INL/CON-22-67920-Revision-0

Assembling Multiphysics Nuclear Reactor Simulations Using the MOOSE Framework

November 2022

Guillaume Louis Giudicelli, Cody J Permann, Fande Kong, Derek R Gaston, A. Abdelhameed, Emily Shemon, Yinbin Miao

Changing the World's Energy Future

Idaho National Laboratory

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

INL/CON-22-67920-Revision-0

Assembling Multiphysics Nuclear Reactor Simulations Using the MOOSE Framework

Guillaume Louis Giudicelli, Cody J Permann, Fande Kong, Derek R Gaston, A. Abdelhameed, Emily Shemon, Yinbin Miao

November 2022

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

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

Assembling Multiphysics Nuclear Reactor Simulations Using the MOOSE Framework

C. Permann,* A. Lindsay,* G. Giudicelli,* F. Kong,* D. Gaston,* E. Shemon,† Y. Miao,† A. Abdelhameed†

*Idaho National Laboratory, 1955 N. Fremont Avenue, Idaho Falls, ID, 83415, cody.permann@inl.gov
†Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, eshemon@anl.gov

doi.org/10.13182/T127-39768

INTRODUCTION

The Multiphysics Object-Oriented Simulation Environment (MOOSE) [1] is an open-source, parallel finite element framework which provides the foundation for many advanced modeling and simulation tools developed under the Department of Energy's (DOE's) Nuclear Energy Advanced Modeling and Simulation (NEAMS) program [2] for the analysis of advanced reactors. The MOOSE framework provides the common foundational capability on which many NEAMS codes for reactor analysis are built. The MOOSE framework also includes several systems to assemble unique workflows and coupling among MOOSE-based applications. In particular, the MultiApp and Transfer systems are widely used to assemble different MOOSE-based or MOOSE-wrapped physics applications together to perform loosely or tightly coupled multiphysics simulations. The National Reactor Innovation Center's (NRIC's) Virtual Test Bed (VTB) [3] hosts publicly available nuclear reactor multiphysics simulation examples which leverage MOOSE's MultiApp system to meet the modeling needs of different reactor types. The flexibility and robustness of coupling provided by MOOSE permits rapid development of coupled physics models for a wide range of reactor types and events.

MOOSE PLUG-AND-PLAY MODEL FOR MULTIPHYSICS REACTOR ANALYSIS

The MultiApp and Transfer systems within MOOSE provide the functionality to couple different MOOSE-based or MOOSE-wrapped physics applications without additional code being written to execute the interactions. The coupling workflow and transfer of data is specified within input files, and all operations are handled internally by MOOSE. In today's nuclear reactor landscape which consists of a wide variety of proposed advanced reactor concepts, each with specific physics needs, the MOOSE plug-and-play model provides the flexibility to assemble existing "single" physics applications such as neutronics, thermal hydraulics, thermo-mechanics, chemistry, fuel performance, and systems analysis into a customized multiphysics workflow. A liquid-fueled molten-salt reactor, for example, would have a different workflow from a liquid-metal-cooled fast reactor, although both workflows may leverage some common physics components.

Common NEAMS tools for reactor analysis which can be assembled using MOOSE's MultiApp system include Griffin [4] (neutronics), Pronghorn [5] (engineering scale flow), Bison [6] (fuel performance), SAM [7] (systems analysis), Sockeye [8] (heat-pipe simulator), and Cardinal [9] (includes NekRS computational fluid dynamics), although this list is not exhaustive.

Numerous examples of customized multiphysics workflows for reactor analysis are publicly available on the VTB site. These examples leverage subsets of the above codes to model steady state and transient events for liquid and solid-fueled reactors, gas and liquid coolants, and thermal and fast spectrums. Documentation on using MOOSE's MultiApp system to construct multiphysics reactor workflows is available on the VTB site. Additionally, a more general tutorial on MultiApps is available on the MOOSE website [10].

Multiphysics Simulation Hierarchy

The MultiApp System in MOOSE can be used to build multiphysics workflows leveraging loose or tight coupling (fixed point iterations). These workflows can be visualized as a hierarchy in which the top-level application is referred to as the "parent application," and next level applications are referred to as "child applications." The parent application has awareness of its children and can push and pull data from them. Children are not aware of their parents, or the fact they are children themselves.

MOOSE supports parallel execution for MultiApp multiphysics simulations. The entire simulation is given a set of processors, and each MultiApp branch executes in serial, one level at a time. Therefore, all processors are made available to each MultiApp; meaning that for many types of multi-level simulations, no CPUs are left idle during different stages of execution. However, developers may optionally limit the number of processors used when running on smaller domains or when coupling to legacy codes, which may only support serial execution.

The parent application imposes the maximum time steps used for the entire simulation, and this, as well as mesh globality, is generally the criteria of choice for the parent application. The child applications may use smaller timesteps than the parent application, which is called sub-cycling in this context.

Multiphysics Data Transfers

The hierarchy indicates the flow of information from one application to another as well as dependencies for convergence and time step inheritance. Many types of transfers are available: direct copy (when working on an identical mesh), projection, nearest node, user-object based, field sampling, postprocessor-based, etc. Additionally, many transfers support conserving the quantity of interest being transferred and can be restricted to operate only on a subset of the domain (such as when moving data from one surface to another in a volume).

Until recently, child applications have not been able to communicate directly to each other, instead being required to communicate through the parent application to receive each other's data. This design was intentional to reduce the complexity of assembling MultiApp simulations since it supported the design of not having applications aware of their presence in a hierarchical configuration. It also made it straightforward for the framework to schedule transfers in a typical well-defined traversal of the MultiApp tree. However, there were two significant drawbacks to this design. First, data, otherwise not needed by the parent, still had to pass through the parent to reach a sibling application in a tree. This meant two potentially expensive transfers to get data into its final destination. Additionally, if the parent mesh was unrelated or perhaps coarser than its children, there was a real possibility that significant information could be lost during the child-to-child transfer. To solve both of these issues, data transfer among "siblings" (children of the same parent) has been enabled [11]. In the new design, the parent application still fully manages all transfers among siblings, but data can be moved directly in a single transfer. This has several speed, efficacy, and accuracy implications.

Transferring information between applications that may have different orientations in space (rotation), units (scaling), coordinate system types (Cartesian vs. cylindrical), and translations is also recently supported [11]. The coordinate transformation class is essential for performing reactor multiphysics computations such as coupling three-dimensional Cartesian neutronics calculations with two-dimensional axisymmetric nuclear fuel performance computations. Most transformation information such as units (scaling), physical orientation (rotation), and coordinate system type is encapsulated within each MultiApp's application code or input file, which is consistent with the philosophy of allowing each application to focus on its own domain physics, leaving MOOSE to handle the internals of data transfer transformations.

MULTIAPP SIMULATION OF ADVANCED REACTORS

A selection of MultiApp models of advanced reactors present on the VTB is described in the following sections.

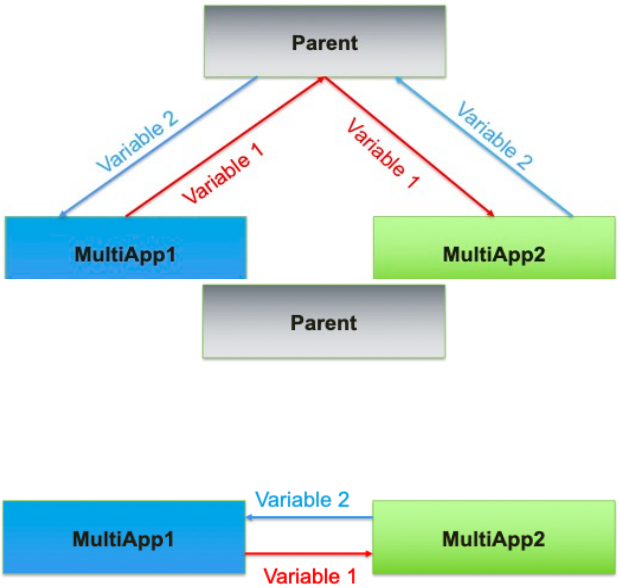

Fig. 1: Examples of coupling between a parent application and two child applications (top) and directly between sibling applications (bottom)

Heat-Pipe-Based Microreactors [12]

The heat-pipe microreactor model uses a two-level MultiApps approach with a Bison parent application to govern the thermal physics within all the solid reactor components except for heat pipes and a Sockeye child app to deal with heat-pipe performance. Development is in progress to add a Griffin neutronics model as the new parent application.

Molten-Salt-Cooled Reactors [13]

The molten-salt fast reactor model adopts a two-level MultiApps structure with Griffin as the parent app and Pronghorn as the child app. The parent app governs neutronics simulation, providing power distribution to the child app. The child app handles fluid dynamics simulation of molten-salt fuels, providing temperature information as feedback to the parent app. Molten-salt reactors are a special class of reactors as the molten salt is both fuel and coolant. As the molten salt flows out of the core region to transfer heat, the delayed neutron precursors concentrations are also calculated and transferred back to Griffin.

Pebble-Bed Gas and Fluoride Salt-Cooled Reactors [14, 15]

Both pebble-bed microreactor and fluoride high-temperature models are available on the VTB using multi-level MultiApps structure. Griffin is the parent application governing neutronics simulations, while Pronghorn solves a homogenized porous media flow problem. A CentroidMultiApp is then used to calculate the representative temperature profile within pebbles and TRISOs (tristructural isotropics). This is an example of

using MultiApps to handle similar physical phenomena (thermal heat transfer here) at different space scales.

Sodium Fast Reactor [16]
The sodium-cooled fast reactor model on the VTB presently has the most complex MultiApp hierarchy. Neutronics (Griffin) is naturally the parent application as it considers the entire system and is not a transient calculation. The radial thermal expansion is simulated by a child application using the Tensor Mechanics module on the support plate with a fixed inlet coolant temperature. The fuel behavior and notably the axial temperature profile are simulated by a Bison child application. The fuel temperature radial boundary condition is calculated by a grandchild SAM coolant channel model. A special approach used in this example is that the axial expansion of the fuel is simulated by another grandchild Bison (i.e., two Bison simulations/MultiApps are involved) application instead of fully coupled with fuel thermal physics.

EXECUTION OF MULTIPHYSICS SIMULATIONS

To launch a MOOSE-based multiphysics simulation example hosted on the VTB site, three steps are needed: (1) clone input files from the VTB open git repository, (2) obtain relevant code licenses from the appropriate authority (contact Idaho National Laboratory's (INL's) Nuclear Computational Resource Center (NCRC) [17]), and (3) utilize pre-compiled binaries on the INL High Performance Computing (HPC) system through job script submission or the NEAMS Workbench [18]. As an alternative to step (3), the user can download binary files on their own system using conda commands [19] or compile binaries from source.

The INL HPC system uses pre-built NEAMS code binaries which can be accessed by licensed users. The BlueCRAB application is a "super-application" containing most of the codes of interest needed to run VTB examples (the main exception being Cardinal). Otherwise, dynamic linking options in the multiphysics input files can be used to invoke connections between different applications.

Alternatively, the user can use the `conda install Bison` command to install local binaries on their own system or compile binaries from source if they have that level of access.

Continuous Integration Testing

Continuous integration testing of VTB multiphysics inputs against current code versions ensures the inputs are kept up to date and functioning correctly. The testing is performed both on every update to every MOOSE-based code and code-package and on every update to the VTB models. This makes sure that regressions in each model are caught during the development cycle of the codes and not later during a release. The testing is performed in several layers. First a syntax check makes sure the input file syntax is not deprecated. Then each individual input file is ran separately using the relevant application. The outputs of a few relevant key metrics are compared to a "gold" file for each input file. Finally, the multiphysics models are ran by combined applications such as BlueCRAB and tested for regression on the key metrics in the context of the coupled calculation. This layered workflow ensures regressions are caught as early and as clearly as possible.

SUMMARY

MOOSE-based (or MOOSE-wrapped) physics applications such as Griffin, Pronghorn, and Bison may be assembled into a customized multiphysics workflow for reactor analysis using the MultiApp and Transfer systems within MOOSE. The plug-and-play capabilities of MOOSE's

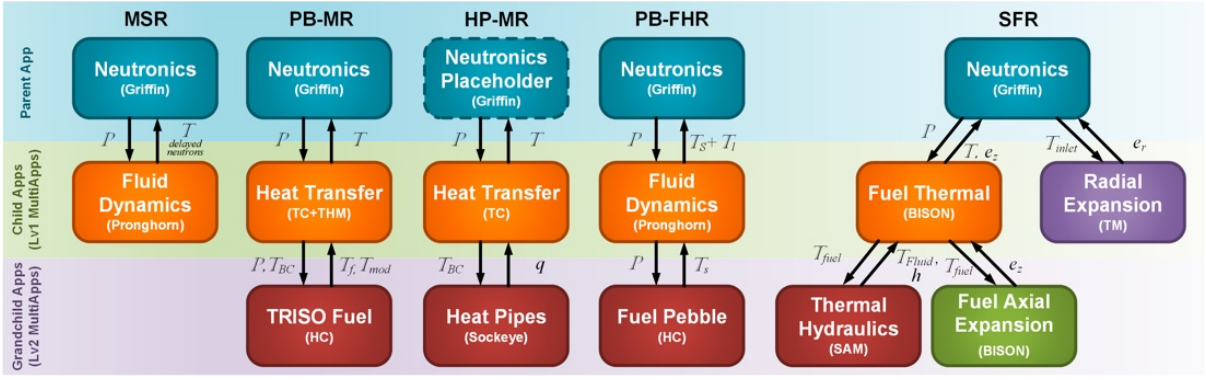

Fig. 2: Example of MultiApp reactor model on the VTB (Virtual Test Bed)

'sis codes to be leveraged for multiple reactor types while tailoring multiphysics coupling and workflows to each reactor type and event using input file syntax only. This plug-and-play paradigm allows rapid development of multiphysics modeling capabilities for a diverse set of problems by non-expert users.

New features include the ability for MOOSE to transfer data between "sibling" applications, rather than just between parent and child, and new syntax to facilitate data transfer between codes using different coordinate systems. The VTB hosts numerous nuclear reactor multiphysics examples using the MOOSE capabilities described here. Documentation on multiphysics coupling for nuclear reactors using MOOSE is also available on the VTB. Continuous integration testing is performed on VTB code inputs to ensure they are kept up to date and functional with most recent code versions.

ACKNOWLEDGEMENTS

Argonne National Laboratory's work was supported by the U.S. Department of Energy (DOE), Office of Nuclear Energy, Advanced Modeling and Simulation Program (NEAMS) under contract DE-AC02-06CH11357. We also thank the National Reactor Innovation Center program for their support for the Virtual Test Bed and targeted model development. This manuscript was also authored by Battelle Energy Alliance, LLC under contract no.~DE-AC07-05ID14517 with the U.S. Department of Energy . The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license to said article, enabling it to reproduce the article, prepare derivative works, distribute copies to the public, and publicly perform or display portions thereof, by or on behalf of the U.S. Government. DOE will provide public access to these results of federally sponsored research, in accordance with the DOE Public Access Plan: http://energy.gov/downloads/doe-public-access-plan.

REFERENCES

1. C.J. PERMANN, et al, "MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, **11**, 100430 (2020) https://doi.org/10.1016/j.softx.2020.100430.
2. C. STANEK, "Overview of the DOE-NE NEAMS Program," LA-UR-19-22247, Los Alamos National Laboratory, (2019).
3. *National Reactor Innovation Center Virtual Test Bed Website*, https://mooseframework.inl.gov/vtb/.
4. C. H. LEE, J. ORTENSI, et al., "Griffin Software Development Plan," ANL/NSE-21/23, INL/EXT-21-63185, Argonne National Laboratory and Idaho National Laboratory (2021).
5. A.J. NOVAK, et al, "Pronghorn: A Multidimensional Coarse-Mesh Application for Advanced Reactor Thermal Hydraulics," *Nuclear Technology,* **7** (2021) https://doi.org/10.1080/00295450.2020.1825307.
6. R.L. WILLIAMSON, et al, "BISON: A Flexible Code for Advanced Simulation of the Performance of Multiple Nuclear Fuel Forms," *Nuclear Technology*, **0,** 1 –27 (2021) https://doi.org/10.1080/00295450.2020.1836940.
7. R. HU, "SAM Theory Manual," ANL/NE-17/4 Rev. 1, Argonne National Laboratory (2021).
8. E. HANSEL, R. Berry, D. ANDRS, M. KUNICK, R. MARTINEAU, "Sockeye: A One-Dimensional, Two-Phase, Compressible Flow Heat Pipe Application", *Nuclear Technology,* **207**, number = {7}, 1096-1117, (2021), https://doi.org/10.1080/00295450.2020.1861879}
9. E. MERZARI, et al, "Cardinal: A Lower Length-Scale Multiphysics Simulator for Pebble-Bed Reactors," *Nuclear Technology,* **7** (2021), https://doi.org/10.1080/00295450.2020.1824471.
10. *MOOSE MultiApps System Documentation*, https://mooseframework.inl.gov/syntax/MultiApps/index.html.
11. A.D. LINDSAY, et al. "User-oriented Improvements in the MOOSE Framework in Support of Multiphysics Simulation," INL/RPT-22-67144, Idaho National Laboratory (2022).
12. N.E. STAUFF, et al, "Preliminary Applications of NEAMS Codes for Multiphysics Modeling of a Heat Pipe Microreactor," *Trans. Am. Nucl. Soc.*, **124** (2021).
13. A. ABOU-JAOUDE, et al, "A Workflow Leveraging MOOSE Transient Multiphysics Simulations to Evaluate the Impact of Thermophysical Property Uncertainties on Molten-Salt Reactors," *Annals of Nuclear Energy*, **163 (**2021), https://doi.org/10.1016/j.anucene.2021.108546.
14. P. BALESTRA, et al, "PBMR-400 Benchmark Solution of Exercise 1 and 2 Using the Moose-Based Applications: MAMMOTH, Pronghorn," *EPJ Web of Conferences*, (2021), https://doi.org/10.1051/epjconf/202124706020.
15. G. GUIDICELLI, et al, "Coupled Multiphysics Multiscale Transient Simulations of The Mk1-Fhr Reactor Using Finite Volume Capabilities of The Moose Framework," *Proc. Int. Conf. of Math and Comp. for Nucl. Sci. and Eng.* (2021).
16. N. MARTIN, R. STEWART, S. BAYS, "A multiphysics model of the versatile test reactor based on the MOOSE framework", *Annals of Nuclear Energy*, **172**, (2022), https://doi.org/10.1016/j.anucene.2022.109066.
17. *Idaho National Laboratory Nuclear Computational Resource Center*, https://inl.gov/ncrc/.
18. B. R. LANGLEY, A. R. LEFEBVRE, "Workbench Analysis Sequence Processor and {MOOSE} Framework Parser Benchmarks," ORNL/LTR-2022/16, Oak Ridge National Laboratory (2021).
19. Conda-Forge Community, "The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem," Zenodo (2015) http://doi.org/10.5281/zenodo.4774216.