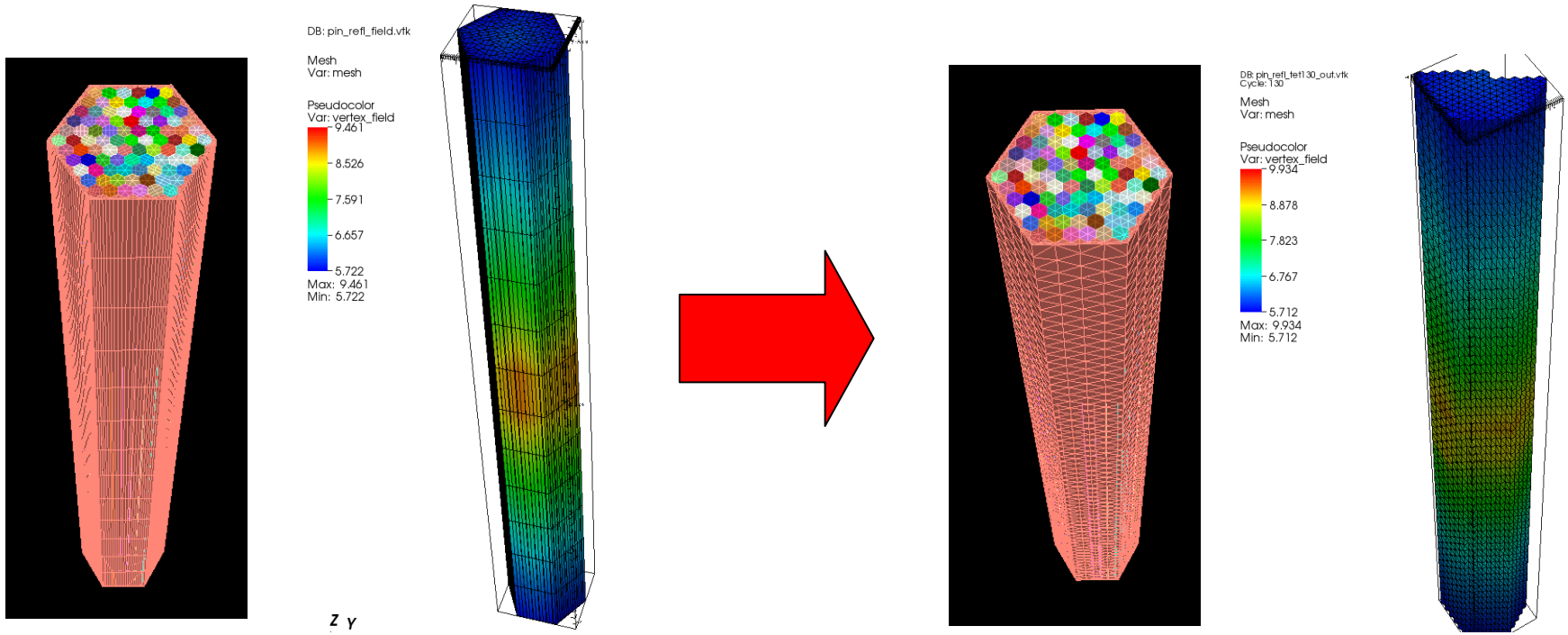
Parallel Decomposition: Zoltan

- Sandia tool for decomposing mesh across processors using a variety of decomposition methods
- Connected to iMesh/MOAB
  - Read/write mesh directly from/to SHARP
  - No loss of meta-data (boundary conditions, materials, etc.)



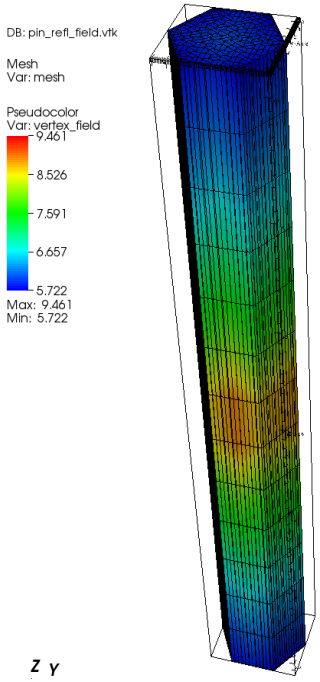
Solution Transfer

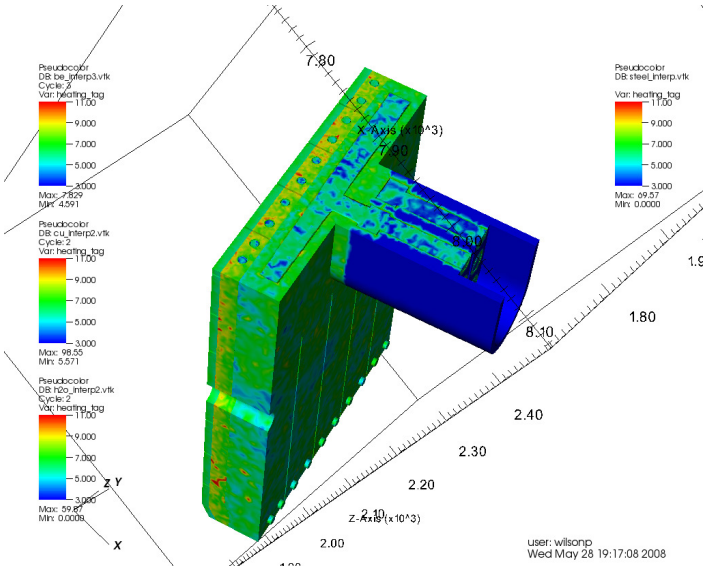
- Solution & mesh stored in iMesh instance on each processor
- Two meshes can be disjoint & distributed independently
- Overall structure is similar whether you're coupling between FD, FEM, SEM

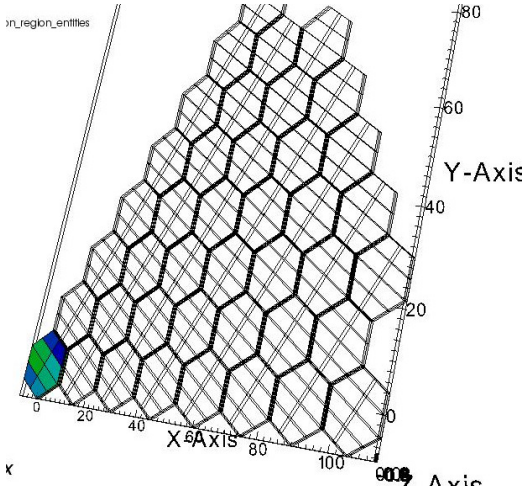


Visualization

- Using LLNL VisIt tool
  - Qt/VTK-based
  - Client/server parallel model
  - Interface to iMesh







Conclusions

- Focus on bottom-up framework(mesh interface & services connected to it) simplifies application development
  - Don't need to re-implement physics
  - Access to services commonly needed
- Basic connection of physics modules done, now need to focus on coupling
- Most of these tools are freely available as Open Source Software