Nuclear Engineering and Design 367 (2020) 110751

Contents lists available at ScienceDirect

Nuclear Engineering and Design

journal homepage: www.elsevier.com/locate/nucengdes

Multiphysics for nuclear energy applications using a cohesive computational framework

R. Martineau[a,*], D. Andrs[a], R. Carlsen[a], D. Gaston[a], J. Hansel[a], F. Kong[a], A. Lindsay[a], C. Permann[a], A. Slaughter[a], E. Merzari[b], Rui Hu[b], A. Novak[c], R. Slaybaugh[c]

[a] Computational Frameworks Department, Idaho National Laboratory, P.O. Box 1625, Idaho Falls, ID 83404, USA
[b] Nuclear Engineering Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA
[c] Department of Nuclear Engineering, University of California, Berkeley, CA 94708, USA

ARTICLE INFO

Keywords:
MOOSE
Multiphysics
Cohesive coupling
Conjugate heat transfer

ABSTRACT

With the recent development of advanced numerical algorithms, software design, and low-cost high-performance computer hardware, reliance on coupled multiphysics to predict the behavior of complex physical systems is beginning to become standard practice. This is especially true in nuclear energy applications where strong nonlinear interdependencies exist between reactor physics, radiation transport, multi-scale nuclear fuels performance, thermal fluids, etc. Resolving these nonlinear dependencies requires choices in multiphysics software approaches. Two main multiphysics modeling and simulation approaches have emerged. The first is based upon "code coupling" where disparate physics codes of different software design, code languages, and spatial and temporal integration schemes are coupled together with relatively complex data passing interfaces. The second multiphysics software approach is to employ a "cohesive" framework where all physics applications are developed with a common software design, i.e., data structures, syntax, input format, integrated spatial and temporal discretization schemes, etc. Here we present the Multiphysics Object-Oriented Simulation Environment (MOOSE) development and runtime framework and describe the framework's cohesive modeling and simulation multiphysics approach. Then, a "cohesive-like" extension of the MOOSE framework is presented where MOOSE-based physics software applications are efficiently coupled to non-MOOSE (external) physics codes to form multiphysics applications using MOOSE's unique interface capabilities. Finally, several examples of MOOSE's cohesive and cohesive-like multiphysics applications will be demonstrated. These multiphysics demonstrations will incorporate both MOOSE-based applications and external codes, including Nek5000, RELAP-7, TRACE, BISON, and Pronghorn.

1. Introduction

Multiphysics can be defined as solving two or more physical models simultaneously in order to resolve interdependent nonlinearities between the models. The nonlinear dependencies are generally defined by coefficients of mathematical operators which are functions of the physical system's dependent variables. Multiphysics in nuclear power applications can be exceptionally complex. Even if we limit ourselves to "only" considering reactor physics, thermal fluids, and nuclear fuels performance, the nonlinear interdependencies resulting from the classical feedback mechanisms, including nonlinear heat conduction, two-phase flow with contaminants, and evolving material properties and chemical composition due to fission product generation, radiation damage, etc., will vary by several orders of magnitude. In addition, mathematical and numerical consistency is required between the various spatial and temporal discretization approaches. This consistency issue has always presented a challenge for traditional "code-coupling" strategies for multiphysics solutions where disparate codes developed with different solution algorithms, such as nonlinear implicit, linearized implicit, semi-implicit, and explicit time integration techniques, possibly employing various spatial discretization schemes and design philosophies with different languages, input syntax, data structures, and

* Corresponding author.
E-mail addresses: Richard.Martineau@inl.gov (R. Martineau), Robert.Carlson@inl.gov (R. Carlsen), Derek.Gaston@inl.gov (D. Gaston), Joshua.Hansel@inl.gov (J. Hansel), Fande.Kong@inl.gov (F. Kong), Alexander.Lindsay@inl.gov (A. Lindsay), Cody.Permann@inl.gov (C. Permann), Andrew.Slaughter@inl.gov (A. Slaughter), emerzari@anl.gov (E. Merzari), rhu@anl.gov (R. Hu), novak@berkeley.edu (A. Novak), slaybaugh@berkeley.edu (R. Slaybaugh).

https://doi.org/10.1016/j.nucengdes.2020.110751
Received 26 April 2020; Accepted 27 June 2020
Available online 07 August 2020
0029-5493/ Published by Elsevier B.V.

library dependencies, are "coupled" together with complex interfaces to provide a multiphysics simulation capability. Even installation and compilation of a code-coupled approach presents its own special difficulties.

The Department of Energy Office of Nuclear Energy's (DOE-NE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program defines "advanced modeling and simulation" as a multi-scale (in space and time), multiphysics simulation capability based on greater understanding of fundamental physical phenomena and enabled by the latest in software design, numerical methods, and advanced architecture. The design objective of advanced modeling and simulation is to develop multiphysics software based upon mathematically and numerically consistent formulations to achieve converged solutions of important interdependent physics at all scales. Verification of physics models and numerical methods in advanced modeling and simulation is part of "Best Practices" in software development and software quality. However, the key to developing a useful advanced modeling and simulation capability for a given physical system is a rigorous validation paradigm primarily founded in the integration of advanced modeling and simulation development with a rigorous multi-scale, multiphysics experimental program that is repeatable and delivers data with quantifiable uncertainties.

Idaho National Laboratory's (INL) Multiphysics Object-Oriented Simulation Environment (MOOSE) (Gaston et al., 2015) main design goal is to satisfy the key multi-scale, multiphysics aspect of the above NEAMS advanced modeling and simulation vision. MOOSE (www.mooseframework.org) began as a Laboratory Directed Research and Development (LDRD) in May of 2008. Inside MOOSE, the physical systems are generally represented (modeled) as a system of fully coupled nonlinear partial differential equations (PDEs). The Jacobian-Free Newton Krylov (JFNK) (Knoll and Keyes, 2004) method is implemented as a parallel nonlinear solver that naturally supports implicit coupling between physics equation systems. Extensive research in JFNK preconditioning (Kong et al., 2018) has been conducted and implemented into MOOSE to improve the parallel performance of large-scale multiphysics calculations. The MOOSE team has been the recipient of multiple INL Laboratory Director Awards and two Presidential Early Career Award for Scientists and Engineers (PECASE) awards. MOOSE itself received an R&D 100 award in July of 2014. INL obtained an open-source (LGPL 2.1) license for MOOSE in February of 2014.

The development of the MOOSE framework required somewhat different semantics from traditional software development efforts. MOOSE-based physics software packages are not referred to as "codes." The reason being is that these packages are not stand-alone codes and that the MOOSE framework is required to run the packages and constitutes the vast majority of the code. Thus, MOOSE-based physics software packages are often referred to as "applications." Since 2011, both national and international laboratories and universities have stood up more than sixty known MOOSE-based applications and over 58,000 known MOOSE framework build package downloads. MOOSE-based software applications are in development for nuclear power (radiation transport, reactor physics, nuclear plant safety and systems analysis, CRUD growth and effects, and multi-scale nuclear fuels performance), materials (fundamental materials development, effects of corrosion, damage and aging evolution, and irradiated material analysis), structural dynamics, multi-phase flow, waste analysis, geophysics (seismic, geothermal, geochemistry, and isotope transport) and advanced digital manufacturing (laser welding, spark plasma sintering).

The main benefit of MOOSE is its simplification of multiphysics algorithmic coupling for advanced modeling and simulation efforts. Because all MOOSE-based software applications are built using the same programming interfaces, following identical software design and library dependencies, there is a high degree of "cohesiveness" between the MOOSE-based applications. Because of this cohesiveness, MOOSE provides a simplified path to tightly couple physics that have vastly different space and time scales (multi-scale, multiphysics) through a unique data transfer system specifically designed for multi-scale simulations employing multiple software applications. This multi-scale approach, called "MOOSE MultiApps and Transfers," has been extensively employed in the numerical investigation of nuclear fuels performance, where the physics vary in space and time by many orders of magnitude, and functionally coupled to reactor physics, radiation transport, and thermal fluids (Gaston et al., 2015).

Practically speaking, there is no "solve-all" approach for elliptic, parabolic, and hyperbolic partial differential equations, represented in Eulerian or Lagrangian reference frames, with varying spatial and temporal integration schemes, and MOOSE is no different than any other in this respect. Furthermore, there is a sizeable investment into a considerable array of high-quality, high-performance physics software algorithmically optimized for a given set of governing equations. These codes should be considered for inclusion into MOOSE multiphysics. For example, Nek5000 (Nek5000 Version 17.0, 2017) and OpenMC (Romano et al., 2015) are active development efforts under the NEAMS program. Both of these codes have thousands of users and are highly successful. It is a great advantage for MOOSE to utilize these code efforts for multiphysics efforts in an efficient manner. A method we have developed, called "MOOSE-Wrapped Apps," utilizes MOOSE MultiApps and Transfers, along with a minimal Application Programmer Interface (API), to treat external codes as if they were MOOSE-based, in effect, a "cohesive-like" manner.

For this paper, we will describe MOOSE's multiphysics coupling strategies for nuclear power modeling and simulation efforts. In so doing, we will provide an overview of the MOOSE MultiApps and Transfers and MOOSE-Wrapped Apps. Finally, we will demonstrate several conjugate heat transfer (CHT) multiphysics examples, defined as tightly coupled solid state heat transfer and thermal energy transport in fluids, using both MOOSE-based software applications (codes) and external codes.

2. Overview of moose's cohesive multiphysics coupling strategies

All INL MOOSE-based applications are developed under the MOOSE framework software quality paradigm. This paradigm revolves around a common set of software quality requirements for all INL MOOSE-based software application development. All MOOSE-based applications are required to adhere to a strict MOOSE coding convention standard, syntax checking, unit tests, lines of code coverage, verification tests, and code documentation requirements (theory and user's manuals, software requirements, and software management plans), etc. Because of these requirements, and additional NQA-1 documentation requirements, MOOSE is recognized to be NQA-1 compliant, and all MOOSE-based application development under INL control adheres to this standard of quality. BISON (Williamson et al., 2012) nuclear fuels performance code and the RELAP-7 (Berry et al., 2014) thermal hydraulics simulator have undergone external assessment and found to be adhering to these requirements. The advantage of following identical software development practices for MOOSE-based software is that, in reality, all INL MOOSE-based software applications are part of the same "code". Thus, coupling of INL applications for multiphysics simulation is a straightforward process using MOOSE MultiApps and Transfers, which will be described below.

In the context of the computer science field, "Coupling" and "Cohesion" are software engineering terms that are used to describe the complexity of relationships between software modules. Coupling relates to external dependencies, including external libraries that are linked into an application and the measure of the degree of interdependence between those dependencies. A high-quality software will have "low coupling" leading to simplified maintenance and quality assurance. Cohesion is a measure of how well the modules fit together. All MOOSE-based applications natively build off from the MOOSE framework, inheriting MOOSE's underlying meshing and finite-element library (libMesh) (Kirk et al., 2006) and linear and nonlinear solvers (PETSc

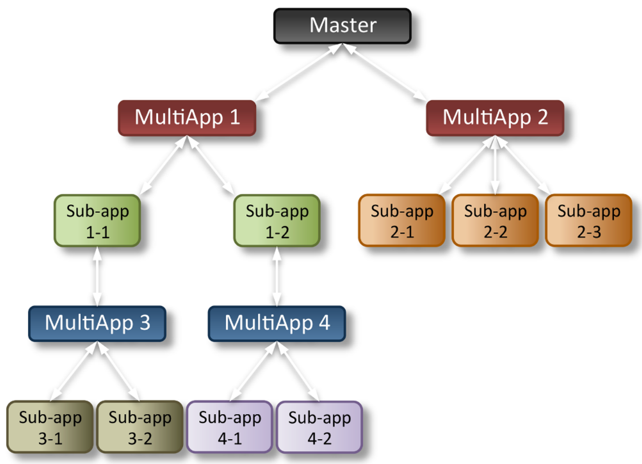


Fig. 1. Depiction of the MultiApp system.

(Balay et al., 2015), in order to leverage modern advanced software environments and numerical methods. Thus, all MOOSE-based applications have low coupling requirements. And, because all MOOSE-based software applications are developed on the MOOSE framework and follow MOOSE's NQA-1 compliant software development plan, these applications are highly cohesive.

2.1. MOOSE MultiApps and Transfers

The MOOSE MultiApps and Transfers systems give MOOSE the flexibility to run both tightly-coupled and loosely-coupled simulations among MOOSE-based applications. The MultiApp system handles the creation, spatial positioning, and execution control of multiple applications, which can be coupled together to build up more complex simulation models from existing applications. As shown in Fig. 1, each application can have one or more "MultiApps", which are the containers for one or more individual instances of a MOOSE-based application, which is called a "sub-app." To illustrate this construction, consider a full-scale AP1000 simulation consisting of the Rattlesnake transport solver, BISON fuels performance code, and the RELAP-7 thermal hydraulics simulator modeled in Wang et al. (2017). The reactor was modeled with a single instance of Rattlesnake (xxxx) covering the entire domain of the core. Inside of the Rattlesnake domain, one BISON MultiApp was created, which contained all 41,448 fuel rods. Each fuel rod was modeled as a separate independent instance of BISON. In this model, the MultiApp managed the separate MPI communicators for each instance of BISON. This setup drastically reduces the complexity of the communication model. The single Rattlesnake instance only needs to communicate with the collective BISON instances simplifying the solution transfers to and from the fuel rod meshes.

The MultiApp system is mainly responsible for the concurrent execution of one or more MultiApps. To achieve cohesive coupling, solution field transfers must be performed among all of the MultiApps. This is where the "Transfers" system comes in. MOOSE contains a pluggable system for moving information back and forth between pairs of Multiapps. Transfers are responsible for in-memory movement of solution information, including parallel communication due to domain decomposition. Transfers are classified into three categories: field transfers, scalar transfers, and custom transfers. Field transfers are spatial transfers between separate domains, generally on similar spatial scales. An example of this would be the movement of the "fission rate" or power from Rattlesnake to all of the BISON rods. Scalar transfers are used for moving aggregate quantities between MultiApps, often where there is a significant difference in spatial scale. An example of this would be the point calculations of a quantity such as thermal conductivity on a fuel rod obtained through an embedded mesoscale solve. Each mesoscale solve would yield only a single infinitesimally small value at the engineering scale. However, several of these point solutions could be combined to create an interpolative solution throughout the whole domain. Finally, the custom transfers category is a catch-all for every other special case not handled by the former categories. MOOSE's built-in "Layered Average" transfer falls into this category. With a layered average transfer, a new pseudo-field is created such as a 1D step function. This new function can then be sampled by the destination field on the receiving sub-app. An example of where this type of Transfer may be used would be the BISON rods sending temperature back to the Rattlesnake domain. While the BISON rods are indeed contained within the reactor core, they do not completely cover the entire domain. Therefore, a field transfer may not work as expected in the spaces between the fuel rods. The layered average object, however, can be sampled everywhere, thus filling in the gaps. Custom Transfers can be developed for many special cases, giving the developer a lot of flexibility in the algorithms that can be written to perform data transformation among applications.

2.2. MOOSE-wrapped apps

The combination of the MultiApp and Transfer systems makes MOOSE an excellent coupling platform. The generality of these systems can be abstracted to handle a wide variety of execution and data movement strategies. The maturation of these two systems has led to the "MOOSE-Wrapping" paradigm. MOOSE-Wrapping is the name given to the process of adding a MOOSE API to an existing application in a non-intrusive way. The wrapping process begins through the creation of a standard MOOSE-based application (called "storking") and linking in the external application such that it can be built as a library underneath the new skeleton MOOSE-based application. Once the application construction has been completed, a few interfaces must be extended to provide MOOSE with the hooks it needs to interact with the external application. The complete set of objects for providing the full MOOSE API are illustrated in Fig. 2. The two main objects that should be extended are the "ExternalProblem" and "ExternalMesh" objects. The former is a wrapper around the idea of execution control (e.g. initialization, take step, post execute, etc.) as well as the interface for syncing solution information from the external application to MOOSE's in-memory mesh data structures (ExternalMesh). The ExternalMesh object is the object that holds the application's external spatial information and acts as a map for the underlying solution data, which is stored in the ExternalProblem object. The duplication of the external application's data in a MOOSE-native format is intentional, to allow domain decomposition and avoid the complex reimplementation of advanced parallel field to field transfers. Normally, an application developer implements a conceptually straightforward direct copy of information in a "one-to-one" manner from the wrapped application's internal data structures to a mesh that exactly matches the geometric representation of the external application. Once those two objects exist, an application can begin to interact with other MOOSE-based and MOOSE-wrapped applications through two-way, in-memory, loose

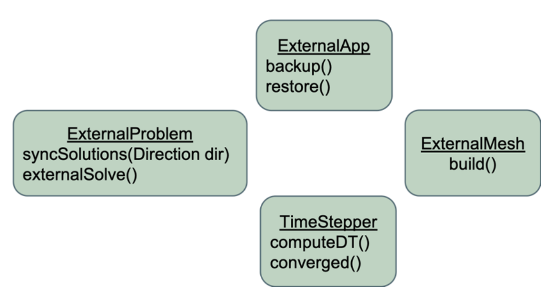


Fig. 2. Minimal API for MOOSE-Wrapping.

coupling.

To fully support the MOOSE-Wrapped Apps interface and to enable more advanced execution schemes, two more methods need to be implemented, namely the "backup" and "restore" methods. These two methods are designed to save and restore application states, which is necessary for tightly-coupled schemes such as a Picard iteration. In this scheme, each separate application must be completely solved multiple times to fully converge the coupled solutions. This requires the ability to save and restore state so that applications can "resolve" using the same starting criteria each iteration. These methods may be omitted if only pure "loose" coupling is intended to be used. Finally, it may be necessary to define a custom TimeStepper object to handle the computation of time step sizes for the external application and to define the behavior of handling failed solves or convergence criteria. The complete set of extended objects is housed within the MOOSE-Wrapped application, providing the cohesive-like interface for coupling with other MOOSE-based and MOOSE-Wrapped applications. To demonstrate the usage of MOOSE-Wrapped App capability, we use a native PETSc-based application as an example, where we created an "ExternalMesh" class called PETScDMDAMesh that mirrors the structured mesh information of PETSc DMDA object. Based on the mesh information, the solution of external native PETSc application can be directly copied over to the MOOSE-wrapped application, and then the copied solution can be transferred to any MOOSE-based applications using the existing transfers. The same mechanism can be applied in the opposite direction as well. This example is implemented as a regular MOOSE module called PetscExternalSolver that is available to all MOOSE users.

3. Demonstrations of CHT using moose cohesive coupling approaches

Here, we will demonstrate several conjugate heat transfer (CHT) multiphysics applications using both MOOSE-based software applications (codes) and external codes. The first is the coupling of the MOOSE-based multi-dimensional CHT Pronghorn application (Novak et al., 2018a) with one-dimensional flow using the RELAP-7 nuclear plant systems analysis application, which is also MOOSE-based. In this example, MOOSE MultiApps and Transfers are used, as both applications are MOOSE-based. For the second demonstration, we will describe a MOOSE-Wrapped Application, called BlueCRAB, that couples the NRC TRACE nuclear plant systems analysis code to the NEAMS MOOSE-based BISON nuclear performance application. And finally, the Cardinal MOOSE-Wrapped Application is designed to provide coupled high-resolution multiphysics simulations in pebble beds using the Nek5000 computational fluid dynamics (CFD) code and BISON.

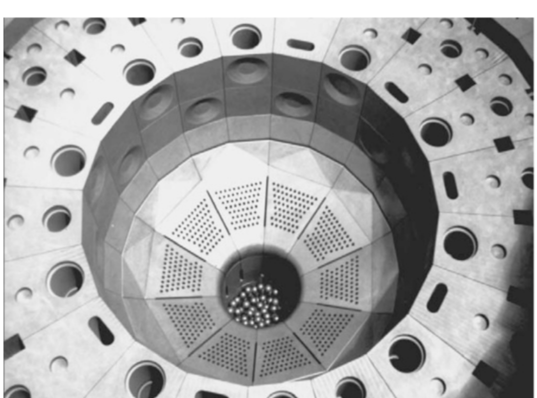
3.1. Three-dimensional heat transfer in a HTGR core tightly coupled to one-dimensional flow using MOOSE multi-app and transfers

Under DOE-NE's Office of Advanced Reactor Technologies (ART), the Advanced Reactor Concepts (ARC) program supports the research of advanced reactor subsystems and addresses long-term technical barriers for the development of advanced nuclear fission energy systems utilizing coolants such as liquid metal, fluoride salt, or gas. DOE-NE NEAMS program supports ART through the development of advanced modeling and simulation tools, algorithmically designed and optimized for specific reactor concepts. As described before, these new advanced simulation tools are purposefully developed to operate in a multi-scale, multiphysics fashion to provide a science-based predictive capability and to minimize uncertainties.

In this section, we will couple two NEAMS-developed MOOSE-based applications using MOOSE MultiApps and Transfers to couple one-dimensional flow from RELAP-7 to the multi-dimensional reactor CHT algorithm of Pronghorn (multiplier and upwinding for the multiscale transport capability in rattlesnake" Progress in Nuclear Energy, 101:381–393, 2017). RELAP-7 (Reactor Excursion and Leak Analysis Program) is the next generation RELAP nuclear systems safety analysis code being developed at the INL. The most important development goals of RELAP-7 are to take advantage of the previous thirty years of advancements in computer architecture, software design, numerical methods, and physical models. The four major improvements are 1) A well-posed seven-equation two-phase flow model (liquid, gas, and interface pressures); 2) Improved numerical approximations resulting in second-order accuracy in both space and time; 3) Implicit tightly coupled time integration for long duration transients, providing plant behavior for full life fuel cycle evaluations; 4) The ability to tightly couple to higher fidelity physics, both MOOSE-based and external, as will be demonstrated here by coupling to three-dimensional CHT in Pronghorn.

Under the NEAMS program, a collaboration formed between the University of California Berkeley (UCB), Argonne National Laboratory (ANL), and INL to build Pronghorn, a multi-dimensional, coarse mesh reactor simulator conceived at INL and based upon the MOOSE framework. Pronghorn physics can be described as homogenized conjugate heat transfer (CHT), where each finite element may contain a mixture of coolant, fuel, moderator, or other core internals. It is designed to bridge the spatial length scales between high-resolution lower-length scale calculations and plant scale systems codes, such as SAM (Hu, 2017) and RELAP-7. In other words, Pronghorn's homogenized CHT approach can be defined as resolving large scale flow features with tightly coupled solid state heat transfer and homogenizing small scale flow features and solid-state heat conduction as a mixture.

To illustrate Pronghorn's homogenized CHT approach, view the left-hand picture of Fig. 3. This is a picture of the HTR-10 lower reflector. For the CHT simulation that follows, the riser channels and the annular



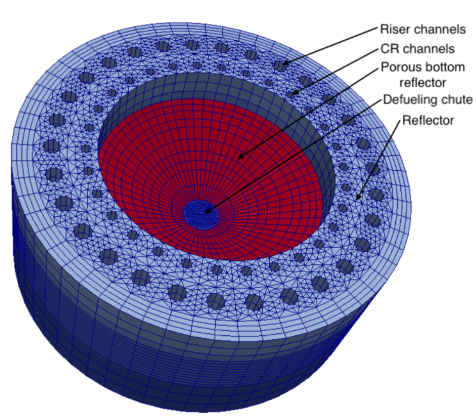


Fig. 3. Lower reflector of the HTR-10 (courtesy of J.C. Kuijper) and a finite element discretized lower reflector of the HTR-PM.

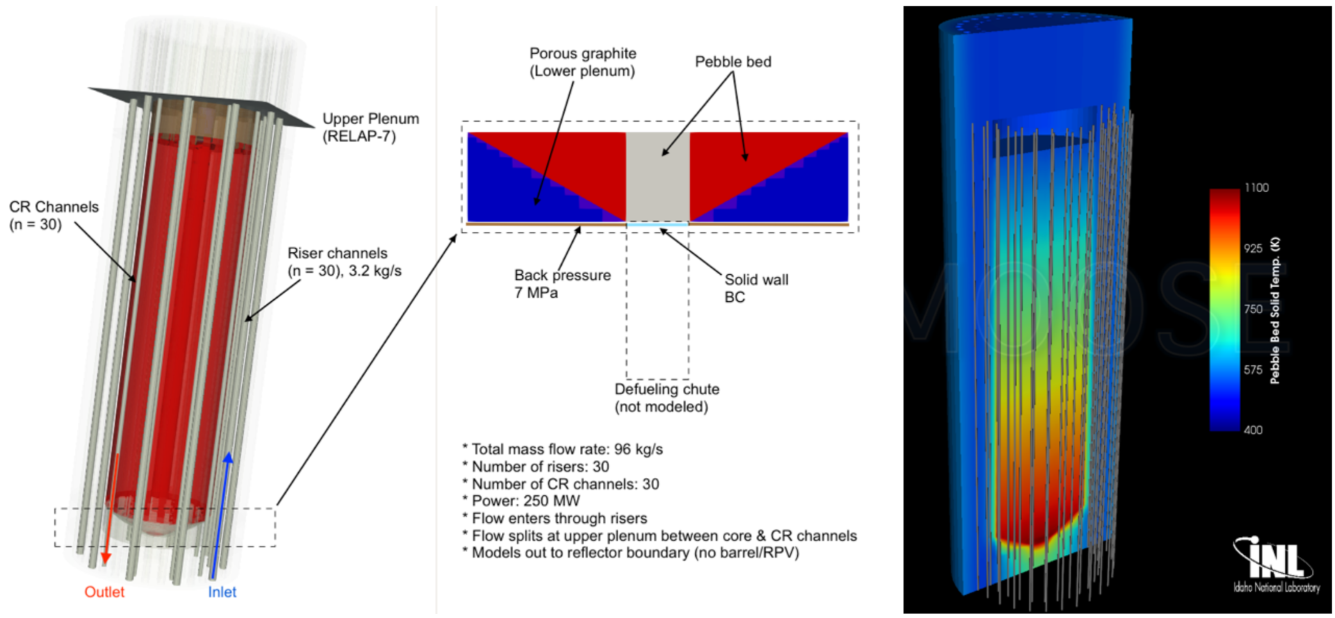


Fig. 4. Three-dimensional representation of HTR-10 reactor using Pronghorn, BISON, RELAP-7, and Rattlesnake.

control rod channels will be modeled by one-dimensional flow channels in RELAP-7. The bottom of the reflector has several hundred holes and the pebble defueling chute that is illustrated in the left-hand picture of Fig. 4. The right-hand picture of Fig. 3 has a red-colored finite element mesh representing a "porous" representation, or homogenized CHT.

The right-hand picture of Fig. 4 shows preliminary results of this coupling between Pronghorn and RELAP-7. Future work will include coupling with radiation transport with Rattlesnake, TRISO fuels performance with BISON, and balance of plant with SAM and Rattlesnake.

3.2. BlueCRAB: A MOOSE-wrapped application coupling NRC's TRACE code to the NEAMS BISON code

Beginning in May of 2016, the DOE-NE NEAMS program began formal discussions with the Nuclear Regulatory Commission's (NRC) Office of Nuclear Regulatory Research (NRR) on the possibility of the NRC adopting NEAMS advanced software for regulatory and licensing efforts for non-Light Water Reactor (LWR) advanced reactor concepts and accident tolerant fuel concepts (ATF). Codes used by the NRC for confirmatory analysis have been designed and assessed for LWRs and are not immediately extendable to non-LWR designs, such as gas-cooled reactors, sodium fast reactors, molten salt reactors, and micro-reactors. By October of 2017, the NEAMS program was working directly with NRR to develop a strategy where MOOSE coupling strategies would be applied to NRC LWR codes for inclusion of ATF analysis capability into the NRC's mature and proven LWR simulation capability. In addition, advanced simulation tools developed by NEAMS for non-LWR reactor concepts would be adapted and coupled to NRC codes as necessary. The first success along these efforts came in October of 2017 when MOOSE was wrapped around TRACE (Nuclear Regulatory Commission and "Trace, 2010). MOOSE and TRACE could then be compiled under one executable capable of exchanging information and spawn single or simultaneously multiple instances of TRACE. The NRC named the MOOSE-Wrapped application BlueCRAB (Comprehensive Reactor Analysis Bundle), or CRAB for short. Fig. 5 illustrates the NRC vision for combining their LWR codes with the NEAMS modeling and simulation tools for advanced fuel analysis and non-LWR confirmatory calculations.

The Block-Implicit Simulation of Nuclear fuels (BISON) is a MOOSE-based finite element-based nuclear fuel performance code applicable to a variety of fuel forms including light water reactor fuel rods, TRISO particle fuel, mixed oxide (MOX) fuel, and metallic rod and plate fuel and for 1D spherical, 2D axisymmetric or 3D geometries. It solves the fully coupled equations of thermo-mechanics and species diffusion, for either two-dimensional axisymmetric or three-dimensional geometries. Fuel models are included to describe temperature and burnup-dependent thermal properties, fission product swelling, densification, thermal and irradiation creep, fracture, and fission gas production and release. Plasticity, irradiation growth, and thermal and irradiation creep models are implemented for clad materials. Models are also available to simulate gap heat transfer, mechanical contact, and the evolution of the gap/plenum pressure with plenum volume, gas temperature, and fission gas addition.

The TRAC/RELAP Advanced Computational Engine (TRACE) (Nuclear Regulatory Commission and "Trace, 2010) is the latest in a series of advanced, best-estimate reactor systems codes developed by the U.S. Nuclear Regulatory Commission (NRC) for analyzing transient and steady-state neutronic-thermal-hydraulic behavior in light water reactors. It is the product of a long-term effort to combine the capabilities of the NRC's four main systems codes (TRAC-P, TRAC-B, RELAP5 and RAMONA) into one modernized computational tool. TRACE has been designed to perform best-estimate analyses of loss-of-coolant accidents (LOCAs), operational transients, and other accident scenarios in pressurized light-water reactors (PWRs) and boiling light-water reactors (BWRs). It can also model phenomena occurring in experimental facilities designed to simulate transients in reactor systems. Models used include multidimensional two-phase flow, non-equilibrium thermo-dynamics, generalized heat transfer, re-flood, level tracking, and reactor kinetics. Automatic steady-state and dump/restart capabilities are also provided.

The actual coupling of TRACE two-phase flow variables and exchanging heat flux and temperature information with BISON cladding finite element mesh began in April of 2018. By September of 2018, excellent BlueCRAB results were obtained for a loss of cooling due station blackout (SBO) event similar to the Fukushima nuclear power plant event of March 2011.

With early success, BlueCRAB's sophistication was increased to include adaptive meshing for fine resolution of boiling fronts in TRACE. The MOOSE FineMeshTransfer object was created as a custom transfer for handling TRACE's fine mesh re-nodalization capability and transferring the variables to BISON. It works by querying TRACE for a list of "elevations" or 1D coordinate values and solutions. A piecewise,

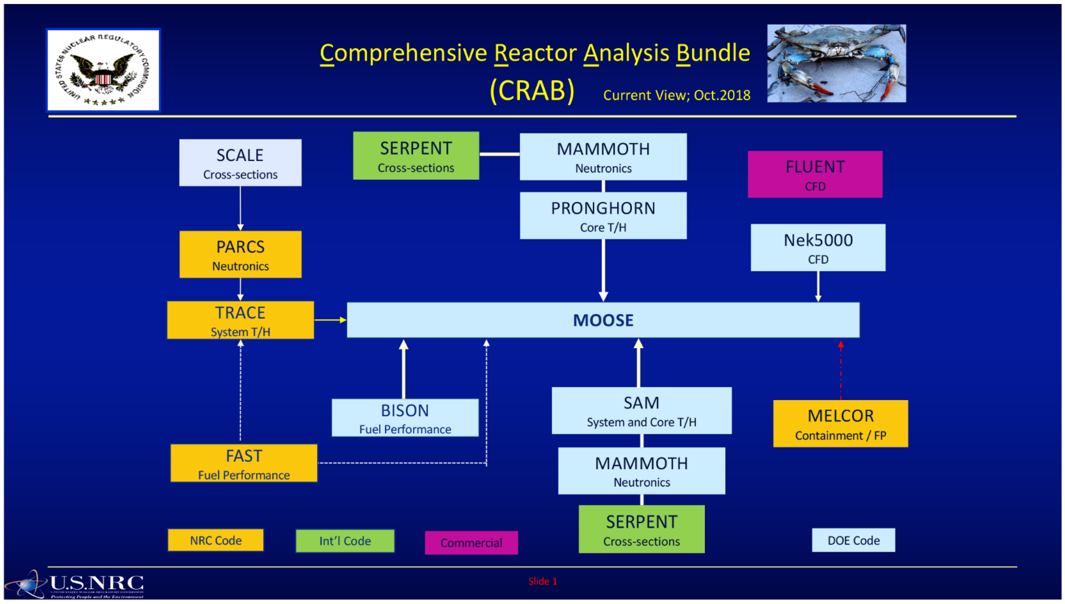


Fig. 5. Snapshot of the BlueCRAB vision (courtesy of Stephen Bajorek, NRC).

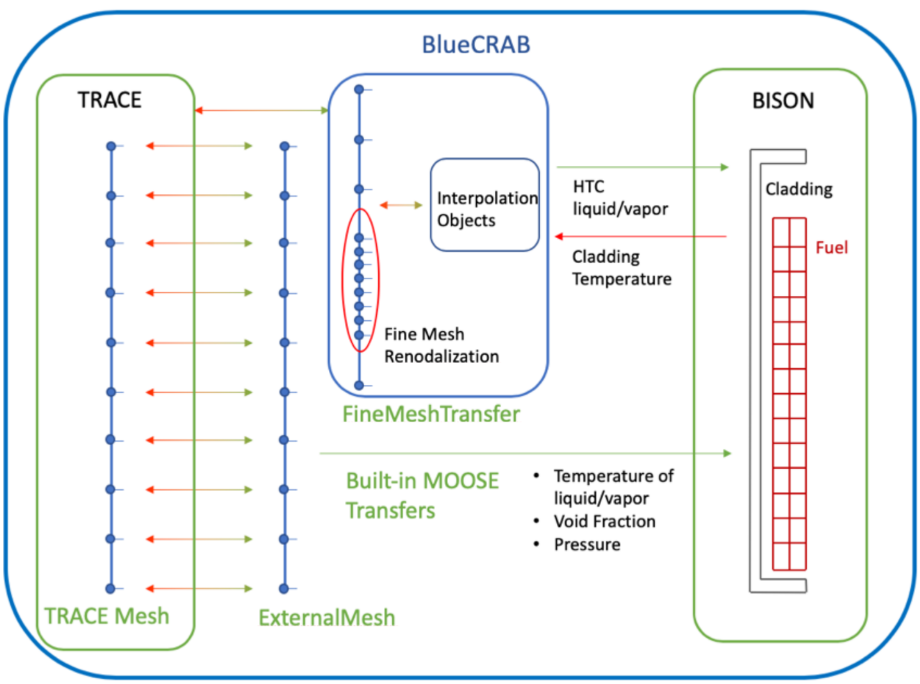


Fig. 6. Illustration of the BlueCRAB MOOSE-Wrapped Application for TRACE-BISON Coupling.

continuous solution is then constructed, which can be sampled anywhere within that domain for an interpolated value. These values are then transferred back to a MOOSE-based application in parallel. Fig. 6 illustrates the data transfer process for the BlueCRAB MOOSE-Wrapped Application for TRACE-BISON.

With the FineMeshTransfer capability for re-flooding and quench fronts, BlueCRAB development turned toward Large Break Loss of Coolant Accidents (LB-LOCA). This is a most challenging problem that will stress the ability to couple two disparate codes together. Here, we will present the results of this development effort applied to simulating the Idaho National Engineering Laboratory's (INEL) Loss of Fluid Test (LOFT) facility (Nalezny, 1983). The LOFT facility was a 50MWt single-loop pressurized water reactor designed to be operated to simulate LB-LOCAs. Funded by the NRC Reactor Safety Research Program, the facility was able to simulate a larges series of full-sized pipe breaks for various LOCA events by employing a second "broken" loop, a large blowdown suppression tank, and fast-acting valves. Fig. 7 is a well-known diagram of the INEL Loft Facility.

The BlueCRAB demonstration presented here is the often-simulated LOFT L2-5 experiment. This experiment was conducted to investigate a PWR system and core thermal response during Emergency Core Cooling System (ECCS) reflood following the simulated double-ended cold leg break transient.

Fig. 8 shows the graphical comparison of the results obtained with BlueCRAB (TRACE-BISON), stand-alone TRACE (green line), and LOFT L2-5 LOCA data (red line). Throughout the entire transient, BlueCRAB results are more representative of the experimental data than the standalone TRACE results.

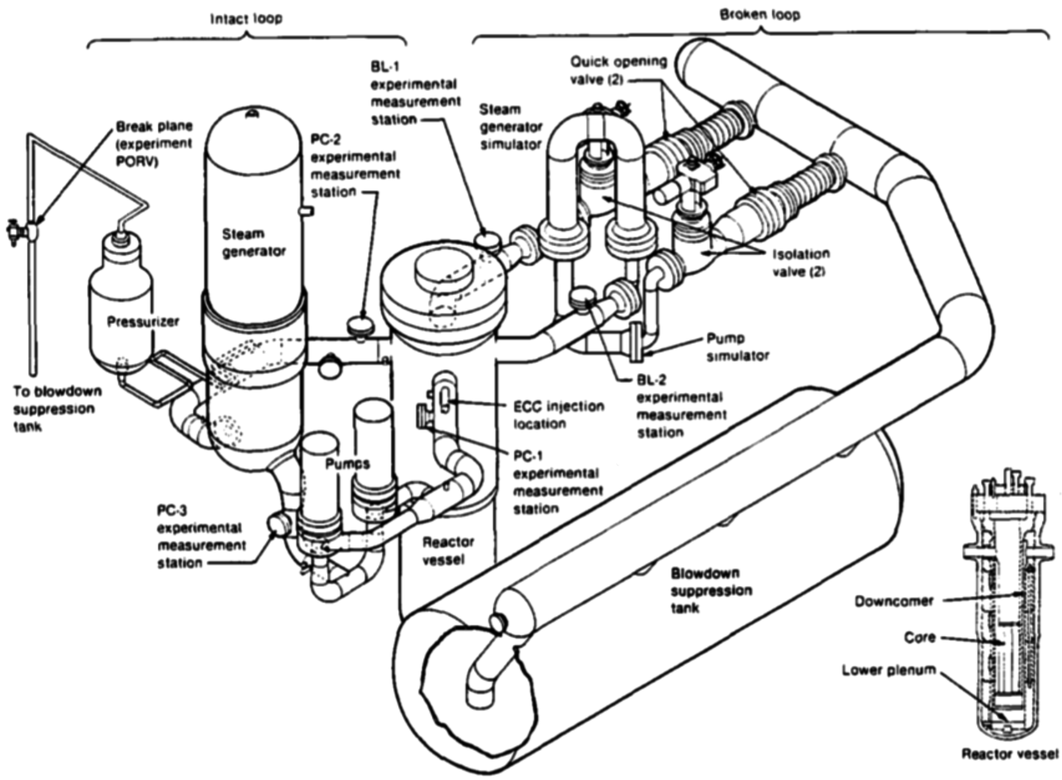


Fig. 7. INEL LOFT system configuration, from Technical Report NUREG/CR-3214.

3.3. Cardinal: A MOOSE-wrapped application for Lower-Length scale multiphysics calculation

Cardinal is a new MOOSE-Wrapped application that leverages the multi-app structure of MOOSE to deliver coupled high-resolution multiphysics simulations in pebble beds. OpenMC, an open source Monte Carlo code (Romano et al., 2015; Novak et al., 2018b), provides the power distribution within the pebbles, which is then transferred to BISON for accurate fuel performance calculations. The flow field and heat transfer in the coolant are solved using a low-Mach number approximation (i.e, allowing for a non-zero thermal divergence of the flow field) by Nek5000, an open source spectral element solver developed under NEAMS. BISON and Nek5000 are coupled through solution exchange of heat flux and temperature at the boundary. The coupling of heat flux and temperature can be lagged in time or be tightly coupled through a Picard iteration. Fig. 9 provides an example of the flow field in a pebble bed simulated with Cardinal (Nek5000).

Cardinal leverages recent work done to couple Nek5000 into MOOSE. For example, we have demonstrated that accurate solutions can be achieved using polynomial expansions of the flux and temperature and exchanging coefficients between codes. Fig. 10 shows an example for a cylindrical rod, illustrating rapid convergence of the

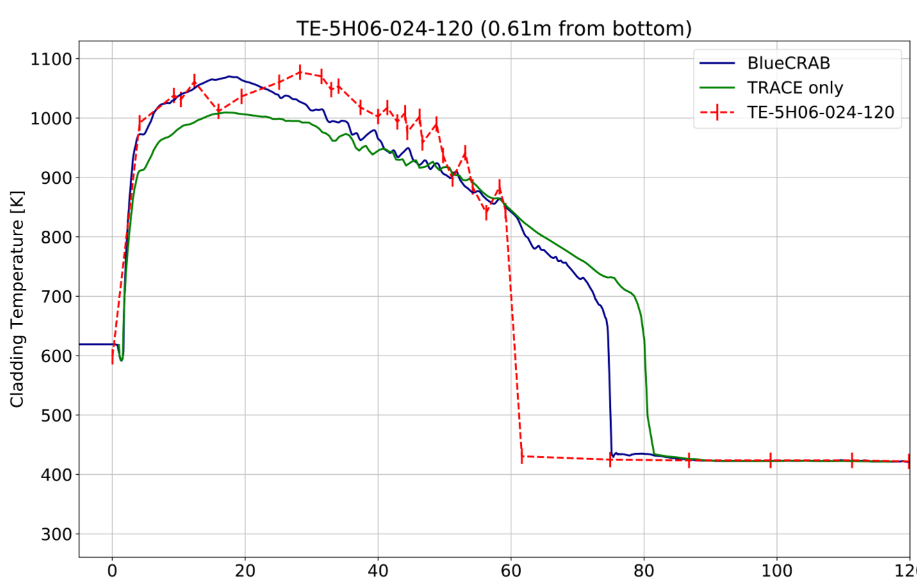


Fig. 8. Comparison between BlueCRAB, standalone TRACE, and LOFT L2-5 experimental results.

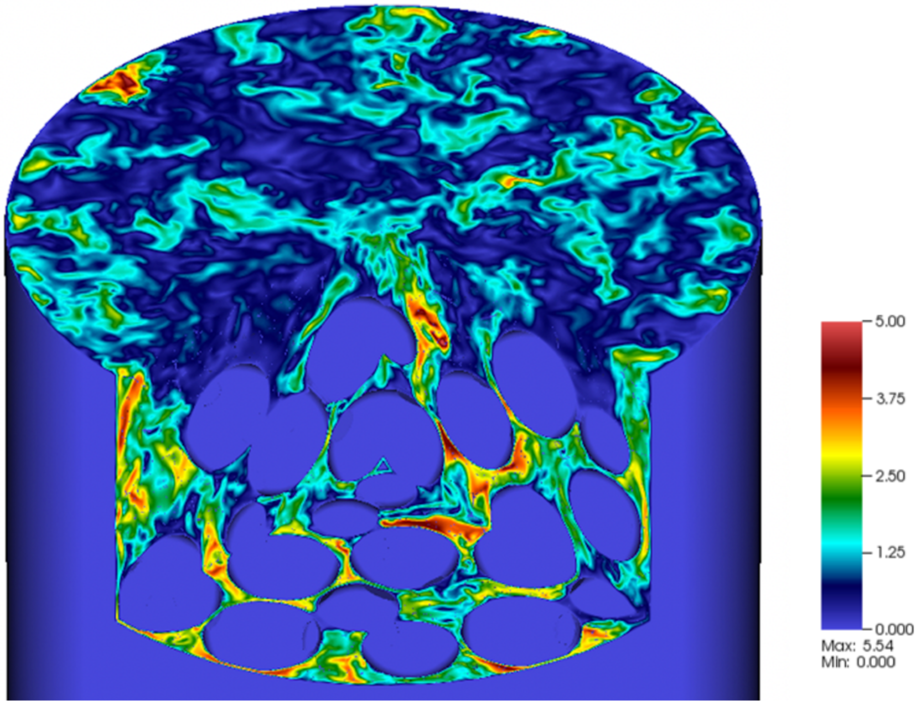


Fig. 9. Nek5000 simulation of the flow in a pebble bed (streamwise velocity).

solution transfer error as the order of the polynomial expansion increases. This is achieved despite completely different meshes being utilized on the structure side and fluid side. In Cardinal we are expanding on this work, implementing a general-purpose solution transfer mechanism that is not tied to a specific geometry type.

Cardinal will be used to simulate the Neutronics, flow field, and heat transfer for complex pebble bed domains. The data obtained will be used to generate needed closure relations in Pronghorn, RELAP-7 and SAM, the engineering-scale and system-scale thermal–hydraulic simulators of NEAMS.

4. Conclusions

We have demonstrated the MOOSE framework's extensive multiphysics coupling capability for both MOOSE-based applications and external codes. The MultiApps and Transfers capability is a uniquely efficient approach for achieving multiphysics coupling between MOOSE-based applications. With MultiApps and Transfers, both loosely-coupled (one-way data transfer) and tightly-coupled (implicitly converged) multiphysics solutions are available. This "cohesive" coupling between MOOSE-based applications has been utilized to couple with non-MOOSE codes using the MOOSE-Wrapped App capability, which utilizes MOOSE MultiApps and Transfers, along with a minimal Application Programmer Interface (API), to treat external codes as if they were MOOSE-based, in effect, a "cohesive-like" manner. The MOOSE-Wrapped App approach inherits both loosely-coupled (one-way data transfer) and tightly-coupled (implicitly converged) multiphysics solutions with full restart capability. This is a significant contribution to the advanced simulation community and adds great value to the MOOSE framework.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

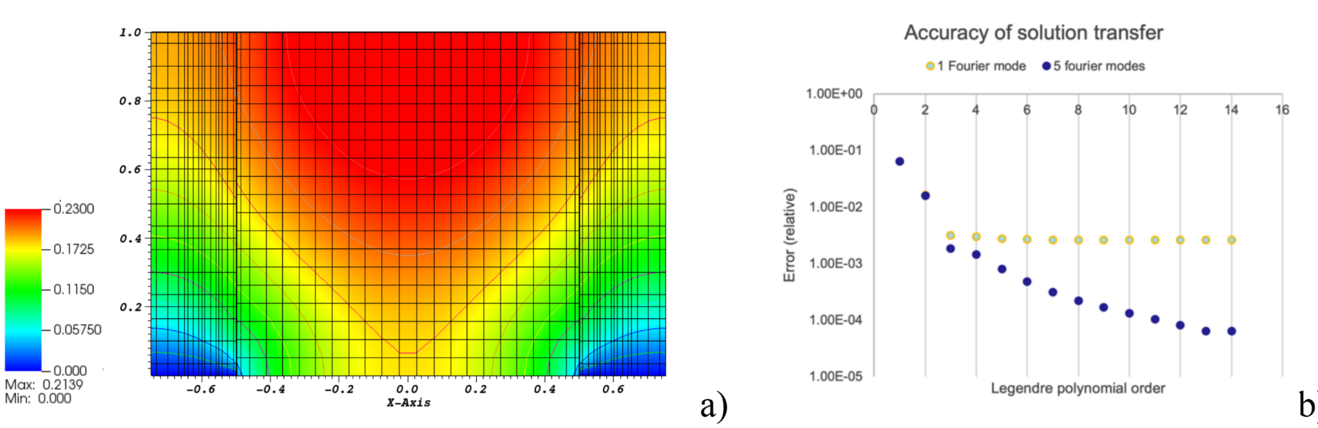


Fig. 10. Early Results on MOOSE-Nek5000 coupling. Laminar Axial flow on a cylindrical fuel rod: a) symmetry plane cut of the non-dimensional temperature illustrating the different meshes between solid and fluid. b) convergence of the error as a function of polynomial order.

Acknowledgments

This work was funded by the Department of Energy Nuclear Energy Advanced Modeling and Simulation program. This manuscript has been authored by Battelle Energy Alliance, LLC under Contract No. DE-AC07-05ID14517 with the US Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

This research made use of the resources of the High-Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

We would like to acknowledge the considerable contribution to the NRC's BlueCRAB MOOSE-Wrapped App by Drs. Matthew Bernard and Stephen Bajorek of the NRC and Russell Gardener of INL. Without their effort and expertise, the TRACE-BISON coupling could not have occurred.

References

S. Balay S. Abnyankar M. Adams J. Brown P. Brune K. Buschelman L. Dalcin V. Eijkhout W. Gropp D. Karpeyev D. Kaushik M. Knepley L. Curfman McInnes K. Rupp B. Smith S. Zampini H. Zhang "PETSc User's Manual. 3.6", Mathematics and Computer Science Division, Argonne National Laboratory 2015.

R. A. Berry, L. Zou, H. Zhao, H. Zhang, J. W. Peterson, R. C. Martineau, S. Y. Kadioglu, and D. Andrs, "RELAP-7 theory manual," Tech. Rep. INL/EXT-14-31366 (Rev. 2), Idaho National Laboratory, U.S.A. (2014).

Gaston, D.R., Permann, C.J., Peterson, J.W., Slaughter, A.E., Andrs, D., Wang, Y., Short, M.P., Perez, D.M., Tonks, M., Ortensi, J., Martineau, R.C., 2015. Physics-based multiscale coupling for full core nuclear reactor simulation. Ann. Nucl. Energy 84.

Hu, R., 2017. A fully-implicit high-order system thermal-hydraulics model for advanced non-LWR safety analyses. Ann. Nucl. Energy 101, 174–181.

Kirk, B.S., Peterson, J.W., Stogner, R.H., Carey, G.F., 2006. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. Eng. Comput.-Germany 22 (3–4), 237–254.

Knoll, D.A., Keyes, D., 2004. 2003, "Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications". J. Comput. Phys. 193, 357–397.

Kong, F., Wang, Y., Schunert, S., Peterson, J.W., Permann, C.J., Andrš, D., Martineau, R.C., 2018. A fully coupled two-level Schwarz preconditioner based on smoothed aggregation for the transient multigroup neutron diffusion equations. Numerical Linear Algebra with Appl. 25, 21–62.

Nalezny, C.L., 1983. "Summary of Nuclear Regulatory Commission's LOFT Program Experiments", NUREG/CR-3214. NRC, US.

Nek5000 Version 17.0, Argonne National Laboratory 2017 Illinois. Available.

April J. Novak Ling Zou, John W. Peterson, David Andrs, Joe Kelly, Rachel N. Slaybaugh, Richard C. Martineau, and Hans D. Gougar, "Pronghorn Theory Manual," Tech. Rep. INL/EXT-18-44453, Idaho National Laboratory, U.S.A. (2018).

A. Novak, P. Romano, B. Wendt, R. Rahaman, E. Merzari, L. Kerby, C. Permann, R. Martineau, and R.N. Slaybaugh, "Preliminary coupling of OpenMC and Nek5000 within the MOOSE framework." PHYSOR 2018: Reactor Physics paving the way towards more efficient systems, Cancun, Mexico, April 22-26, 2018.

Romano, P.K., Horelik, N.E., Herman, B.R., Nelson, A.G., Forget, B., 2015. OpenMC: A state-of-the-art Monte Carlo code for research and development. Ann. Nucl. Energy 82, 90–97.

U.S. Nuclear Regulatory Commission "Trace, 2010. V5.0 Theory Manual: Field Equations, Solutions Methods and Physical Models", US Nuclear. Regulatory Commission.

Y. Wang, S. Schunert, M. DeHart, R. Martineau, and W. Zheng. "Hybrid pn-sn with lagrange multiplier and upwinding for the multiscale transport capability in rattlesnake" Progress in Nuclear Energy, 101:381 – 393, 2017. Special Issue on the Physics of Reactors International Conference PHYSOR 2016: Unifying Theory and Experiments in the 21st Century.

Williamson, R.L., Hales, J.D., Novascone, S.R., Tonks, M.R., Gaston, D.R., Permann, C.J., Andrs, D., Martineau, R.C., 2012. Multidimensional multiphysics simulation of nuclear fuel behavior. J. Nucl. Mater. 423, 149.