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 Laboratory. 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. The benchmark provides analytical solutions for the temperature and flux distributions, as well as the k-eigenvalue. Using these solutions, an L_2 error norm was computed for each spatial discretization (finite element heat conduction mesh, Monte Carlo cells). The temperature error showed linear convergence on a log-log plot of error vs. mesh element number, with a slope of -0.9982 ($R^2 \approx 1.0$). All flux predictions (as a function of space) were within 2σ of the analytic solution, for Monte Carlo cell counts between 50 and 1000. The eigenvalue k_{eff} also agrees well with the benchmark value for each mesh size. The outcome of this work is verification of coupled Monte Carlo-thermal conduction modeling using Cardinal.

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 more conservative safety factors with physics-based simulation. 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 accuracy of these high-fidelity models for reactor design and analysis.

Cardinal [*] is an open-source code [1,2] that wraps OpenMC [3] Monte Carlo particle transport and NekRS [4] CFD within the MOOSE [5] framework. 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.

[*]https://cardinal.cels.anl.gov/

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 are achieved. In this work, we pursue verification of these multiphysics aspects of Cardinal using a 1-D analytical benchmark from the Naval Nuclear Laboratory [6]. This work does not require CFD, and thus NekRS will be left out of the 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 couples 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. The assumed model for thermal conductivity has already been inserted into the energy conservation equation.

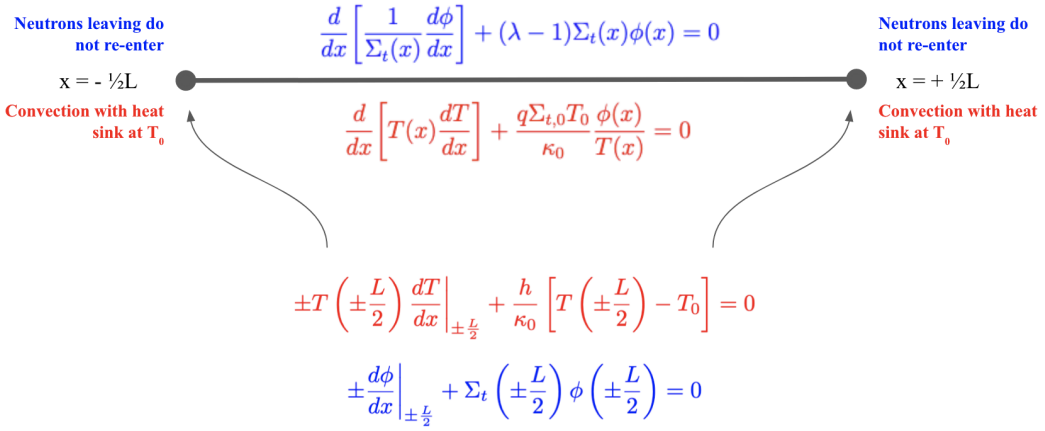

Figure 1: The domain, ODEs, and boundary conditions for the slab.

This benchmark uses a one-group assumption for the neutron cross sections. As neutrons transport, heat from fission events deposits 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 into 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,

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

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

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

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 κ, the energy released per reaction q, the total, macroscopic cross section Σ_t, and the neutron flux ϕ,

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

Note that typically q represents the energy per fission, but in this benchmark, q is per total reaction.

By defining the problem in this way, the differential equations for thermal conduction and neutron transport have similar forms, suggesting that it may be possible to assume that the temperature and flux are related by a constant, $T(x) = f\phi(x)$. This assumption is satisfied by manipulating the ODEs, inserting the cross section temperature dependence, and matching coefficients so that the two ODEs are of the same form. 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. An important consequence of this problem is that the fission power is uniform throughout the geometry. For full details on the analytical solution's derivation, see [6].

3. COMPUTATIONAL MODEL

This section describes the OpenMC and MOOSE computational models, followed by the convergence criteria used for coupling and a description of the mapping between OpenMC and MOOSE geometries. At the time at which this benchmark modeling was performed, Cardinal did not support moving geometries in OpenMC. Instead of using thermomechanics for thermal expansion, our simulation simply uses the analytic solution for the equilibrium length L from [6] 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 – as was mentioned in Section 2. We note, in a companion paper in this conference [7], that Cardinal now supports mesh-based geometry (and deforming-mesh problems) through the DAGMC plugin, and hence incorporating the thermomechanics feedback will be included in our future work.

3.1. OpenMC Model

The geometry used in OpenMC must be finite, despite the benchmark being infinite in the y and z directions. The slab is divided evenly into N identically-shaped, rectangular cells along the x-dimension; the cases here are $N = [50, 100, 250, 500, 1000]$ cells. 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. While reflective boundary conditions in the y and z directions are necessary, in general, to establish a 1-D problem, in this benchmark particles only move in the $\pm x$ direction. To simplify normalization of the power integral in Equation (19) of the benchmark, the geometry has 1 cm dimensions in the y and z directions[6]. Fig. 1 shows a diagram of the 1-D 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 (a range slightly larger than, but including the analytic temperature range from the benchmark) and was exported to a library. Then, when running OpenMC in multigroup mode, that library is loaded to determine the appropriate cross section in each region as temperature changed via input from Cardinal from iteration to iteration. For a temperature T between two library data points, we simply use the closest temperature available.

OpenMC 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 $\pm x$.

As with any Monte Carlo eigenvalue method, each Picard iteration requires sufficient inactive batches to converge the fission source, followed by sufficient active batches to accumulate the relevant tally results. A Shannon Entropy [8] study for this system was conducted on the 1000 cell case, and found that the Shannon Entropy converges after about 5 inactive batches. While this may seem like a low number, the problem is 1-D and the initial guess provided to OpenMC is a uniform distribution, which is the same shape as the actual converged solution. The k-eigenvalue simulation used 50,000 particles per batch, with 50 inactive batches and 50 active batches for every case.

For simplicity, we run 200 Picard iterations for each case, though more advanced metrics in Cardinal can be used for programmatically evaluating coupled physics convergence. Once the temperature was converged following 200 Picard iterations, a final OpenMC run was performed using the temperature distribution produced from the last Picard iteration with 250,000 particles per batch and the same number of batches.

Cardinal kept track of a few tallies in order to compare to the analytical solution: a flux cell tally, a fission heating (kappa-fission) cell and global tally, and the eigenvalue k_{eff}, all based on track-length estimators. 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. For an eigenvalue problem, the tally normalization relies on a source strength, S, related to the total system power:

$$S = \frac{P}{H}\left[\frac{\text{sp}}{\text{s}}\right] \tag{5}$$

where $P = 10^{22}\frac{\text{eV}}{\text{s}}$ is the slab integrated power from the benchmark and $H\left[\frac{\text{eV}}{\text{sp}}\right]$ is the total tallied fission heating tally. Each tally must be multiplied by this source strength to convert to physically meaningful units. Since OpenMC does not report volume-normalized tallies, it may be necessary to divide by the cell volume for quantities such as neutron flux and volumetric heating.

3.2. MOOSE Heat Conduction Model

MOOSE is used to solve for the temperature distribution within the slab via the Heat Conduction Module. In this MOOSE model, a mesh is used that has identical dimensions to the cells used in the OpenMC model. While this is not required by Cardinal, it allows for 1-to-1 feedback between the temperature computed in MOOSE and the heat source computed in OpenMC (and is adequate for this geometrically simple benchmark). MOOSE solves the conduction equation using the Finite Element Method (FEM):

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

where κ_s is the thermal conductivity in the solid and $\dot{q}_s$ is the heat source, in this case from fission. The boundary conditions used by MOOSE match the temperature boundary conditions in red shown in Fig (1). MOOSE is coupled to OpenMC by receiving the heat source in each element from the fission heating tally. During each iteration, MOOSE 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 MOOSE heat conduction solve used a Jacobian Free Newton Krylov (JFNK) iterative method, with 10^{-7} for absolute tolerance and 10^{-9} for relative tolerance. The Monte Carlo method is a stochastic

method, so convergence is achieved by using an appropriate number of batches and histories per batch. The configurations used here were already discussed in Section 3.1.

In terms of converging global iterations across all single physics, each case used 200 Picard iterations. To assist with convergence of Monte Carlo quantities, Robbins-Monro relaxation was applied to the flux and fission heating tallies. This updates the quantities of interest for the $n+1$th iteration as an average over the previous n Monte Carlo solutions [9]. So the flux at iteration n would be given by

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

where Φ_n is the relaxed flux at step n and 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.

MOOSE provides steady-state detection as an alternate way of detecting convergence. By setting a steady-state tolerance, MOOSE will iterate until the solution norm changes by less than this tolerance. In Cardinal, OpenMC and NekRS variables are "external" to other MOOSE non-linear variables. In this case, the auxiliary variables, including some variables that should not be included in the steady-state detection, get added. Some of these auxiliary variables may be irrelevant to the solution's convergence, causing a contribution to the steady state norm that is not changing between iterations. This can trick the steady-state detection into "detecting" convergence faster than it should. Thus, this feature was not adopted for this work, though a new issue was opened in the MOOSE repository to allow the user to control which auxiliary variables are used to assess convergence.

In this benchmark, a high number of Picard iterations was used in order to ensure convergence, but other modelers may not want to incur the computational cost of running 200 Picard iterations. As long as the user removes any auxiliary variables not used for coupling, they could rely on steady-state detection instead.

3.4. Data Mapping

Cardinal does not require the geometry models used in each single-physics sub-application to exactly match the meshes/geometry used in other coupled codes. At the beginning of every Cardinal simulation, a mapping between MOOSE's mesh and other geometry representations is established. For a MOOSE-OpenMC coupling, the centroid of every mesh element is mapped to a corresponding OpenMC cell. Cardinal then creates a tally for every user-requested type in each of the regions of OpenMC that a MOOSE centroid falls in. This mapping is used for transferring relevant quantities back and forth between Picard iterations. For this simulation, the MOOSE geometry and OpenMC geometries were intentionally created to have the same dimensions so that the mapping is 1-to-1. Each region in the `MooseMesh` computes a temperature and sends it to a geometrically identical OpenMC region. This same region in an OpenMC solve computes a heat source via the fission heating tally and sends it back to MOOSE for the next conduction solve. Again, we want to emphasize that this strict 1:1 mapping is *not* mandatory in Cardinal, and rather a general centroid-based many-to-one mapping from elements to cells can be used. In order to explore the convergence with spatial resolution, five different cases were analyzed with $N = [50, 100, 250, 500, 1000]$ spatial elements/cells in both MOOSE and OpenMC.

4. RESULTS

The results to compare with the benchmark are the temperature distribution, flux distribution, and k-eigenvalue. The benchmark provides analytical solutions for all of these quantities. An example temperature and flux distribution is shown in Figure 2 for the 50 mesh element case. In order to assess the accuracy of the numerical quantities and verify that refining the mesh increases the accuracy, the analytical solution was used to compute an L_2 error norm for temperature and flux. The error computed, ε_T, is given by

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

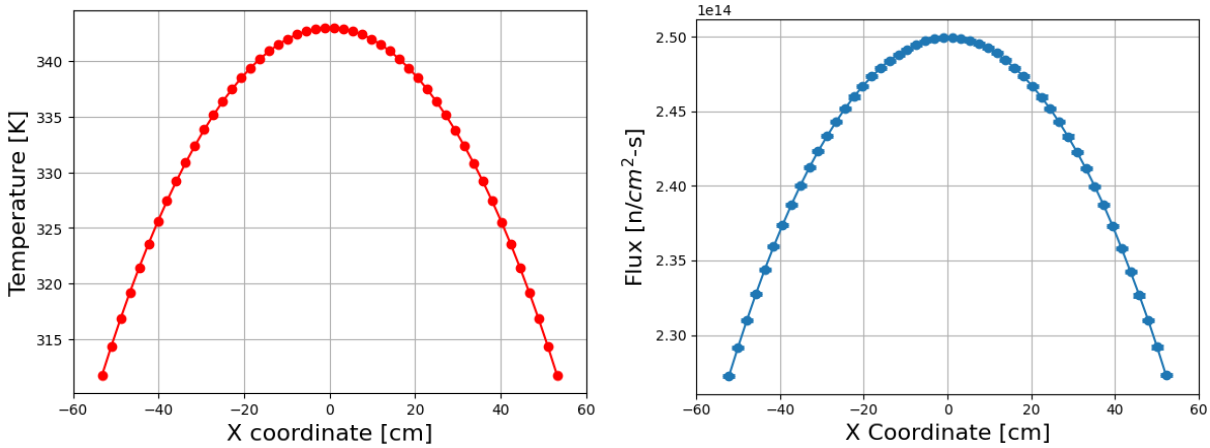

Figure 2: The computed temperature and flux for 50 mesh elements. The error bars show the relative error of the flux (nearly smaller than the circular marker size).

where T_a is the analytical solution evaluated at the x-centroid of each mesh element and T_x is the temperature computed for that voxel from the multiphysics simulation. The same convergence test was used on the flux solution as well. Using the same notation convention, the flux error norm, ε_ϕ, is defined as

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

Figure 3 shows how the temperature and flux error norms relate to the number of mesh elements and cells used.

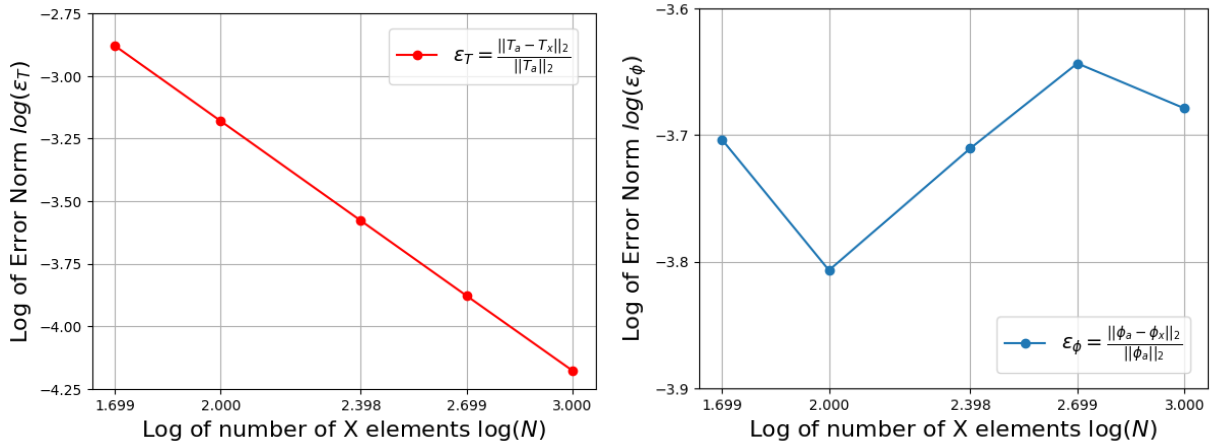

Figure 3: Temperature (left) and flux (right) error norms as a function of heat conduction mesh element count and OpenMC cell count, respectively.

For the temperature distributions, linear convergence is achieved, as the slope of the temperature line in Figure 3 is -0.9982 ($R^2 \approx 1.0$). The flux error norms do not appear to follow a linear pattern like the temperature error does. Spatial discretization contributes to the error and should decrease with increasing number of cells, but there are other error terms contributing. In the Monte Carlo method, as the volume of a region filtering a tally gets smaller, fewer events will happen in that region, and thus more histories are required to get the same uncertainty. When refining the spatial discretization, other discretizations, such as the temperature cross section library, may need to be increased to gain the full benefit of using a finer discretization in space. For example the resolution of the temperature data used to compute the cross section

could be contributing and overshadowing the decrease due to discretization. All cases used a library with 50 temperatures, so it is possible that the finer cases are reaching a limit where more data refinement would be required to achieve better convergence. In future work, we will repeat this benchmark with fewer than 50 elements/cells to explore whether we are in such a regime.

While the L_2 error norm of the flux does not exhibit the expected convergence, the results do compare favorably with the analytical results throughout the domain and for all spatial resolutions. The ratio between the computed quantity, C, and the expected answer, E, for the temperature in each element and flux in each cell are shown in Figure 4. A black line for $C/E = 1$ is marked in each plot indicating perfect agreement. Note the scales of the y-axes - the temperature is everywhere being predicted to within 0.006% (for the coarsest temperature mesh) and flux is everywhere being predicted to within 0.06%.

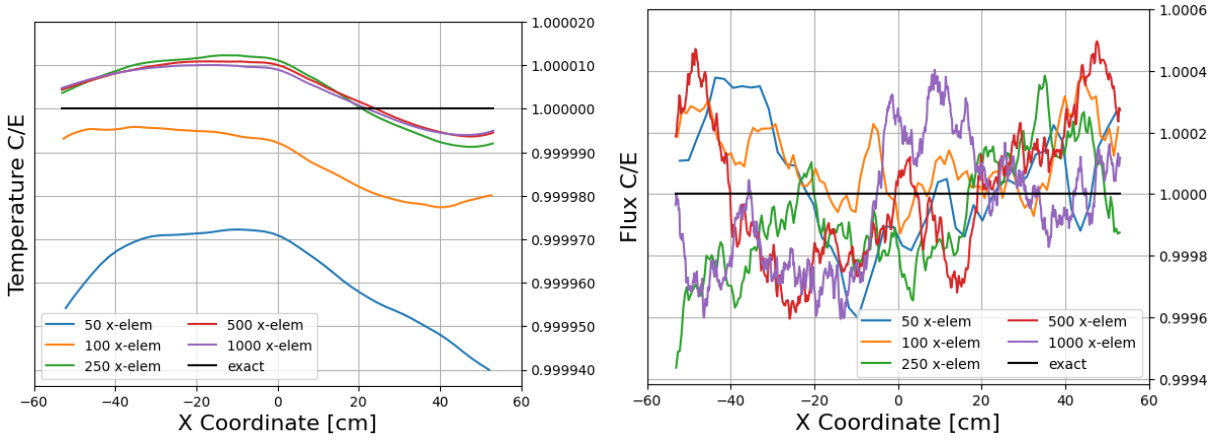

Figure 4: Temperature (left) and flux (right) C/E.

As the refinement in the mesh increases, the temperature C/E get closer to the ideal. The C/E ratio depends on two quantities with uncertainty; the flux from OpenMC $\hat{\phi}$ has uncertainty $\sigma_{\hat{\phi}}$ and the fission heating tally $\hat{H}$ has uncertainty $\sigma_{\hat{H}}$. Writing the C/E explicitly

$$C/E = \frac{P\hat{\phi}}{V_{cell}\hat{H}\phi_A} = \frac{S\hat{\phi}}{V_{cell}\phi_A}, \tag{10}$$

where V_{cell} is the tallied region's volume and ϕ_A is the analytical flux. Propagating both errors gives

$$\sigma^2_{C/E} = \left(\frac{\partial C/E}{\partial \hat{\phi}}\right)^2 \sigma^2_{\hat{\phi}} + \left(\frac{\partial C/E}{\partial \hat{H}}\right)^2 \sigma^2_{\hat{H}}. \tag{11}$$

Now denote ϕ, σ_ϕ, H, and σ_H as the quantities with physical units, i.e. the OpenMC outputs multiplied by the source strength. Evaluating the derivatives and simplifying gives

$$\sigma^2_{C/E} = \left(\frac{\sigma_\phi}{\phi_A}\right)^2 + \left(\frac{\phi}{\phi_A}\frac{V_{cell}}{P}\sigma_H\right)^2. \tag{12}$$

Figure 5 shows individual C/E with 2σ error bars (95% confidence interval). All points are within 2σ for each mesh choice, indicating that Cardinal is computing a correct flux distribution, within statistical uncertainty.

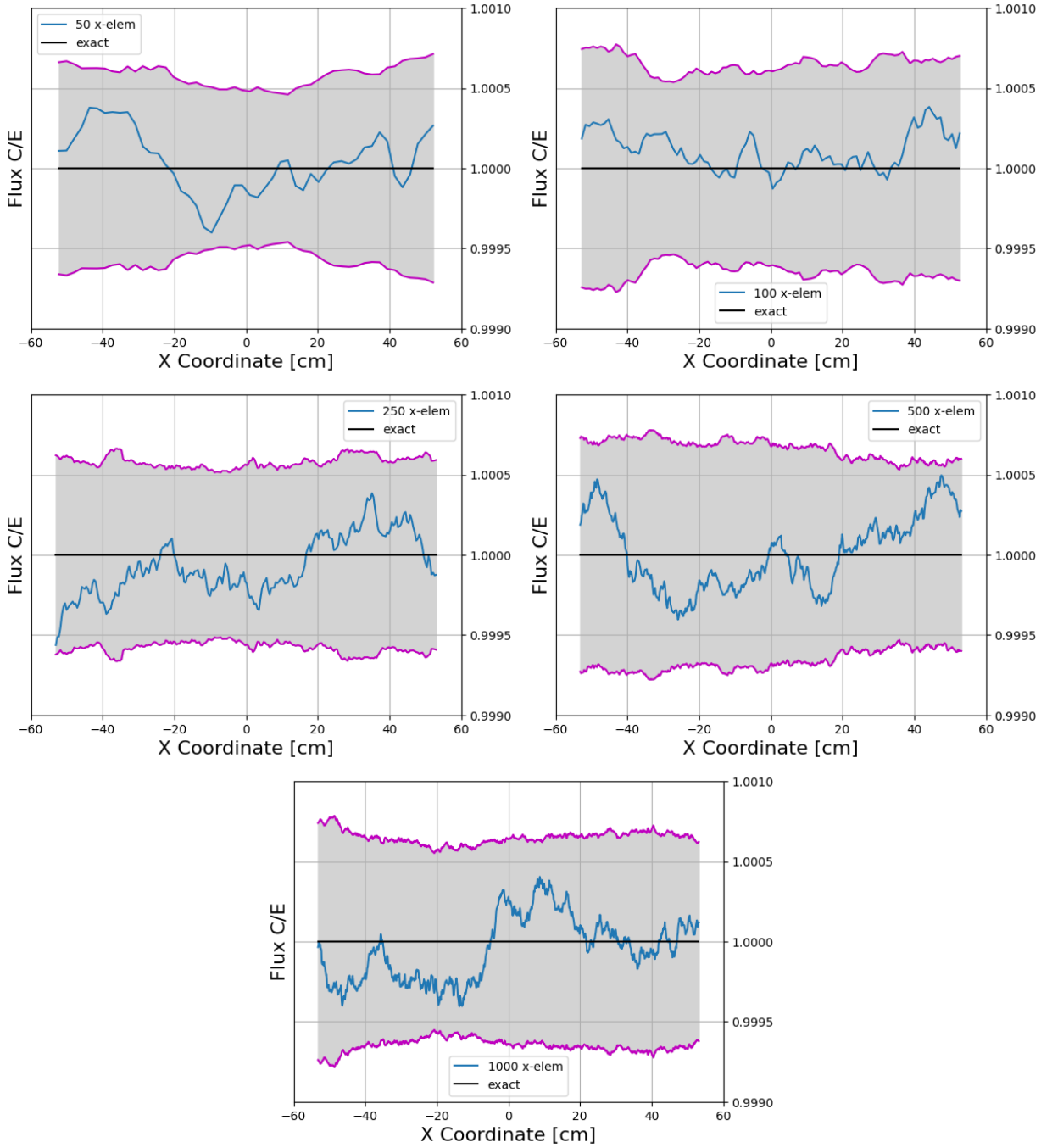

Figure 5: Each figure above shows a C/E with two-sigma error bars. The top left is the 50 case, followed by the 100 case, 250 case, 500 case, and 1000 case, in order.

The k-eigenvalue for each resolution is reported in Table 1. The results show close agreement with the analytical solution, as all cases within 9 pcm. The eigenvalues generally agree better as more mesh elements are used. The only exception being that the $N = 100$ case agrees better by 1 pcm over the $N = 250$ case. The good agreement is due to k_{eff} being a system-wide parameter which is less sensitive to spatial discretization of temperature feedback than the flux distribution itself.

Table 1: Eigenvalue with uncertainty for each mesh size, and the difference from the analytical.

Resolution	k_{eff}	(analytical - numerical) [pcm]
analytical	0.29557	-
50	0.29548 ± 0.00004	9 ± 4
100	0.29551 ± 0.00005	6 ± 5
250	0.29564 ± 0.00004	-7 ± 4
500	0.29552 ± 0.00006	5 ± 6
1000	0.29553 ± 0.00005	4 ± 5

5. CONCLUSIONS

Overall, the numerical results show agreement with the analytical solutions. The convergence is first order for the temperature and the C/E for the fluxes contain the analytical solution within two-sigma. The eigenvalue also agrees very well for all cases.

While experimental benchmarking is important in for nuclear M&S, agreement with analytical benchmarks also serve an important purpose. The ability to agree with physics theory increases confidence in multi-physics coupling. Though typical industry-grade simulations would not run S_2 transport, this modification allows Cardinal to be compared against a theoretical problem. A companion paper in this conference with Cardinal [10] couples OpenMC Monte Carlo transport to NekRS heat conduction. In future work, we will extend our computational model to include NekRS providing the heat conduction solution.

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.

Regarding the second and third authors, the submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (Argonne). Argonne, a U.S. Department of Energy (DOE) Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

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] "Cardinal: An Open-Source Coupling of NekRS and OpenMC to MOOSE." https://cardinal.cels.anl.gov (2022).

[3] 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).

[4] P. Fischer, S. Kerkemeier, M. Min, Y. Lan, M. Phillips, T. Rathnayake, E. Merzari, A. Tomboulides, A. Karakus, N. Chalmers, and T. Warburton. "NekRS, a GPU-Accelerated Spectral Element Navier-Stokes Solver." (2021). ArXiv:2104.05829.

[5] 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).

[6] D. Griesheimer and G. Kooreman. "Analytical Benchmark Solution for 1-D Neutron Transport Coupled with Thermal Conduction and Material Expansion." In *Proceedings of M&C*. Pittsburgh, Pennsylvania (2022).

[7] A. Novak, H. Brooks, P. Shriwise, A. Hegazy, and A. Davis. "Multiphysics Coupling of OpenMC CAD-Based Transport to MOOSE using Cardinal and Aurora." In *Proceedings of M&C*. Niagara Falls, Ontario, Canada (2023).

[8] F. Brown. "On the Use of Shannon Entropy of the Fission Distribution for Assessing Convergence of Monte Carlo Criticality Calculations." In *Proceedings of PHYSOR*. Vancouver, British Columbia, Canada (2006).

[9] J. Dufek and W. Gudowski. "Stochastic Approximation for Monte Carlo Calculation of Steady-State Conditions in Thermal Reactors." *Nuclear Science and Engineering*, **volume 152**, pp. 274–283 (2006).

[10] A. Hegazy and A. Novak. "Verification of the Cardinal Multiphysics Solver with the Doppler Slab Benchmark." In *Proceedings of M&C*. Niagara Falls, Ontario, Canada (2023).