M&C 2023 - The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering · Niagara Falls, Ontario, Canada · August 13 – 17, 2023

Verification of the Cardinal Multiphysics Solver for 1-D Coupled Heat Transfer and Neutron Transport

L.I. Gross[1], A.J. Novak[2], P.Shriwise[2], and P.P.H. Wilson[1]
[1]University of Wisconsin – Madison
1500 Engineering Drive, Madison, WI 53706

[2]Argonne National Laboratory
9700 S Cass Avenue, Lemont, IL 60439

ligross@wisc.edu, anovak@anl.gov, pshriwise@anl.gov paul.wilson@wisc.edu

ABSTRACT

Cardinal is a multiphysics software tool that couples OpenMC Monte Carlo transport and NekRS Computational Fluid Dynamics (CFD) to the Multiphysics Object-Oriented Simulation Environment (MOOSE). In this work, we verify Cardinal for coupled heat conduction and neutron transport using a 1-D analytic solution from previous work by the Naval Nuclear Laboratories. This numerical benchmark includes S_2 transport, Doppler-broadened cross-sections, thermal conduction and expansion, and convective boundary conditions. The goals of this work are to verify Cardinal's basic multiphysics modeling capabilities for coupled neutronics and heat conduction, while serving as a computational "test bed" for assessing the performance of various multiphysics coupling strategies and algorithms. The key numerical metrics to match from the analytical benchmark are the temperature and flux distributions, as well as the eigenvalue. Using the analytical solution for temperature and flux, an L_2 error norm was computed for each mesh size. **INSERT CONVERGENCE ONCE DECISION MADE ABOUT LAST FLUX** The ratio of the Computed to Expected Ratio (C/E) temperature and flux in each mesh element shows agreement as the C/E for each mesh approaches the ideal $y = 1$ as the mesh size increases. Both plots show that with increasing mesh elements, the C/E approaches the ideal $y = 1$. The eigenvalue k_{eff} also agreed well with the benchmark's 0.29557 for each mesh size. These correct computation of these key metrics demonstrates confidence in the fidelity of Cardinal's multiphysics modeling capabilities. **TODO COMMENT ON FLUX.**

KEYWORDS: Cardinal, MOOSE, OpenMC, multiphysics, verification

1. INTRODUCTION

With recent advancements in methods, software, and computing, high-fidelity multiphysics Modeling and Simulation (M&S) is becoming an important component of the nuclear engineer's "toolbox." These high-fidelity models substitute excessively pessimistic safety factors with predictive science. This can reduce uncertainty in analyses, enabling tighter margins to realize improved economics and licensing certainty. However, analytical benchmarks and comparison to experimental data are required to assess the stability, convergence, and predictive capability of these high-fidelity models for reactor design and analysis.

Cardinal [1] is an open-source code that couples OpenMC [2] Monte Carlo particle transport and NekRS CFD to MOOSE [3]. This coupling brings high-fidelity multiphysics feedback to the MOOSE "ecosystem." Cardinal couples OpenMC and NekRS to MOOSE simulations by copying data between the internal code data structures (e.g. a vector of tally results in OpenMC) and a `MooseMesh`, or the unstructured mesh class in MOOSE. MOOSE's mesh-to-mesh interpolation system then communicates between the `MooseMesh` "mirror" of the external code's solution and an arbitrary coupled MOOSE application in the form of boundary conditions (such as for conjugate heat transfer with NekRS) or source terms (such as for volumetric heating with OpenMC). Convergence is obtained with Picard iteration.

For coupled neutronics-thermal-fluid simulations with OpenMC, each Picard iteration consists of several steps: 1) a MOOSE application (e.g. BISON, Pronghorn, NekRS via Cardinal, ...) solves for temperatures and densities; 2) Cardinal transfers temperatures and densities to the OpenMC model; 3) OpenMC solves for the nuclear heating; and 4) Cardinal transfers the tally values to the `MooseMesh` "mirror."

These steps continue until convergence criteria is achieved. In this work, we pursue verification of these multiphysics aspects of Cardinal using a 1-D analytical benchmark from the Naval Nuclear Laboratories [4]. This work does not require CFD, and thus NekRS will be left out of discussion from this point on.

The remainder of this paper is organized as follows. In Section 2, we summarize the analytical benchmark modeled in this work. Section 3 then describes the Cardinal computational model of the benchmark. Section 4 presents comparisons between Cardinal and the analytical benchmark. Finally, Section 5 presents conclusions and outlines ongoing and future efforts in the verification and validation of Cardinal.

2. BENCHMARK PROBLEM DESCRIPTION

The analytical benchmark couple between three physics: S_2 neutron transport with Doppler broadening, heat conduction, and thermal expansion. S_2 transport restricts the neutron direction to only the $\pm x$ direction. A summary of the governing Ordinary Differential Equations (ODEs) and boundary conditions in the 1-D slab is shown in Fig. 1.

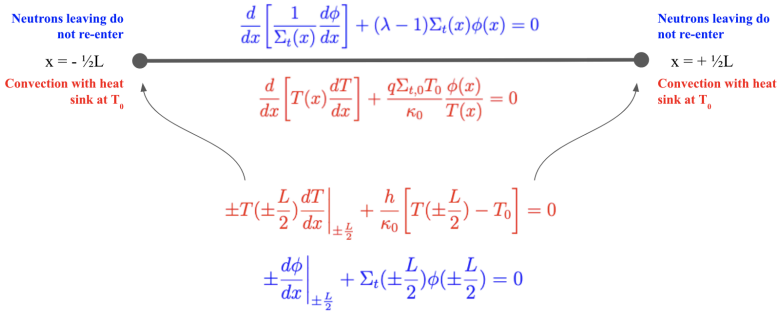


Figure 1: This diagram summarizes the domain, ODEs, and boundary conditions for the slab.

This benchmark uses a one-group assumption for the neutron cross-sections. As neutrons transport, heat from fission events deposit volumetric power in the slab, causing thermal expansion and affecting the temperature distribution; thermal expansion is restricted to only the x-direction. This slab elongation feeds back neutronics and heat conduction by influencing the domain length and material density. The slab has convective boundary conditions at the endpoints $x = \pm\frac{L}{2}$ with the heat sink temperature T_0. The Doppler-broadened total, microscopic cross-section follows an inverse-root temperature relationship,

$$\sigma_t(T) = \sigma_{t,0}\sqrt{\frac{T_0}{T(x)}} \tag{1}$$

Due to thermal expansion, the slab density varies as

$$\rho(x) = \rho_0\sqrt{\frac{T_0}{T(x)}}, \tag{2}$$

This gives a Doppler-broadened, macroscopic, total cross-section that accounts for changes in density due to temperature as

$$\Sigma_t(x) = \frac{\rho_0\sigma_{t,0}N_A}{A}\frac{T_0}{T(x)} = \Sigma_{t,0}\frac{T_0}{T(x)}, \tag{3}$$

where $\sigma_{t,0}$ is the total, microscopic cross-section at T_0, N_A is Avogadro's number, and A is the mass number of the medium. The conduction equation governs heat flow in the slab and can be described in terms of the

thermal conductivity $\kappa(T)$, the energy released per fission q, the total, macroscopic cross-section Σ_t, and the neutron flux ϕ.

$$\frac{d}{dx}\left[\kappa(T(x))\frac{dT(x)}{dx}\right] + q\Sigma_t(x)\phi(x) = 0. \tag{4}$$

The main assumption used to craft the analytical solution is that $T(x) = f\phi(x)$. This assumption is guaranteed by manipulating the ODEs, inserting the cross-section temperature dependence, and matching coefficients so that the two ODEs are of the same solution type. The matching of coefficients imposes two constraints that give equations for the total, microscopic cross-section $\sigma_{t,0}$ and the heat transfer coefficient h, in terms of the slab parameters. These are two key input parameters required for the simulation. For full details on the analytical solution's derivation, see [4].

3. COMPUTATIONAL MODEL

This section describes the OpenMC and MOOSE computational models, followed by the convergence criteria used for coupling. Note that Cardinal currently does not yet support moving geometries in OpenMC. Instead of using thermo-mechanics for thermal expansion, the simulation uses the formula for equilibrium length L from [4] that accounts for all three physics effects. The change in density due to temperature is accounted for in this simulation by using the total, macroscopic cross-section – which absorbs density to become a $\frac{1}{T}$ dependence – from (3), as was mentioned in Section 2.

3.1. OpenMC Model

The geometry used in OpenMC must be finite, despite the benchmark being infinite in the y and z directions. In general, reflective boundary conditions can be used to simulate infinite dimensions. Thus, this benchmark can be represented with vacuum boundary conditions at $x = \pm\frac{L}{2}$ and reflective boundary conditions at the y and z boundaries. Note that particles only move in the $\pm x$ direction, so no y or z boundaries are crossed, but with non-S_2 scattering, the reflective boundaries would come into play. In addition to being finite for the model, the y and z dimensions need to be set to 1 cm in order for Equation (19) from the benchmark to uphold its one dimensional power integral, i.e. that integration in y and z is implied to contribute a factor of 1. [4] Fig. 1 shows a diagram of the $1D$ geometry, governing equations, and boundary conditions for the different physics.

The benchmark's one-group assumption was satisfied using OpenMC's multigroup mode. Since the benchmark uses a fictitious material with a known function for the temperature dependence, the simulation took advantage of OpenMC's capability for user-defined cross-sections via the `XSdata` class. The cross-section for each reaction was specified for 50 evenly spaced temperatures between 308 K and 358 K and was exported to a library that OpenMC used to determine the appropriate cross-section in each region as temperature changed via input from Cardinal from iteration to iteration.

The OpenMC model required slight source code modifications to accommodate S_2-like transport. In typical OpenMC simulations, the physics of each reaction describes the scattering dynamics. The Monte Carlo algorithm is agnostic to the direction particles move, but it is not typically constrained to a discrete directional distribution. However, when modeling this benchmark, any history with a particle moving perpendicular to the x-direction would attenuate particles in less x-distance than if particles were constrained to either $\pm x$. To address this, we first use OpenMC's `PolarAzimuthal` distribution to restrict the birth direction of particles to the $\pm x$ direction. We then also modified OpenMC in a patched branch to mimic S_2 transport in two ways. First, when determining the angular cosine of scattering events, μ, particles either continue forward ($\mu = 1$) or are back-scattered ($\mu = -1$) with equal probability, as opposed to sampling the reaction physics for μ. Second, since the simulation uses k-eigenvalue mode, neutrons born in subsequent generations of the simulation also need to have their angular birth distributions restricted to $\pm x$.

The k-eigenvalue simulation used 50,000 particles per batch, with 50 inactive batches and 50 active batches for every case. A few tallies were used in order to compare to the analytical solution: a flux cell tally, a kappa-fission rate cell and global tally, and the eigenvalue k_{eff}. Though flux and the eigenvalue are the only quantities to compare with the benchmark, these other tallies are needed in order to compute a source

strength. OpenMC reports flux, denote it $\hat{\phi}$, in units of $\frac{n-cm}{sp}$, but typical flux, denote it ϕ, has units of $\frac{n}{cm^2-s}$. In fixed source modes, the source strength is typically known, but in eigenvalue mode, it must be computed, as it depends on the physical source and fission source. The following was used to compute the source strength

$$ss = \frac{P}{\kappa V_{voxel}} \qquad \text{THUS} \qquad \phi = \frac{P}{\kappa V_{voxel}}\hat{\phi}, \tag{5}$$

where $P = 1e22\frac{ev}{s}$ is the slab integrated power from the benchmark, κ is the tallied kappa-fission rate across the slab, and V_{voxel} is the volume of a voxel in the mesh. The same source strength could be used to convert the kappa-fission tally from OpenMC, which has units of $\frac{ev}{sp}$, to more physically meaningful units of $\frac{ev}{cm^3-s}$.

3.2. MOOSE Heat Conduction Model

The MOOSE Heat Conduction Module (HCM) is used to solve for the temperature distribution within the slab. It solves the conduction equation using the Finite Element Method (FEM) available in MOOSE:

$$-\nabla \cdot (k_s(\mathbf{r}, T_s)\nabla T_s(\mathbf{r})) = \dot{q}_s, \tag{6}$$

where k_s is the thermal conductivity in the solid and $\dot{q}_s$ is the heat source, in this case from fission. It uses a `HeatConductionMaterial` to specify that the thermal conductivity obeys the benchmark's temperature dependence $\kappa(T) = \kappa_0 T$. The boundary conditions used by MOOSE match temperature boundary condition in red shown in Fig (1). The HCM is coupled to OpenMC by receiving the heat source in each element from the kappa-fission tally. During each constituent iteration, it recomputes the temperature distribution from the heat source and boundary conditions, and then sends the temperatures back to OpenMC for the next transport solve.

3.3. Convergence Criteria

The HCM uses iterative methods to compute temperature. It uses a combination of linear and non-linear iterations to update the solution, stopping once both of the convergence criteria are met. This study used convergence criteria of 10^{-7} for absolute tolerance and 10^{-9} for relative tolerance. The following petsc options were included:

```
petsc_options_iname='-pc_type -pc_hypre_type'
petsc_options_value='hypre boomeramg'
```

Monte Carlo is not an iterative method, so convergence is achieved by using an appropriate number of batches and histories per batch. The configurations used here were mentioned in Section 3.1.

In terms of converging global iterations across all single physics, each mesh cases used 200 Picard Iterations. To assist with convergence of Monte Carlo quantities, Robbins-Monro relaxation was applied to the flux and kappa-fission tallies. This updates the quantities of interest for the $n+1$th iteration as an average over the newest solution and the n previous iterations. [5] So the flux at Picard Iteration $n+1$ would be given by

$$\phi_{n+1} = \frac{1}{n+1}\sum_{i=0}^{n}\phi_i, \tag{7}$$

where ϕ_i is the flux output from the ith Monte Carlo solve. In order to assess the convergence of each physics, plots of error versus mesh element size were used and will be shown in Section 4.

4. RESULTS

The results to compare with the benchmark are the temperature distribution, flux distribution, and eigenvalue. The benchmark provides analytical solutions for all of these quantities. An example temperature distribution is shown in the figure below for the 50 mesh element case.

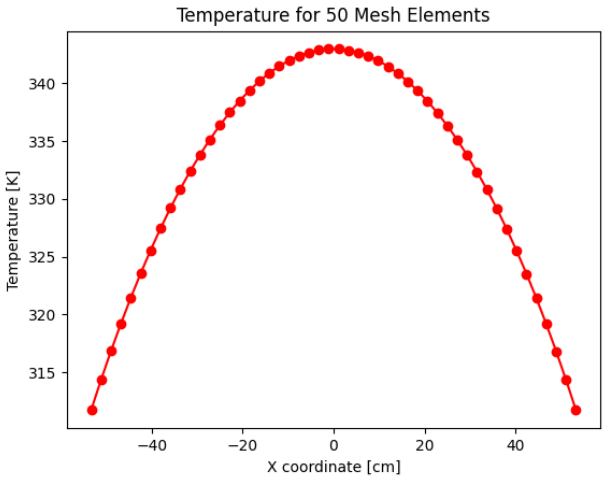


Figure 2: This plot shows the computed temperature for 50 mesh elements.

In order to assess the accuracy of the temperature solution and verify that refining the mesh increases the accuracy, the analytical solution was used to compute an L_2 error norm for temperature. The error computed, ε_T, is given by

$$\varepsilon_T = \frac{||T_a - T_x||_2}{||T_a||_2}, \tag{8}$$

where T_a is the analytical solution evaluated at the x-centroid of the mesh element and T_x is the temperature computed for that voxel from the multiphysics simulation. The figure below shows how the temperature error relates to the number of mesh elements

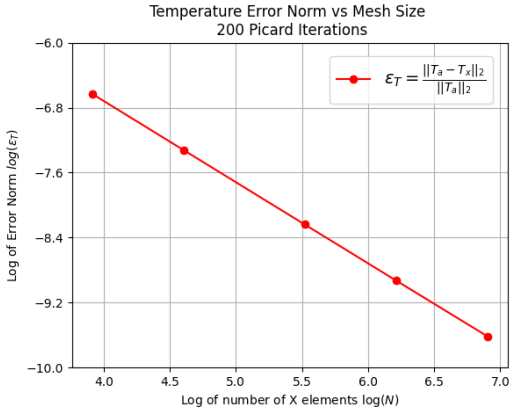


Figure 3: This plot shows log of the number of mesh elements vs the log of the L_2 error norm for the computed vs analytical temperature.

Linear convergence is achieved, as the slope of the line on the $\log(N)$ vs $\log(\varepsilon_T)$ plot is -0.99924828. The same convergence test was used on the flux solution as well with the flux error norm, ε_ϕ, defined as

$$\varepsilon_\phi = \frac{||\phi_a - \phi_x||_2}{||\phi_a||_2}. \tag{9}$$

The figure below shows how the flux error relates to the number of mesh elements

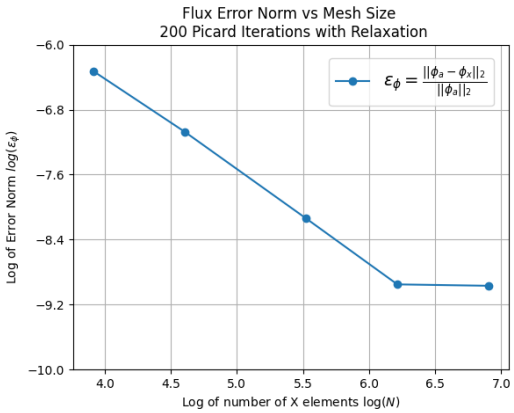
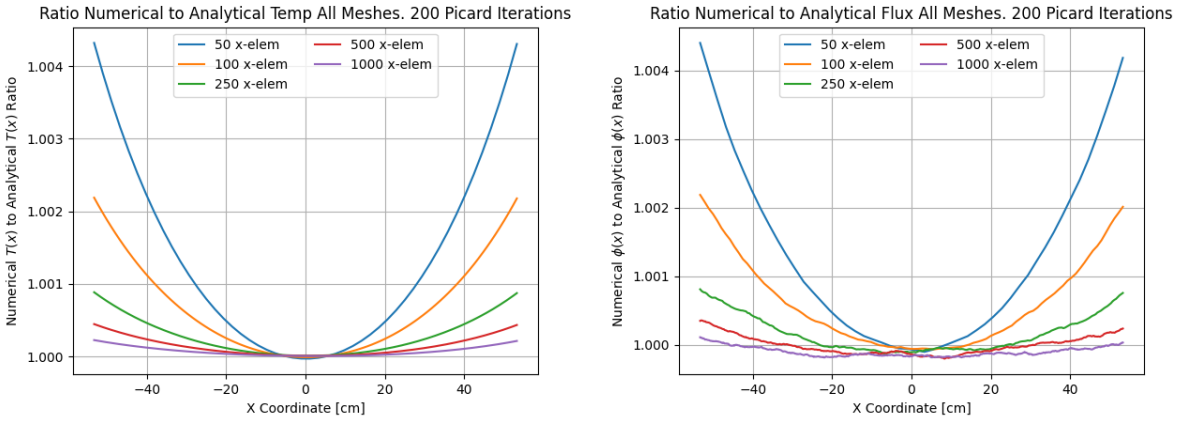


Figure 4: This plot shows log of the number of mesh elements vs the log of the L_2 error norm for the computed vs analytical flux.

[CONVERGENCE] , as the slope of the line on the $\log(N)$ vs $\log(\varepsilon_\phi)$ plot is **TODO SLIGHT PROBLEM W FLUX LAST BIN CONVERGENCE**. Another measure of comparison is the C/E for each mesh size. The ideal would be exact equality, or a ratio of 1. The temperature and flux C/Es are shown in the figure below.



Lastly, the eigenvalue for each mesh is reported in the table below.

Table 1: Eigenvalue with uncertainty for each mesh size. TODO update and get the newest ones

mesh elements	k_{eff}	pcm diff
50	0.29555 ± 0.00009	9
100	0.29551 ± 0.00011	6
250	0.29585 ± 0.00012	-28
500	0.29574 ± 0.00013	-17
1000	0.29551 ± 0.00013	6

5. CONCLUSIONS

Overall, the numerical results show strong agreement with the analytical solutions. The convergence is first order for the temperature and **flux**.The stochastic nature of the Monte Carlo algorithm is likely a factor

for the convergence of the flux. Since zeroth order shape functions are used in the FEM, the first order temperature spatial convergence is to be expected. [6] The C/E plots also show expected behavior. As the mesh fineness increases, the computed to expected ratio gets closer and closer to the ideal $y = 1$. The eigenvalue agrees very well for all cases and appears to be independent of the mesh size.

While code to code benchmarking is common in for nuclear M&S, agreement with analytical benchmarks greatly increases the confidence in Cardinal's multiphysics coupling. Though typical industry-grade simulations would not run S_2 transport, this modification allows Cardinal to show it can match against theoretical expectations.

future work to mention?

NOMENCLATURE

Nomenclature

C/E Computed to Expected Ratio

CFD Computational Fluid Dynamics

FEM Finite Element Method

HCM Heat Conduction Module

M&S Modeling and Simulation

MOOSE Multiphysics Object-Oriented Simulation Environment

ODE Ordinary Differential Equation

ACKNOWLEDGEMENTS

The authors would like to thank the OpenMC and MOOSE development teams for their guidance in model setup and assistance with software. We also want to thank Dr. Griesheimer and Dr. Kooreman for their modeling advice and knowledge about the analytical benchmark.

REFERENCES

[1] A. Novak, D. Andrs, P. Shriwise, J. Fang, H. Yuan, D. Shaver, E. Merzari, P. Romano, and R. Martineau. "Coupled Monte Carlo and Thermal-Fluid Modeling of High Temperature Gas Reactors Using Cardinal." *Annals of Nuclear Energy*, **volume 177**, p. 109310 (2022).

[2] P. Romano, N. Horelik, B. Herman, A. Nelson, B. Forget, and K. Smith. "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development." *Annals of Nuclear Energy*, **volume 82**, pp. 90–97 (2015).

[3] A. D. Lindsay, D. R. Gaston, C. J. Permann, J. M. Miller, D. Andrš, A. E. Slaughter, F. Kong, J. Hansel, R. W. Carlsen, C. Icenhour, L. Harbour, G. L. Giudicelli, R. H. Stogner, P. German, J. Badger, S. Biswas, L. Chapuis, C. Green, J. Hales, T. Hu, W. Jiang, Y. S. Jung, C. Matthews, Y. Miao, A. Novak, J. W. Peterson, Z. M. Prince, A. Rovinelli, S. Schunert, D. Schwen, B. W. Spencer, S. Veeraraghavan, A. Recuero, D. Yushu, Y. Wang, A. Wilkins, and C. Wong. "2.0 - MOOSE: Enabling massively parallel multiphysics simulation." *SoftwareX*, **volume 20**, p. 101202 (2022).

[4] D. P. Griesheimer and G. Kooreman. "Analytical Benchmark Solution for 1-D Neutron Transport coupled with Thermal Conduction and Material Expansion." p. 10. Pittsburgh, Pennsylvania. (2022).

[5] A. Novak and P. Shriwise. "OpenMCCellAverageProblem." URL https://cardinal.cels.anl.gov/source/problems/OpenMCCellAverageProblem.html.

[6] INL. "FEM Convergence." URL https://mooseframework.inl.gov/getting_started/examples_and_tutorials/tutorial03_verification/step02_fem_convergence.html.