SAND2008-7619P

An Update on the Thermal / Fluids Code Consolidation and SIERRA Framework/Toolkit Effort

Basil Hassan

Sandia National Laboratories

November 12, 2008

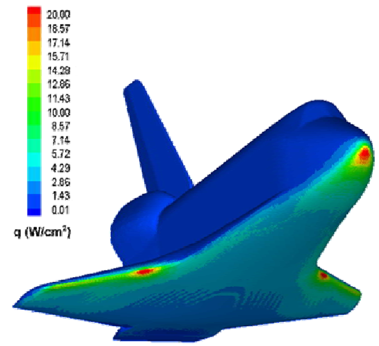
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

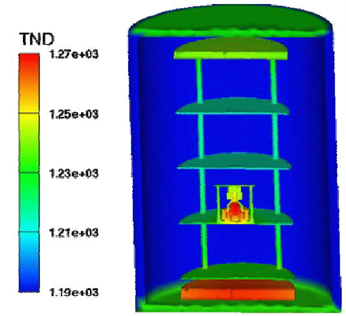
New Scrum Process for T/F Codes Aria, Calore, Premo, Fuego

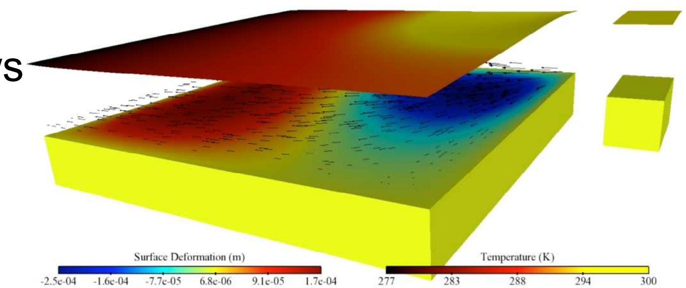
- **Agile code development process has been adopted**
  - Iterative, Incremental, Reflective, Transparent
  - Development occurs in three week Sprint Cycles
  - Scrum team consists of code developers, scrum master and product owner
- **Scrum Master (Alfred Lorber)**
  - Represents the Development Team
  - Facilitates Scrum process, removes impediments
  - Keeps team's progress up-to-date and visible
- **Product Owner (Amalia Black)**
  - Represents the Customer/User Community
  - Collects requirements from Customers
  - Assembles Product Backlog (i.e., development list)
  - Prioritizes work in each Sprint cycle

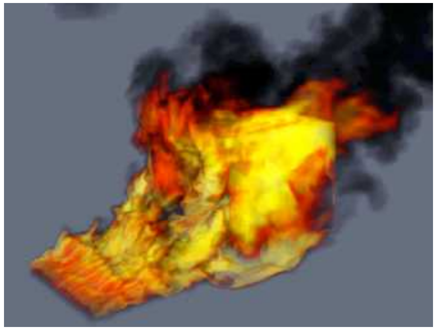
Thermal / Fluids Capabilities

- **Calore** – Heat Transfer, Enclosure Radiation and Chemistry
  - Dynamic enclosures
  - Element birth death
  - Contact
- **Premo** – Compressible Fluid Mechanics
  - Subsonic through hypersonic
  - Laminar and turbulent
  - Unstructured mesh
- **Aria** – Non-Newtonian, Multi-physics, and Free Surface Flows
  - Complex material response
  - Level sets for surface tracking
  - Flexible coupling schemes
- **Fuego** – Low Speed, Variable Density, Chemically Reacting Flows (Fire)
  - Eddy dissipation and mixture fraction reaction models
  - RANS and LES based turbulence models
  - Unstructured Mesh
  - Pressurization models









Sandia National Laboratories

Motivation for Code Consolidation

- **User Benefits:**
  - Tightly-coupled thermal/fluid capability in one code
    - "One" syntax
    - Added robustness and faster convergence due to tight coupling
  - Faster response to user needs
    - Agile programming teams, simplified distribution (see below)
- **Development Benefits:**
  - Capabilities need not be duplicated
    - Though under one Framework, implementation details of adaptivity, error estimation, load balancing, solution control, etc., are duplicated in each code
  - Agile programming teams.
    - Previously: 1 or 2 developers per code
    - Now: core team contributing to all application areas
    - Core team is growing as more developers gain experience
  - Lower cost associated with maintaining a single code
  - Simplified distribution
    - Reduced inter-code dependence make releasing and shipping code easier
  - Increased collaboration between different groups

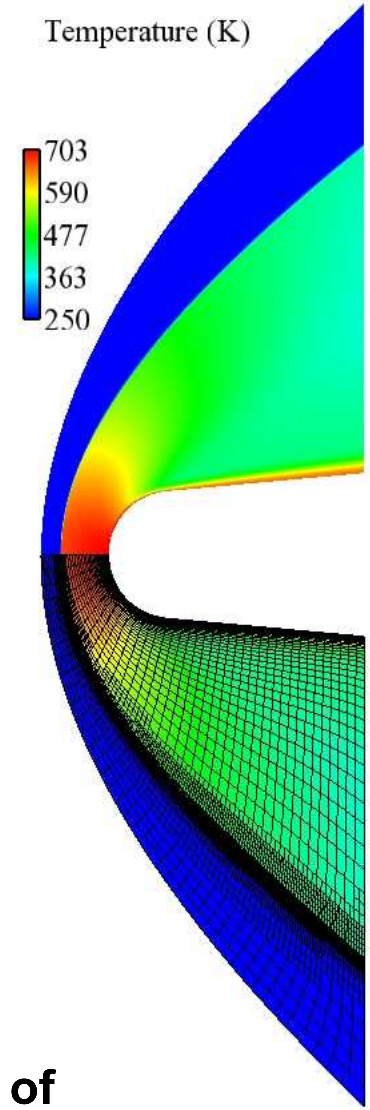
Consolidation Approach

- **Phased Rollout: Calore ⇒ Premo ⇒ Fuego**
- **Not abandoning current development or support of original codes. Development targeted to most needed features.**
- **Each consolidation effort is unique:**
  - Thermal: Galerkin FEM, Enclosure radiation (non-PMR), contact, shells
  - Compressible Flows: Cell-Centered Finite Volume Method & Streamwise Upwind Petrov Galerkin Method (SUPG/FEM)
  - Turbulent, Low Speed, Reacting Flows: Control-Volume FEM capability, segregated equation systems, complex algorithms need simple UI
  - Multiphase Flows: Need to continue with scheduled deliverables; need to maintain cohesion while growing the design
- **Taking the time to do it right**
  - Better Design
  - Verification of implementation
  - Documentation of implementation

Sandia National Laboratories

Consolidation Status

- **Thermal:**
  - Enclosure radiation and generalized contact
  - ChemEQ chemistry for energetic materials
  - Local coordinate systems for anisotropic materials
  - Standard shells implemented; gradient shells in development
  - Banded wavelength enclosure radiation
  - Heavy acceptance testing underway and verification testing is underway
  - **Goal: Release to users early in 2009**
- **Compressible Flow:**
  - First-order Euler CCFVM implementation
  - Second-order CCFVM scheme under development
  - FEM/SUPG implementation
  - Non-equilibrium chemistry (more reactions needed)
  - Acceptance tests underway for high speed aero (CCFVM & SUPG)
  - **Goal: Select final formulation by the end of CY08 and implement initial Aero-Thermal Ablation capability by end of FY09**



Sandia National Laboratories

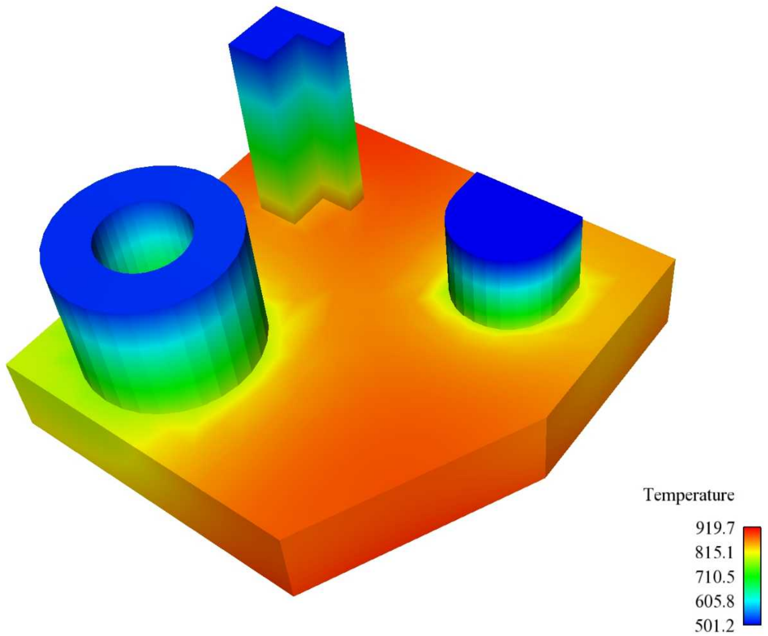
Consolidation Status Continued

- **Low Speed Turbulent Reacting Flows:**
  - 2nd order CVFEM implementation
  - Low Mach, variable density flow solver
  - Turbulence models under development
  - Formal verification using manufactured solutions is underway
  - **Goal: Implement reacting flow capability with segregated solver by end of FY09**
- **Multiphase:**
  - Level set development used to support a Foam Level 2 Milestone
  - Porous flow capability is under development
  - Conformal Decomposition FEM is under development
  - Shallow water formulation is under development to support hydroplaning application
- **Verification testing is an integral part of the code development cycle**
- **Goal is to consolidate major features by end of FY09**

Sandia National Laboratories

Initial Thermal Capability & Status

Capability	Status
Primary physics	
Thermal diffusion	Native to Aria
Thermal advection (limited scope)	Done
Thermal contact (Generalized)	Done
Secondary physics	
Volumetric and nodal heating source	Native to Aria
Non-diffusive chemistry	Done
Boundary conditions	
Convective BC	Native to Aria
Heat flux BC	Native to Aria
Far-field radiation BC	Native to Aria
Dirichlet BC	Native to Aria
User Subroutine BC	Done
Enclosure radiation BC	Done
Bulk fluid element BC	To do
Elements	
Quad and Triangle support	Native to Aria
Hex and Tet support	Native to Aria
Wedge and Pyramid support	To do
Shells support	Standard Done
Bar support	Done
Materials - conductivity, specific heat, density	
Constant	Native to Aria
User function	Native to Aria
User subroutine	Done
Materials - emissivity	
Constant	Native to Aria
User Subroutine	Done
Post processing	
Integrated power for flux BCs	To do
Surface power for arbitrary surfaces	To do
Utility	
Locally scoped material constants	Done
User variables (mesh object types and global)	Done
User query functions	Done
Special output point probes	Native to Encore



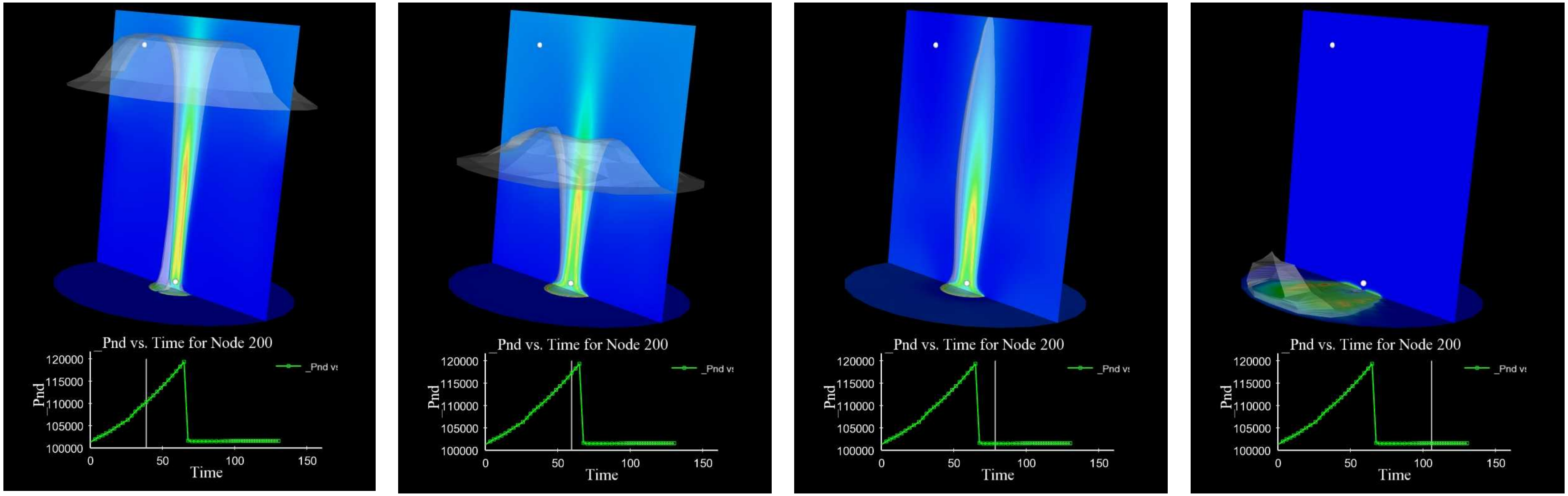
Sandia National Laboratories

5 meter JP-8 Pressurizing Fire
Jaime Severn Visit in August

- **Case 1: Fire continues to pressurize vessel**
- **Case 2 and 3: Fire pressurizes vessel until a pressure threshold is met at which point the valve either remains open or closes**

pressure = 101.25e3
pressure release threshold = 120.00e3 and model total failure

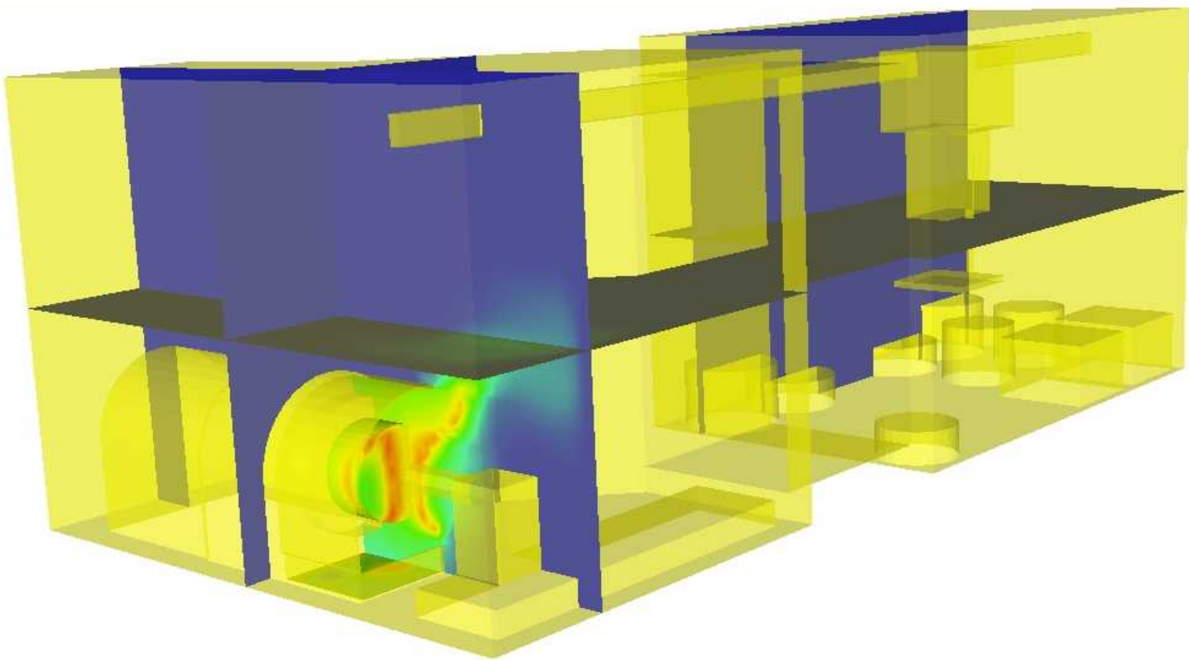
pressure = 101.25e3
pressure release threshold = 120.00e3



Time sequence for modeling total failure of pressure valve

Pressurizing
Glove Box with Evaporative Pool

- Dr. Jamie Severn's provided glove-box mesh (~180K elements)
- Input file created with coupling to PMR and evaporation pool fire
- Simulation run without issue, input files provided



FY'08 Objectives for Toolkit Development (Parallel Mesh Data Management)

- Initial conceptual model for mesh module in Sierra Toolkit
  - Document: high-level description of capabilities & requirements
- API for mesh module
  - Header files and documentation
- Prototype implementation(s) for mesh module
- Performance testing of mesh module prototype(s)
  - Definition of performance test, collection/comparison of timing data
- Algorithm-support infrastructure: mesh traversal, application of element-routines to heterogeneous mesh, etc.
  - API documentation, prototype implementation

Sandia National Laboratories

Initial conceptual model for mesh module in Sierra Toolkit

- “Domain model” document describes the high-level concepts, capabilities and requirements for the mesh module.
  - ‘living’ document which will continue to change in response to the needs of application developers (i.e. the users of the mesh module).

- Important distinction: here the term “mesh module” refers to the in-memory storage of mesh data structures for use by the simulation code. *Not* the on-disk mesh database produced by mesh-generators.

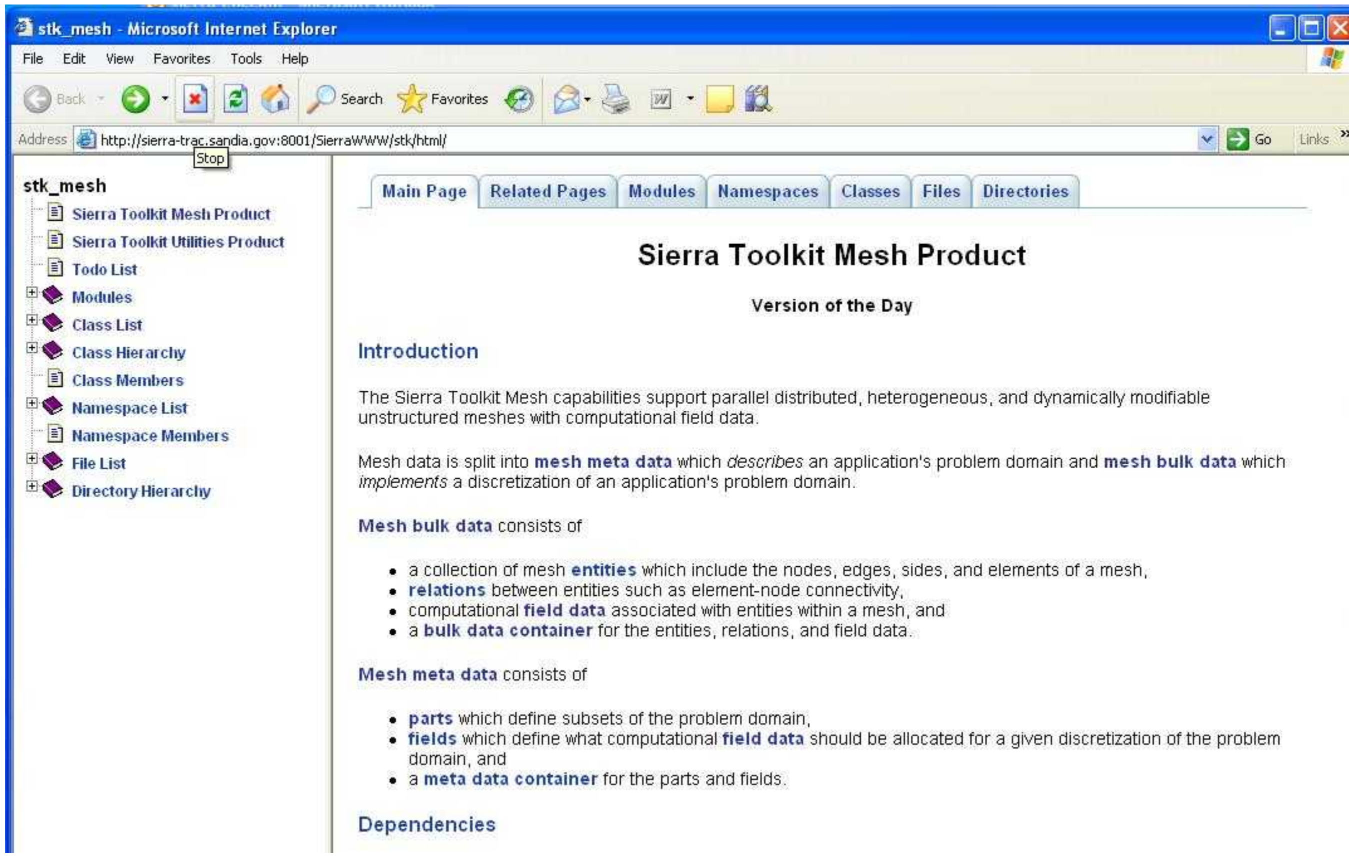
- Significant requirements:
  - Heterogeneous mesh and field data
    - Mixture of element topologies, ability to define fields on subsets of the mesh, and on subsets of the nodes on each element.
  - Dynamic mesh modifications
    - To support element death, parallel load balancing, H-adaptivity
  - Don’t allow one feature to impose overhead on users of other features.

Sandia National Laboratories

API (and documentation) for mesh module

- Documentation for the mesh module includes the conceptual overview document, as well as doxygen-generated documentation which can be delivered as an HTML package, or as PDF or Latex.

- Screen-shot of HTML



Sandia National Laboratories

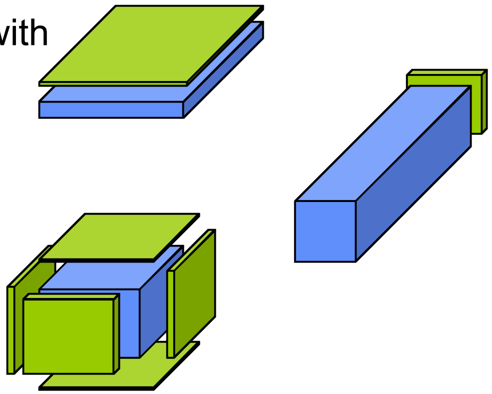
Prototype implementation(s) for mesh module

- We developed and experimented with three mesh implementations:
  - Parallel Heterogeneous Dynamic Mesh (phdMesh)
    Had been initially developed under LDRD funding, had many of our needed capabilities already in place.
  - Really Simple Mesh (rsMesh)
    Developed from scratch, attempted to replicate capabilities with a simpler API and implementation than phdmesh.
  - Array-Based Mesh (abMesh)
    Developed by a third party (another Sandia group), uses simple array data structures and was hoped to provide superior performance.

Performance testing of mesh module prototype(s)

- Performance refers to the cost of operations such as traversing the mesh (visiting the data for all nodes or all elements, etc) and delivering data to element-routines or other computational algorithms.

- Developed two performance tests:
  1. Simple element-loop and node-loop calculations on a large mesh in 3 configurations:
     - Mesh 1: flat square plate of 1 million hex-8 elements with 1 million quad-shells on one surface.
     - Mesh 2: long beam of 50x50x500 hex-8's with 2500 quad-shells on one end
     - Mesh 3: cube of 100x100x100 hex-8's with 10,000 quad-shells on each of the 6 sides



  2. More realistic calculation (element internal force) on arbitrary mesh configuration (mesh read from file).

- Initial tests showed all mesh prototypes were competitive for the loop and traversal calculations, but with some differences in mesh-creation times, etc. (The internal-force performance testing is ongoing.)

- We decided to proceed with a modified version of phdMesh, merging in aspects of the rsMesh API, and calling the result "stk_mesh" (Sierra ToolKit Mesh)

Sandia National Laboratories

Consolidated development efforts to a single mesh implementation

- We decided to proceed with a modified version of phdMesh, merging in aspects of the rsMesh API, and call the result "stk_mesh" (Sierra ToolKit Mesh)

- Performance testing showed that phdMesh was competitive for the traversal and loop calculations, may need some optimization of mesh-creation phase.

- abMesh was attractive due to its simplicity, but lacked support for important capabilities (adding support for heterogeneity and dynamic mesh modifications would have eliminated the simplicity).

- rsMesh had some attractive API features, which are being merged into stk_mesh.

Sandia National Laboratories

Algorithm-support infrastructure

- Refers to functionality that assists the application developer in performing tasks such as mapping element-routines to different element-blocks of a heterogeneous mesh, etc.
- This work is still in progress.
- Some functionality exists in the mesh module itself, and we are still deciding which other modules are appropriate to provide more of this support infrastructure.
- We are proceeding to flesh out the overall “toolkit” in response to user-provided use-cases and feedback.