Executing Scalable Mesh-based Simulation Workflows on Exascale Computers

Jeff Banks, Lori Diachin, Bill Henshaw, LLNL
Mark Shephard, Onkar Sahni, RPI
Tim Tautges, ANL
Most discussion and effort for PDE-based simulation has focused on the scalability of the solvers, both explicit and implicit.

*A scalable solver is of limited use if it takes months to generate the meshes and there is no means to control solution reliability.*

Full execution of the simulation workflows requires:
- Simulation set up/control
- Complex Geometries
- Mesh Generation
- Adaptation/Regeneration

Previous discussion has focused on:
- General issues of exascale computing (e.g., concurrency, fault tolerance)
- Application advances enabled by exascale computing
- Some discussion of mathematics research, particularly solver issues
High-order problem specification is important to many DOE application workflows

- Many DOE applications are modeled using complex, evolving geometrical domains
  - Accelerators
  - Fusion tokamaks
  - Energy systems
  - Nuclear weapons

- Robust, adaptive, high-order modeling of these applications requires
  - Robust and distributed geometry representations
  - Automated geometry coarsening
  - Inline mesh generation and optimization for deforming and evolving geometries
  - Automatics higher-order mesh generation from CAD for unstructured and overset meshes
  - Dynamic load balancing and partitioning for multi-domain solvers and mesh generation

*All operations must be done on the same extreme-scale computers using in-memory linkage of geometry, meshing, simulation, and adaptive control.*
DOE mathematics research investments in this area have resulted in significant advances

- High-order discretizations and solvers with massive numbers of elements running on 100’s of thousands of cores
- Scalable, parallel adaptive mesh refinement for both structured and unstructured grids
- Scalable, optimization-based mesh quality improvement
- Parallel partitioning and predictive load balancing
- SW infrastructure to support current generation HPC platforms
- In-line control flow of adaptive simulations using component software
Mesh and geometry management are still a bottleneck for complex applications on HPC platforms

- **Mesh generation & adaptation on complex geometries must interact with high-level, complete domain representations**
  - Key issue:
    - CAD representations not properly managed
    - CAD models not available on HPC compute nodes
  - Exemplar: 6 months to resolve issues with imprinting and merging in CAD models for ITER geometry

- **I/O for problem set up**
  - Meshes are typically not generated in line with the simulation
  - Spatially varying input parameters, e.g. material properties

- **Fast, efficient load balancing**
  - Critical to maximize application performance at all solution stages
  - Multiple and sometimes conflicting objective functions
  - Migration strategies must minimize data movement

*We must increase automation in pre-processing methods, mesh generation, and adaptive control techniques to ensure reliability of the execution of simulation workflows*
Many areas still require significant improvement for exascale use

- **Geometric information management**
  - How do we improve the robustness in handling evolving geometries?
  - How do we effectively combine multiple sources of geometry information (e.g., multiple CAD, analysis models (meshes), an image data)?
  - How do we approximate/indicate errors introduced by geometric approximations in high-order simulations?

- **Parallel mesh generation and adaptation**
  - How do we distribute mesh and geometry in a consistent manner?
  - Improved techniques for load balance and better algorithms for parallel mesh generation are needed; particularly at massive scale.
  - High-order mesh quality adaptation and improvement algorithms?

- **Partitioning and load balancing**
  - Can we develop multi-objective partitioning schemes that balance the needs of different parts of the solution process?
  - Are there faster, more scalable partitioning techniques for in-line use?
Simulations enabled by exascale computing will require additional mathematics research in these areas

- How can we ‘coarsen’ or reduce geometric representations and input data for adaptive simulations?
  - When is reduced information adequate?
  - What are the accuracy issues?
  - How do you approach this in parallel?
- New methods are needed for mesh generation/pre-processing/partitioning/interaction to meet the needs of complex multi-physics applications
  - How do you partition and load balance multi-domain problems?
  - How do you maximize memory bandwidth performance and minimize communication costs?
- How can one leverage mesh and geometry information to achieve higher efficiency for ensemble simulations for UQ, design studies, shape optimization performed on exascale computers?
  - Which information can/should we leverage?
  - What optimizations can be made, e.g., memory compression?