INL/EXT-13-29814

MOOSE Enhancements Towards Delivery of an Integrated Framework

Derek Gaston
Cody Permann
David Andrs
John Peterson
Andrew Slaughter

July 2013

Idaho National Laboratory

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-13-29814

MOOSE Enhancements Towards Delivery of an Integrated Framework

Derek Gaston
Cody Permann
David Andrs
John Peterson
Andrew Slaughter

July 2013

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

CONTENTS

1. Introduction

1.1 MOOSE: The Multiphysics Object Oriented Simulation Environment

Started in 2008, the Mulitphysics Object Oriented Simulation Environment (MOOSE) project delivers a comprehensive framework for development of parallel, multiphysics simulation tools. MOOSE provides an object-oriented interface that allows scientists and engineers to quickly and efficiently describe all aspects of a physical system relevant to numerical modeling. The MOOSE software development strategy has been quite successful: over thirty different MOOSE-based applications are under development both internally and outside the DOE complex, and over forty different external institutions have licensed MOOSE. Several NEAMS products have been built using MOOSE, including BISON (Williamson et. al., 2012), MARMOT (Tonks et. al., 2012), RELAP-7 (Zou et. al., 2013) and RAVEN (Rabiti, 2013).

1.2 MOAB: A Mesh-Oriented datABase

The MOAB mesh database allows for arbitrary fields of data to be stored spatially across a mesh data structure. The mesh data structure is made up of interconnected "elements" and "nodes" that, together, form the computational "domain" for a simulation. MOAB provides efficient access to these entities and the data stored on them for use by simulation tools such as finite-element analysis (Tautges et. al., 2004). In addition, MOAB provides parallel utilities for reading and writing meshes and data.

1.3 DTK: Data Transfer Kit

DTK (Slattery, 2013) is a standalone data transfer engine for moving solution fields between multiple simultaneously executing simulation tools. Each simulation tool runs independently and utilizes only a subset of the available processors. DTK is used to broker data between these tools via a "rendezvous" mesh for efficient parallel lookup. To achieve this, each tool describes the mesh entities making up its computational domain to DTK. DTK is then able to build a rendezvous mesh and query fields within each application at any spatial point, thereby creating a mapping for a given field variable into another application.

2. New MOOSE Infrastructure

Recently, MOOSE has gained two new features that NEAMS has been able to capitalize on in order to move towards an integrated framework: MultiApps and Transfers. The MultiApp system within MOOSE allows for multiple (possibly thousands) of MOOSE-based applications to be executed simultaneously in parallel. These MOOSE-based applications might represent thousands of rods within an operating reactor, or thousands of microstructure simulations in a multiscale analysis. Using MultiApps, a MOOSE-based simulation can contain an arbitrary number of hierarchical solves that represent each of the individual physics present in a calculation.

The Transfer system within MOOSE allows for moving data between MultiApps. There are three main types of data movement within the Transfer system:

a. Field Mapping

b. Spatially Varying Postprocessed Data

c. Scalar Data

Several Transfer objects have been implemented within MOOSE to allow scientists and engineers to easily piece together complex multiscale, multiphysics applications. For instance, several "Field Mapping" transfers have been developed: L2, Interpolation, Evaluation and several others. A "Field

Mapping” Transfer allows the data in one domain to be efficiently mapped (in parallel) to another domain.

3. MOOSE / MOAB Coupling

3.1 MoabTransfer

Utilizing the Transfer system within MOOSE a MoabTransfer object was developed to move data back and forth between MOOSE and MOAB based applications and coupling frameworks. The MoabTransfer object utilizes DTK to efficiently map fields in parallel between MOOSE-based applications and MOAB datastructures.

3.2 Two-dimensional Mapping

Figure 1 shows an initial calculation utilizing the MoabTransfer object, where a heat generation rate field represented within MOAB datastructures (read from a VTK file in this case—although it could easily be another MOAB-based application or coupling framework) is transferred to a MOOSE-based application. That heat generation rate is then utilized in an axisymmetric (cylindrical) heat conduction calculation. The temperature field from this solve is then mapped back into MOAB datastructures. This surrogate calculation is representative of the cycle that would be necessary for taking a MOOSE-based fuels performance simulation tool such as BISON and linking it with a neutronics application either based on MOAB or communicating with MOAB through a coupling interface.

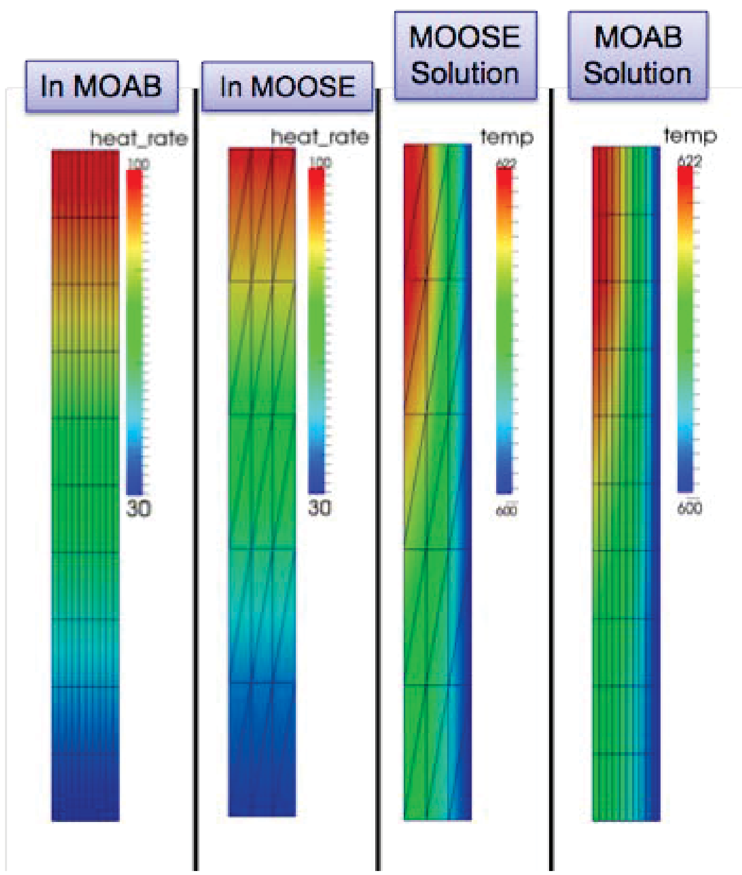

Figure 1: Heat source from MOAB is mapped to MOOSE and an axisymmetric heat conduction calculation is performed and temperature mapped back to MOAB.

3.3 Three Dimensional Mapping

Figure 2 shows a three-dimensional MOAB mesh and a three-dimensional heat generation rate while Figure 3 depicts how that heat generation rate is mapped into three dimensional cylinders, which are representative of three-dimensional fuel rods). The MultiApp system within MOOSE was utilized to conduct three-dimensional heat conduction calculations for each cylinder simultaneously in parallel. Instead of directly mapping the temperature back to the MOAB mesh as in the previous example, a MOOSE "Postprocessor" is utilized to compute the average temperature in the cylinders in axial slices and transfer that data back to the MOAB mesh. This, again, is representative of the type of information transfer that would be necessary in a coupled neutronics and fuels performance calculation.

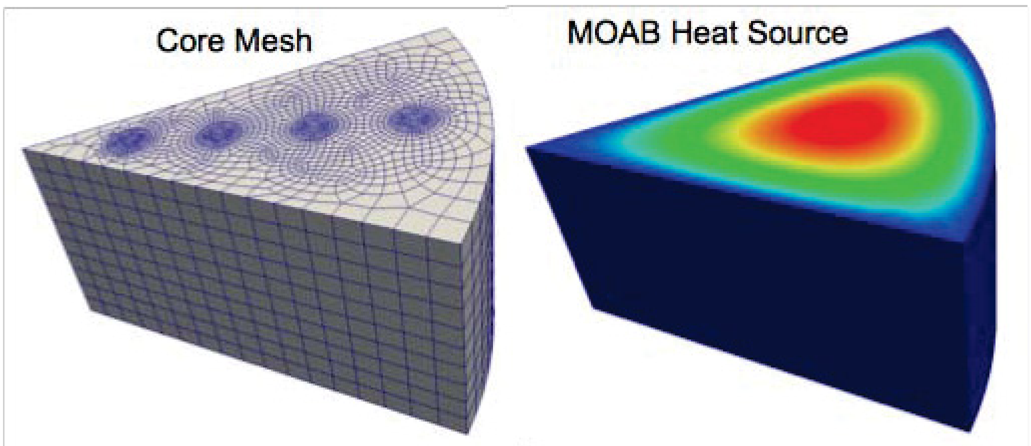

Figure 2: Three-dimensional mesh and MOAB heat source.

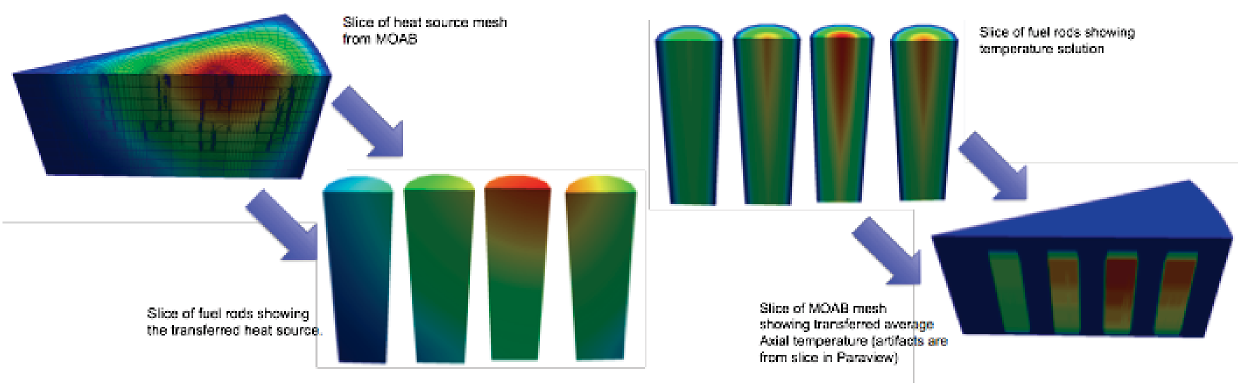

Figure 3: Mapping between cylindrical, three-dimensional meshes that have been sliced to show solution details. Note: visualization anomalies are not present in actual solution fields.

3.4 Ramifications

With this new capability in place, any MOOSE-based application can be easily linked with any MOAB based application or coupling tool within the NEAMS Integrated Framework. The MoabTransfer object exists within the MOOSE framework itself and is capable of moving fields between MOAB and MOOSE-based tools without modification to those MOOSE-based tools. This allows for development on

BISON, MARMOT, RELAP-7 and RAVEN to continue and their efforts easily linked back to the NEAMS Integrated Framework to perform larger multicomponent, multiphysics calculations.

This capability has already been delivered to all NEAMS partners through the MOOSE repository. The NEAMS MOOSE-based applications are all now equipped with this capability and are ready to be integrated with MOAB-based tools.

4. Additional Framework Enhancements in Support NEAMS Applications

The MOOSE framework supports two major NEAMS efforts: BISON and MARMOT. BISON is a nuclear fuel performance simulation tool capable of modeling the behavior of various types of fuel within nuclear reactors. MARMOT is a microstructure evolution simulation tool, primarily utilized in modeling the microstructural changes of nuclear fuel under irradiation. Both BISON and MARMOT require ongoing MOOSE development efforts in support of NEAMS goals. Changes to MOOSE are continually made available to the entire NEAMS community. Below are some specific enhancements the MOOSE team has worked on in support of these modeling and simulation efforts:

4.1 BISON

1. Ability to solve 1D spherical systems
2. New shared memory threading option using pthreads (for better execution on supercomputer architectures like BG/Q)
3. Assembly optimizations:
    a. Second derivative computation
    b. Variable reinitialization restricted to active variables
4. New TimeStepper system for enhanced control over time step size selection
5. Automatic mesh tiling support in MooseMesh
6. Automated boundary “reaction force” computation
7. Quadrature point based geometric search and thermal contact
8. Improved search and contact with triangles
9. New parameter types supported (vectors of Points and MooseEnums)
10. Script to assist in parameter studies (cluster_launcher.py)
11. Peacock (graphical user interface) enhancements:
    a. Enhanced comment preservation
    b. Parameter grouping
    c. Display dynamically generated meshes
12. Postprocessor table-wrapping for better readability (PPS_WIDTH)
13. Better startup with Serial Mesh (serial read and broadcast)
14. Handle variable multi-registration

4.2 MARMOT

1. Grain-Tracker/Remapping capability (see Figures 4 and 5)
2. Ability to dump detailed statistics on grain simulations
3. UserObject/AuxKernels dependency resolution
4. Initial Condition setup interface (helps with ICs that require periodic BC information)
5. Enhancements to MooseMesh Interface for automatic mesh generation
6. Automatic application of Periodic Boundaries to "Regular Orthogonal Meshes" P
7. Peacock (graphical user interface) enhancements:
    a. Improved handling of GlobalParams
    b. More robust updating of solution in Visualize tab
    c. Ability to open solution files directly
8. Mesh adaptivity enhancements:
    a. New adaptivity Markers added specifically for phase-field simulations
    b. max_h_level support added to new Adaptivity system
    c. Mesh adaptivity with Stateful material properties
9. Memory Logger
10. Find hung process script (for debugging)

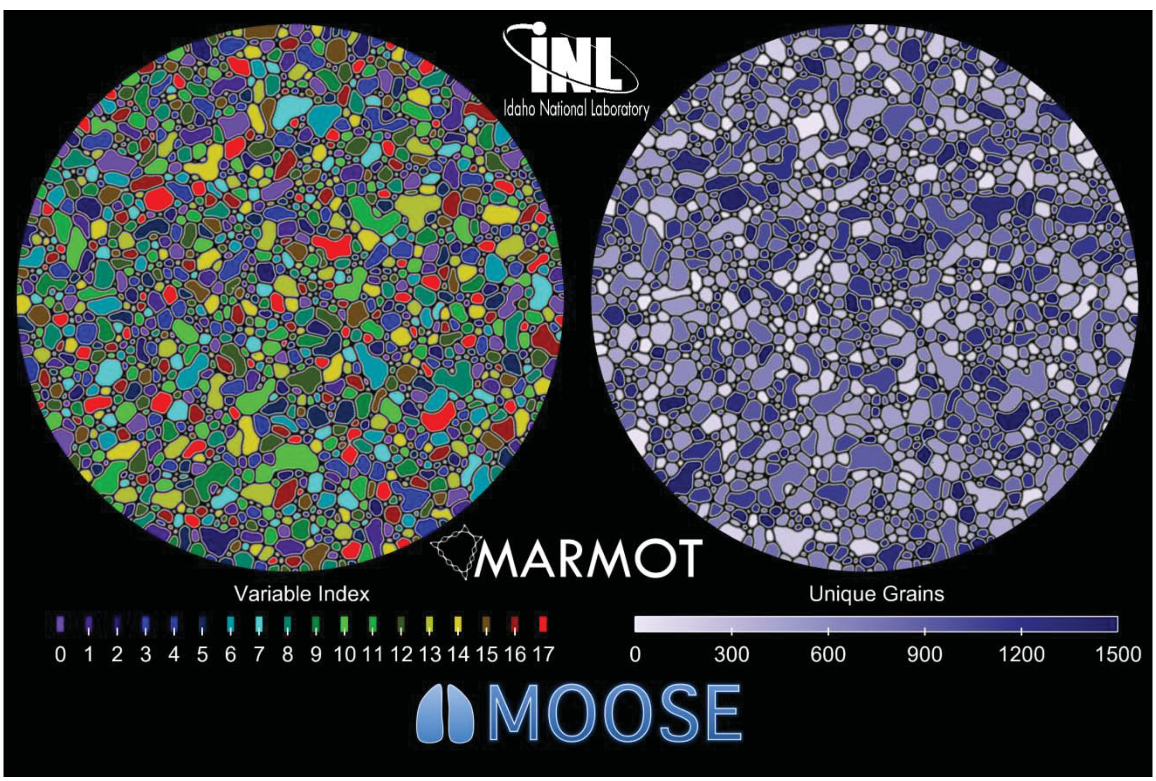

Figure 4: Two-dimensional unique grain-tracker results; 18 variables are used to track 1500 uique grains, greatly reducing solve time and memory consumption.

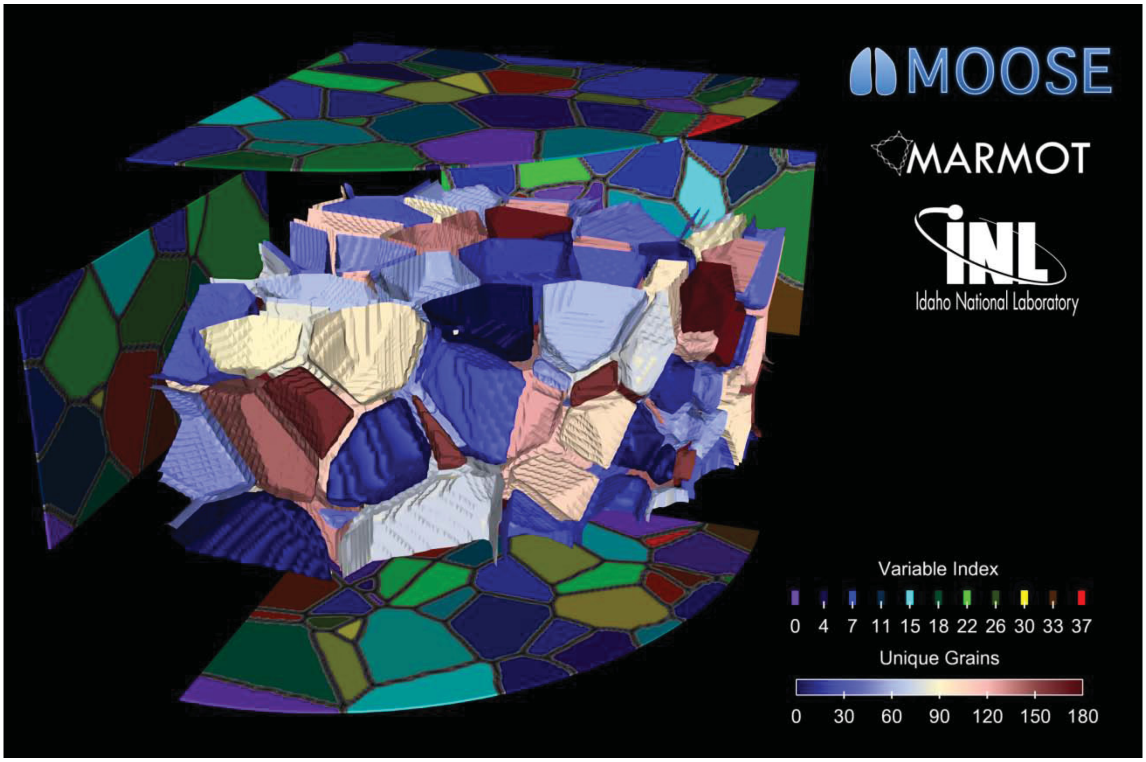

Figure 5: Three-dimensional unique grain-tracker results.

4.3 MOOSE Community Enhancements

In addition to the aforementioned BISON- and MAMROT-targeted enhancements, over 700 other modifications have been made to MOOSE that indirectly benefit these NEAMS projects. Currently, over thirty applications are based on MOOSE, and MOOSE itself is enhanced daily in support of each of these applications. This added benefit is a direct result of the "community-driven" development process utilized by the MOOSE multiphysics framework team. All MOOSE enhancements are immediately made available to the various software development teams using MOOSE—including all NEAMS partners.

5. Summary

Building on top of the capability already present within MOOSE we have been able to deliver a framework capable of integrating with the NEAMS effort while also enhancing the framework in direct support of NEAMS applications. MOOSE-based NEAMS applications will be able to be efficiently combined with other NEAMS projects without modification, allowing progress on those applications to continue unabated while additional work on the Integrated Framework is performed.

6. References

Tautges, T. J.; Meyers, R.; Merkley, K.; Stimpson, C.; Ernst, C. MOAB: A Mesh-Oriented Database, Sandia National Laboratories, 2004

Slattery, S. R.; Wilson, P. P.; Pawlowski., R. P. The Data Transfer Kit: A Geometric Rendezvous-Based Tool for Multiphysics Data Transfer. Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), May 5-9, 2013.

Williamson, R. L.; Hales, J. D.; Novascone, S. R.; Tonks, M. R.; Gaston, D. R.; Permann, C. J.; Andrs, D.; Martineau, R. C. Multidimensional Multiphysics Simulation of Nuclear Fuel Behavior. Journal of Nuclear Materials, 2012, 423, 149-163.

Tonks, M.; Gaston, D.; Millett, P.; Andrs, D.; Talbot, P. An object-oriented finite element framework for multiphysics phase field simulations. Comp. Mat. Sci., 2012, 51, 20-29.

Zou, L.; Peterson, J.; Zhao, H.; Zhang, H.; Andrs, D; Martineau, R. Solving Implicit Multi-mesh Flow and Conjugate Heat Transfer Problems With RELAP-7. Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), May 5-9, 2013.

Rabiti, C.; Mandelli, D.; Alfonsi, A.; Cogliati, J.; Kinoshita, R. Mathematical Framework for the Analysis of Dynamic Stochastic Systems with the RAVEN Code. Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013), May 5-9, 2013.