# AN OPEN-SOURCE DATA-CENTRIC REPRESENTATION OF THE FAST FLUX TEST FACILITY ISOTHERMAL BENCHMARK CORE

J. HADER, M. YOUNG1, N. TOURAN1

1TerraPower, LLC, Bellevue WA,USA

Email contact of corresponding author: ntouran@terrapower.com

Historical data describing the measured characteristics of past nuclear reactors have long proven invaluable for methods development and validation efforts. Coordination and collaboration in international benchmark activities has flourished and produced essential resources such as the International Reactor Physics Experiment Evaluation (IRPhE) Handbook [1]. We have taken an exploratory new step by converting one detailed document-centric description, that of the Fast Flux Test Facility (FFTF) isothermal benchmark [2], into an open-source data-centric computer model and published it on GitHub [3]. We believe that the translation of the core model from the document to the open data representation can improve the efficiency, quality, and impact of benchmarking activities around the world.

The model exists in the form of input files to TerraPower’s open-source Advanced Reactor Modeling Interface (ARMI®) framework, which is itself published on GitHub [4]. Several representations of this benchmark representation are shown in Fig 1.

A picture containing diagram

Description automatically generated

*FIG. 1. The FFTF core model defined by the ARMI input file, in the ARMI grid editor, and as visualized from an ARMI-generated output file in ParaView.*

Users can access the full detail of the model from an ergonomic Python API to generate input files directly for their target analysis tool(s), read in the resulting calculation output, and couple or compare with any other tools that interface with the ARMI system. Users can also leverage the included ARMI framework tools to perform many other common tasks, such as thermal expansion, material homogenization, mesh mapping, and generation of 3D visualizations of the core.

The ARMI FFTF model includes pin-level descriptions of the fuel and control assemblies. Control assemblies are placed in their critical positions, as shown in Fig 2. Users interested in pin-level detailed models can access the pin-level data directly, while users who will only make use of the homogenized assembly descriptions can use the ARMI API to retrieve state information (e.g. nuclide compositions) at the block-homogenized level.

Graphical user interface

Description automatically generated

*FIG. 2. The critical control rod positions via Boron-10 number density (left) and initial heavy metal mass (right) of the FFTF model viewed through an ARMI-generated output file in ParaView.*

Users can employ the existing physics integrations to compute aspects of the benchmark with ease. For example, the FFTF input files can be used directly with the existing open-source plugin [6] to the DRAGON lattice-physics code [7] to generate multigroup cross sections in ISOTXS format. In cases where a user is focused on studying a neutron transport method, this offers a convenient, efficient, and collaborative way to produce meaningful cross sections for the problem.

As an example, the DRAGON code and its associated plugin were used to generate cross sections in collaboration with the Idaho National Laboratory during a contract sponsored by Battelle Energy Alliance in support of the Versatile Test Reactor (VTR). This work involved the creation of an open-source ARMI-DIF3D plugin [5] supporting the DIF3D neutronics code [8]. Note that any user of this plugin must possess a licensed copy of the DIF3D code itself.

Chart, bar chart, box and whisker chart

Description automatically generated

*FIG. 3. A slice of the FFTF flux distribution as computed (left) and as automatically remapped to the ARMI mesh (right) with the DRAGON and DIF3D codes and their associated open-source ARMI plugins and visualized from an ARMI output file in ParaView (reproduced from [5]).*

**BENEFITS OF THE OPEN-SOURCE APPROACH**

We have described several ways that efficiency can be improved through the philosophy of building open data-centric benchmark models. New groups can more rapidly and confidently stand on the shoulders of their colleagues without the need to repeat mundane and error-prone work that has already been done (often multiple times). As more teams share capabilities integrated into a central data model, potential “button-press” tool combinations and comparisons could grow rapidly, allowing more complete evaluations without significantly increased effort.

The quality of benchmarking activities can be improved with the open-source philosophy as well. If multiple parties are investigating the same open data representation of the FFTF core, transcription mistakes and lessons learned can be shared across the community. This is especially impactful when investigating uncommon data that has not benefited from significant scrutiny in recent years.

A perhaps more subtle advantage of the open-source philosophy is the potential impact and reach of the work. When a laborious endeavour is more readily accessible by more people, there is an inherently higher likelihood that more broad and positive use will be made of the output. This advantage is of interest to funding organizations who want their limited funds to produce the most positive change in the world. It’s also an advantage to the engineers doing the work. The near infinite potential reach is intriguing and often inspires workers to take extra care and add extra polish.

A long and proud tradition of sharing and collaboration in the nuclear industry dates back at least to the 1953 Atoms for Peace program. Open-sourcing transformations of already-public benchmark data for R&D purposes is particularly conducive as a first effort for any institution. Such efforts can help acquaint legal staff with the particular issues and concerns related to open source in nuclear.

As for licensing, we have found that permissive open-source licenses such as the MIT or Apache licenses are more likely to be readily used by other institutions as opposed to more opinionated, copy-left licenses, such as the GPL. Effectively all nuclear institutions have significant amounts of code that they will not ever choose open source, and so building atop any GPL code is often considered a liability.

**OVERVIEW OF THE ARMI FRAMEWORK**

The potential benefits of the FFTF model described above depend to a degree on the use of the underlying open-source ARMI framework. Thus, this section provides a brief description and overview of the ARMI framework itself.

The framework provides a central place for all physics kernels to interact: the Reactor Model. All modules read state information from this Reactor and write their output to it in a hub-and-spoke data model. This common interface allows seamless communication and coupling between different physics sub-specialties. If one adds a new physics kernel into ARMI, it becomes coupled to “N” other kernels.

The ARMI approach was born out of the question of how to best leverage an eclectic mix of legacy and modern tools with a small team to do full-scope analyses. We built the environment to automate the tedious, uncoupled, and error-prone parts of reactor engineering/analysis work. For example, it allows us to dispatch hundreds of parameter sweeps to multiple machines and then perform multi-objective optimization on the resulting design space.

ARMI-based applications can quickly and easily produce complex input files with high levels of detail in various approximations. This enables users to perform rapid, high-fidelity analyses to ensure all important physics are captured. It also enables sensitivity studies of different modeling approximations (e.g. symmetries, transport vs. diffusion vs. Monte Carlo, subchannel vs. CFD, etc.).

The ARMI framework is largely written in the Python programming language. Its high-level nature allows nuclear and mechanical engineers to rapidly automate their analysis tasks from their sub-specialties. This helps eliminate the translation step between computer scientists and power plant design engineers. This allows good division of labor; the computer scientists can focus on the overall performance and maintainability of the framework, while the power plant engineers focus on power plant engineering.

The ARMI framework was originally created by TerraPower®, LLC near Seattle, WA starting in 2009. Its founding mission was to determine the optimal fuel management operations required to transition a fresh Traveling Wave Reactor (TWR®) core from startup into an equilibrium state. It started out automating the Argonne National Lab (ANL) fast reactor neutronics codes, MC2 and REBUS. The reactor model design was made with the intention of adding other physics capabilities later. Soon, simple thermal hydraulics were added and it's grown ever since. It has continuously evolved towards a general reactor analysis framework.

Following requests by outside parties, we began working on a more modular architecture that would allow some of the intertwined physics capabilities to be separated out as plugins from the standalone framework.

Similar to how open-source benchmarking as described above can be beneficial, we hypothesize that collaborating on reactor software systems themselves can help align efforts worldwide, increasing efficiency, quality, and impact. Thus, the ARMI framework was released under an open-source license in 2019 to facilitate mutually beneficial collaboration across the nuclear industry, where many teams are independently developing similar reactor analysis/automation frameworks and evaluating similar benchmark problems with similar physics codes.

We hope that others will build data-centric benchmark models using the ARMI framework and related ecosystem.

References

1. BESS, J., International Reactor Physics Experiment Evaluation (IRPhE) Project. IRPhE Handbook-2015 edition. No. NEA--1765-11. Organisation for Economic Co-Operation and Development, 2015.
2. BESS, J., Evaluation of the Initial Isothermal Physics Measurements at the Fast Flux Test Facility, a Prototypic Liquid Metal Fast Breeder Reactor, INL/EXT-09-16524 (2010) <https://digital.library.unt.edu/ark:/67531/metadc1013078/>
3. HADER, J., ARMI FFTF Model (2020), <https://github.com/terrapower/fftf-isothermal-model>
4. TOURAN, N., Advanced Reactor Modeling Interface (2021),

<https://github.com/terrapower/armi>

1. YOUNG, M., DIF3D plugin to the ARMI nuclear reactor analysis framework, <https://github.com/terrapower/armicontrib-dif3d>
2. ONUFER, M., DRAGON plugin to ARMI, <https://github.com/terrapower/dragon-armi-plugin>
3. HÉBERT, A., "DRAGON5: Designing Computational Schemes Dedicated to Fission Nuclear Reactors for Space", paper presented at the Int. Conf. on Nuclear and Emerging Technologies for Space, February 25 - 28, Albuquerque, NM (2013).
4. SMITH, M., DIF3D-VARIANT 11.0: a decade of updates. No. ANL/NE-14/1. Argonne National Lab., Argonne, IL (United States), 2014.