Introduction to Moltres: an Application for Simulation of Molten Salt Reactors

Alexander Lindsay, Gavin Ridley, Andrei Rykhlevskii, Kathryn Huff

December 13, 2017

Abstract

Moltres is a new physics application for modeling coupled physics in fluid-fuelled, molten salt reactors. This paper describes its neutronics model, thermal hydraulics model, and their coupling in the MOOSE framework. Neutron and precursor equations are implemented using an Action system that allows use of an arbitrary number of groups with no change in the input card. Results for many-channel configurations in 2D-axisymmetric and 3D coordinates are presented and compared against other coupled models as well as the Molten Salt Reactor Experiment.

1 Introduction

Molten salt reactor concepts garnered considerable interest in the 1950s and 60s with development of the Aircraft Reactor Experiment (ARE) and later the Molten Salt Reactor Experiment (MSRE) at Oak Ridge National Laboratory (ORNL). With the inclusion of the Molten Salt Reactor (MSR) among the Generation-IV reactor designs [1, 2], this reactor concept has gained renewed research interest in the past decade, with many new nuclear companies proposing both fluid-fuelled and solid-fuelled commercial MSR designs [3, 4, 5, 6, 7]. The key advantages of MSRs generally pertain to improved fuel utilization and reactor safety. In contrast to legacy reactors, only moderator fast neutron damage and fuel chemistry evolution limit burnup. A clever configuration of moderator as in [8] can enable reactor operation without opening the vessel for thirty or more years. Further, several fission products selectively precipitate onto nickel surfaces in fluoride salt, as documented in [8], thus reducing unwanted neutron absorption. Lastly, the epithermal spectrum of graphite-moderated salt reactors incinerates plutonium more efficiently, thus reducing long-lived transuranic waste production [8]. The sum of these characteristics implies the MSR uniqely ameliorates spent fuel burden whilst extending nuclear fuel resources. To top all these benefits off, xenon transients become moot in MSRs due to its insolubility in salt, thus narrowing transient analysis focus to thermalhydraulic concerns.

Simulation tools developed by many authors successfully describe steady-state and transient behavior of myriad MSR concepts. Krepel et al. extended the in-house Light Water Reactor (LWR) diffusion code DYN3D to consider drift of delayed neutron precursors alongside the reactor temperature profile, re-casting the extended code as DYN3D-MSR [9]. That work compared DYN3D-MSR against experimental MSRE data and then used it to simulate local fuel channel blockages as well as local temperature perturbations.

In a similar vein, Kophazi et al. used iterative coupling between in-house three-dimensional neutronic and one-dimensional heat conduction models DALTON and THERM to analyze normal MSRE operation as well as channel-blocking-incident transients [10]. The Kophazi model added entrance effects of heat transfer coefficients as well as thermal coupling between fuel channels through moderator heat conduction. More recently, Cammi et al. performed a 2D-axisymmetric single-channel analysis of the Molten Salt Breeder Reactor (MSBR) using the commercial finite element package COMSOL Multiphysics [11]. That work directly solved the fuel salt velocity field, used heterogeneous group constants in fuel and moderator regions,

and employed the COMmon SOLution (COMSOL) software package intrinsically designed for coupled multi-physics simulation. Fiorina, Lathouwers, and their colleagues conducted a benchmarking exercise [12] in which this Politecnico di Milano approach was expanded to a multi-channel model of the Molten Salt Fast Reactor (MSFR) and compared to code from the University of Delft [13, 14] based on the approach in [10]. These models showed good agreement for multiple accident transients. Meanwhile, leveraging lessons learned from these efforts has resulted in a multiscale approach from Zanetti et al. [15] successfully combines high and low geometric fidelity for graphite-moderated MSRs.

More recently, Aufiero et al. [16] have begun to approach transient simulations in the MSFR within the finite volume OpenFOAM multiphysics toolkit [17]. This approach benefits from pre-implemented turbulence models available in the OpenFOAM library and captures the full-core three-dimensional geometry of the reactor primary circuit. OpenFOAM Computational Fluid Dynamics (CFD) has additionally been shown by Laureau et al. [18] to couple well with Transient Fission Matrix neutronics within the MSFR.

The present work introduces the open source simulation tool, Moltres [19], for simulating MSRs. By implementing deterministic neutronics and thermal hydraulics in the context of the Multiphysics Object-Oriented Simulation Environment (MOOSE) finite element modeling framework, Moltres solves arbitrary-group neutron diffusion, temperature, and precursor governing equations on a single mesh in anywhere from one to three dimensions and can be deployed on an arbitrary number of processing units.

Moltres is an open source code licensed under Lesser GNU Public License (LGPL) terms so the MSR community can freely use, interrogate, and improve it. Its openness is a defining characteristic and promotes quality through transparency and ease of peer review. In the era of GitHub [20] and international scientific collaboration, open and modern software practices must be employed in order for nuclear engineering simulation capability to enable discovery and support the regulatory needs presented by new reactor designs. In that vein, Moltres uses `git` for version control, integration testing to protect developed physics capabilities, and a C++ object-oriented design to enable extension and code reuse. While export control laws have the potential to restrict openness of neutron transport software that is dual use, there is no reason to believe that Moltres could be used for weapons proliferation. Moltres requires that cross sections be provided by the user, implements a combination of algorithms that can be easily derived from open literature, is tailored to molten salt reactor physics. Accordingly, Moltres joins a veritable parade of open academic nuclear engineering software such as OpenMC [21], OpenMOC [22], and PyNE [23, 24].

Moreover, Moltres depends on the MOOSE framework, [25] another LGPL code that itself leans on LibMesh [26], a LGPL finite element library, and PetSc [27], a Berkeley Software Distribution (BSD)-licensed toolkit for solving nonlinear equations yielded by discretizing PDEs. MOOSE and LibMesh translate weak PDE forms defined by applications (e.g. Moltres) into residual and Jacobian functions. These functions are the inputs into PetSc Newton-Raphson solution routines. All codes use MPI for parallel communication and are easily deployed on massively-parallel cluster-computing platforms. MOOSE applications by default use monolithic and implicit methods ideal for closely-coupled and multi-scale physics, such as the model problem described in this work. However, Moltres can also use explicit time-stepping routines as well as segregated solution methods, making it extensible to myriad future modeling challenges.

2 Methods

Moltres [19] is implemented as an application for use atop the MOOSE [25] framework. Accordingly, Moltres includes physics kernels and boundary conditions for solving for neutron fluxes, temperature, and precursor concentrations. In MOOSE jargon, kernels are C++ classes that contain methods for computing residual and Jacobian contributions corresponding to individual pieces of governing equations. Developing the code-base in this way allows modular construction of equation systems; e.g. the kernel used to represent heat conduction can also represent generic chemical diffusion. Moltres also features neutron and precursor "actions." These actions automatically construct the systems of equations for an arbitrary number of neutron and precursor

groups. Therefore, as long as group constants are provided in an appropriate tabular form, a user only has to modify a couple of lines in a Moltres input file to change from say two to forty-four neutron groups.

In Moltres, neutrons are described with time-dependent multi-group diffusion theory as shown in Equation (1):

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} - \nabla \cdot D_g \nabla \phi_g + \Sigma_g^r \phi_g = \sum_{g \neq g'}^{G} \Sigma_{g' \to g}^s \phi_{g'} + \chi_g^p \sum_{g'=1}^{G} (1-\beta)\nu \Sigma_{g'}^f \phi_{g'} + \chi_g^d \sum_i^I \lambda_i C_i \tag{1}$$

where

$$v_g = \text{speed of neutrons in group g} \tag{2}$$
$$\phi_g = \text{flux of neutrons in group g} \tag{3}$$
$$t = \text{time} \tag{4}$$
$$D_g = \text{Diffusion coefficient for neutrons in group g} \tag{5}$$
$$\Sigma_g^r = \text{macroscopic cross-section for removal of neutrons from group g} \tag{6}$$
$$\Sigma_{g' \to g}^s = \text{macroscopic cross-section of scattering from g' to g} \tag{7}$$
$$\chi_g^p = \text{prompt fission spectrum, neutrons in group g} \tag{8}$$
$$G = \text{number of discrete groups, g} \tag{9}$$
$$\nu = \text{number of neutrons produced per fission} \tag{10}$$
$$\Sigma_g^f = \text{macroscopic cross section for fission due to neutrons in group g} \tag{11}$$
$$\chi_g^d = \text{delayed fission spectrum, neutrons in group g} \tag{12}$$
$$I = \text{number of delayed neutron precursor groups} \tag{13}$$
$$\beta = \text{delayed neutron fraction} \tag{14}$$
$$\lambda_i = \text{average decay constant of delayed neutron precursors in precursor group i} \tag{15}$$
$$C_i = \text{concentration of delayed neutron precursors in precursor group i.} \tag{16}$$

Delayed neutron precursors are described by Equation (17):

$$\frac{\partial C_i}{\partial t} = \sum_{g'=1}^{G} \beta_i \nu \Sigma_{g'}^f \phi_{g'} - \lambda_i C_i - \frac{\partial}{\partial z} u C_i \tag{17}$$

with the last term representing the effect of fuel advection. The governing equation for the temperature is given by:

$$\rho_f c_{p,f} \frac{\partial T_f}{\partial t} + \nabla \cdot (\rho_f c_{p,f} \vec{u} \cdot T_f - k_f \nabla T_f) = Q_f \tag{18}$$

where

$$\rho_f = \text{density of fuel salt} \tag{19}$$
$$c_{p,f} = \text{specific heat capacity of fuel salt} \tag{20}$$
$$T_f = \text{temperature of fuel salt} \tag{21}$$
$$\vec{u} = \text{velocity of fuel salt} \tag{22}$$
$$k_f = \text{thermal conductivity of fuel salt} \tag{23}$$
$$Q_f = \text{source term} \tag{24}$$
$$\tag{25}$$

in the fuel, where the source term Q_f is defined by:

$$Q_f = \sum_{g=1}^{G} \epsilon_{f,g} \Sigma_{f,g} \phi_g \tag{26}$$

In the moderator, the governing equation for temperature is given by:

$$\rho_g c_{p,g} \frac{\partial T_g}{\partial t} + \nabla \cdot (-k_g \nabla T_g) = Q_g \tag{27}$$

where

$$\rho_g = \text{density of graphite moderator} \tag{28}$$
$$c_{p,g} = \text{specific heat capacity of graphite moderator} \tag{29}$$
$$T_g = \text{temperature of graphite moderator} \tag{30}$$
$$k_g = \text{thermal conductivity of graphite moderator} \tag{31}$$
$$Q_g = \text{source term in graphite moderator} \tag{32}$$
$$\tag{33}$$

In this work, $Q_g = \gamma V_{\text{core}}^{-1} \int_{\text{core}} Q_f \, \mathrm{d}V$, where γ is a factor representing heat dissipation by gamma and neutron irradiation in the moderator, called the graphite to fuel power density ratio. Robertson's [28] original MSBR analysis included a calculation of γ. [9, 29, 11] all calculated gamma heating through such a factor. We follow [11] and set $\gamma = 0.0144$. Notably, the Moltres physics kernel for radiation heating can be calculate the volume average fission heat in a variety of means: whole-core and local averages are possible. Whole-core averages are employed for simplicity and in accordance with prior literature. Knowledge of fision heat rates, however, require knowledge of neutron fluxes, implying a need for group constants.

Group constants are generated by the modeler with either Serpent [30] or SCALE [31]. Moltres interpolates group constant temperature dependence from prepared tables, which must be constructed separately for fuel and moderator regions. For this report, we generated group constants with SCALE with an infinite square pitch lattice of cylindrical fuel channels surrounded by graphite. This model maintained a fuel fraction of 0.225 to be consistent with the MSRE. Subsequently, a critical buckling calculation was applied. The SCALE input files used for generating the group constants appear in the `io/msre_conc_cuboid_lattice` directory of the `github.com/arfc/scale_io` repository.

2.1 Performance

Building on the massively parallelizable MOOSE framework allows Moltres to run on super-computing platforms like the Blue Waters supercomputer at the National Center for Supercomputing Applications

(NCSA). For some three-dimensional simulations, the number of elements in the mesh and total number of degrees of freedom exceed one million and ten million respectively. To handle problems of this size, we ran Moltres on up to 608 cores. However, reducing the problem dimension from three to two and using a structured mesh, which can be much more coarse in the axial direction, allows the problem to be solved on a single core in under five minutes.

3 Model Description

The model molten salt reactor closely emulates the MSRE. When developing new physics or investigating different types of transients, one can reduce the model problem to a two dimensional axisymmetric one for rapid proof of concept. To approximately simulate the lattice structure of the MSRE under 2D conditions, a geometry is constructed with 14 repeating fuel-moderator regions, as shown in Section 3. The fuel and moderator radii are chosen such that the resulting area/volume fraction of fuel is 0.225 as for the MSRE. The base 2D mesh has a characteristic size of 10 cm in the axial direction and .4 cm in the radial direction to capture the variation from moving between fuel and moderator subdomains. To determine whether results were converged, a mesh convergence study was conducted with up to three levels of isotropic refinement (e.g. each element was in half in both axial and radial directions, resulting in four new elements). The metric used to assess convergence was the integrated fast group flux. The result of the mesh convergence study is shown in section 3. From refinement level 2 to 3 the metric changes by only 1%. We regard the results as sufficiently converged at level 2 for the purposes of our study; consequently, the results reported in the following section correspond to a radial element dimension of .1 cm and an axial element dimension of 2.5 cm.

The model fuel composition is the Beginning-of-Life (BOL) enriched uranium composition in the MSRE and is given in Table 1 [32].

Component	Mass Fraction
Li-7	.1090
Li-6	5×10^{-6}
F-19	.6680
Be-9	.0627
U-235	.0167
U-238	.0344

Table 1: Fuel salt composition is the BOL enriched uranium composition in the MSRE design [32].

Other simulation inputs are outlined in Table 2. We chose a reactor simulation height of 151.75 cm to produce an approximately critical reactor configuration corresponding to BOL MSRE composition. This differs from the actual MSRE height, which was 162.56 cm.

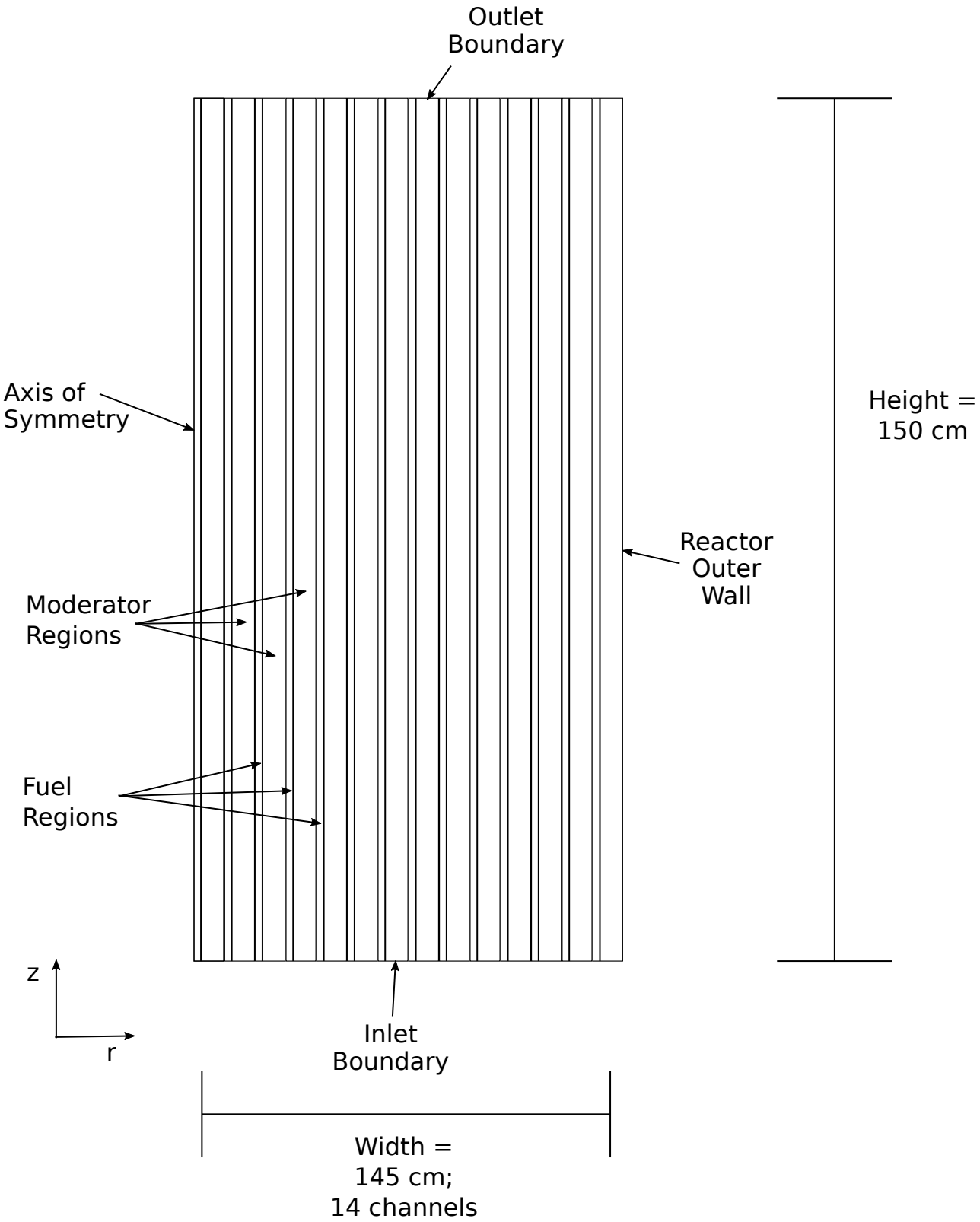

Figure 1: A sketch of the MSR model geometry

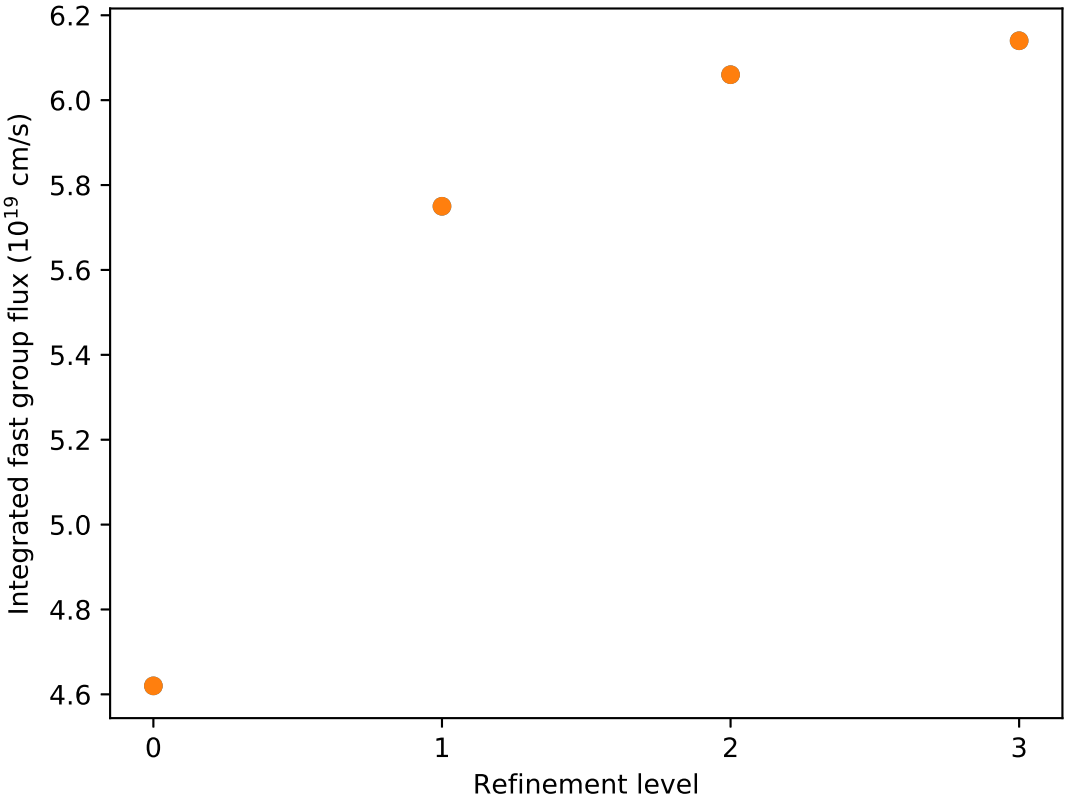

Figure 2: Plot of the integrated fast group flux as the mesh in isotropically refined.

Parameter	Value	Units	Source
Inlet temp.	922	K	MSRE nominal [32]
Wall temp.	922	K	MSRE nominal [32]
Neutron groups	2	1	User
Precursor groups	6	1	User
Reactor radius	72.5	cm	≈MSRE radius (70.2 cm) [32]
Reactor height	151.75	cm	User
k_f	.0553	$Wcm^{-1}K^{-1}$	[32]
$c_{p,f}$	1967	$JK^{-1}kg^{-1}$	[32]
ρ_f	$2.146 \cdot 10^{-3} e^{-\alpha_f (T_f - 922)}$	$kg\ cm^{-3}$	[32]
α_f	$2.12 \cdot 10^{-4}$	K^{-1}	[33]
k_g	.312	$Wcm^{-1}K^{-1}$	[11]
$c_{p,g}$	1760	$JK^{-1}kg^{-1}$	[11]
ρ_g	$1.86 \cdot 10^{-3} e^{-\alpha_g (T_g - 922)}$	$kg\ m^{-3}$	[32]
α_g	$1.8 \cdot 10^{-5}$	K^{-1}	[33]

Table 2: Simulation input parameters

4 Results & Discussion

Group fluxes are shown in Figures 3 and 4. The cosinusoidal shapes in radial and axial directions are caused by the vacuum boundary conditions. Both the fast and thermal fluxes are striated, with the fast group preferring the fuel and the thermal group preferring the moderator.

In Figure 5 the temperature rises along the reactor height because of fuel advection. The temperature gradient is negative in the radial direction, as expected.

Figure 6 shows the concentration of the longest lived precursor in the reactor. Not surprisingly, the channel concentrations are higher in fuel channels with higher neutron fluxes and corresponding fission events. Because of the small decay constant of the precursor, the maximum concentration in any given channel occurs at the core outlet due to advection.

With its much larger decay constant, the sixth and last precursor has its maximum concentration around the center-plane of the reactor as shown in Figure 7. As for all other precursors, its concentration decreases with increasing radius and decreasing neutron flux.

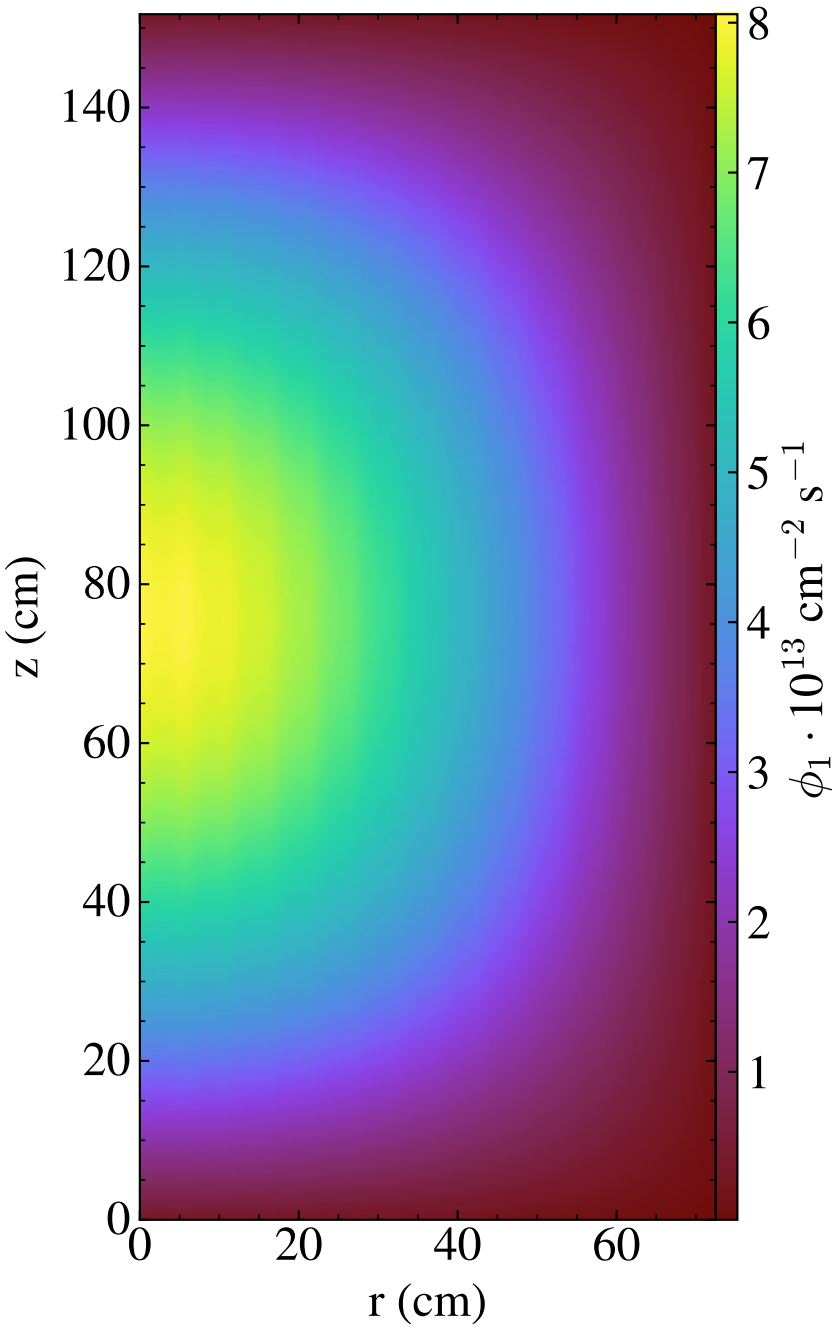

Figure 3: The group 1 flux in this 2-D cylindrical axisymmetric model has the anticipated magnitude and canonical cosine shape ($r = 0$ is center of core).

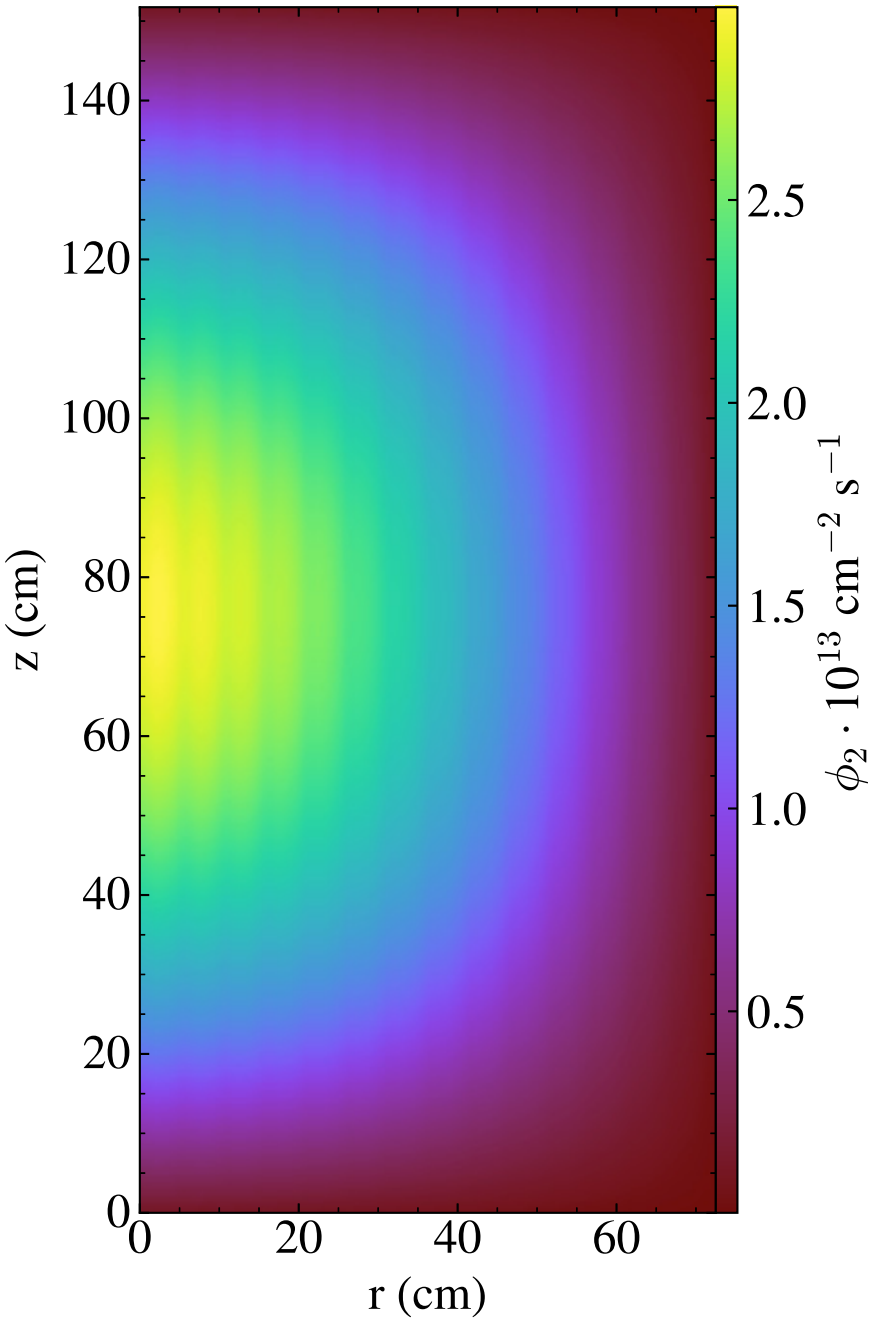

Figure 4: The group 2 flux in this 2-D cylindrical axisymmetric model has the anticipated magnitude and canonical cosine shape ($r = 0$ is center of core).

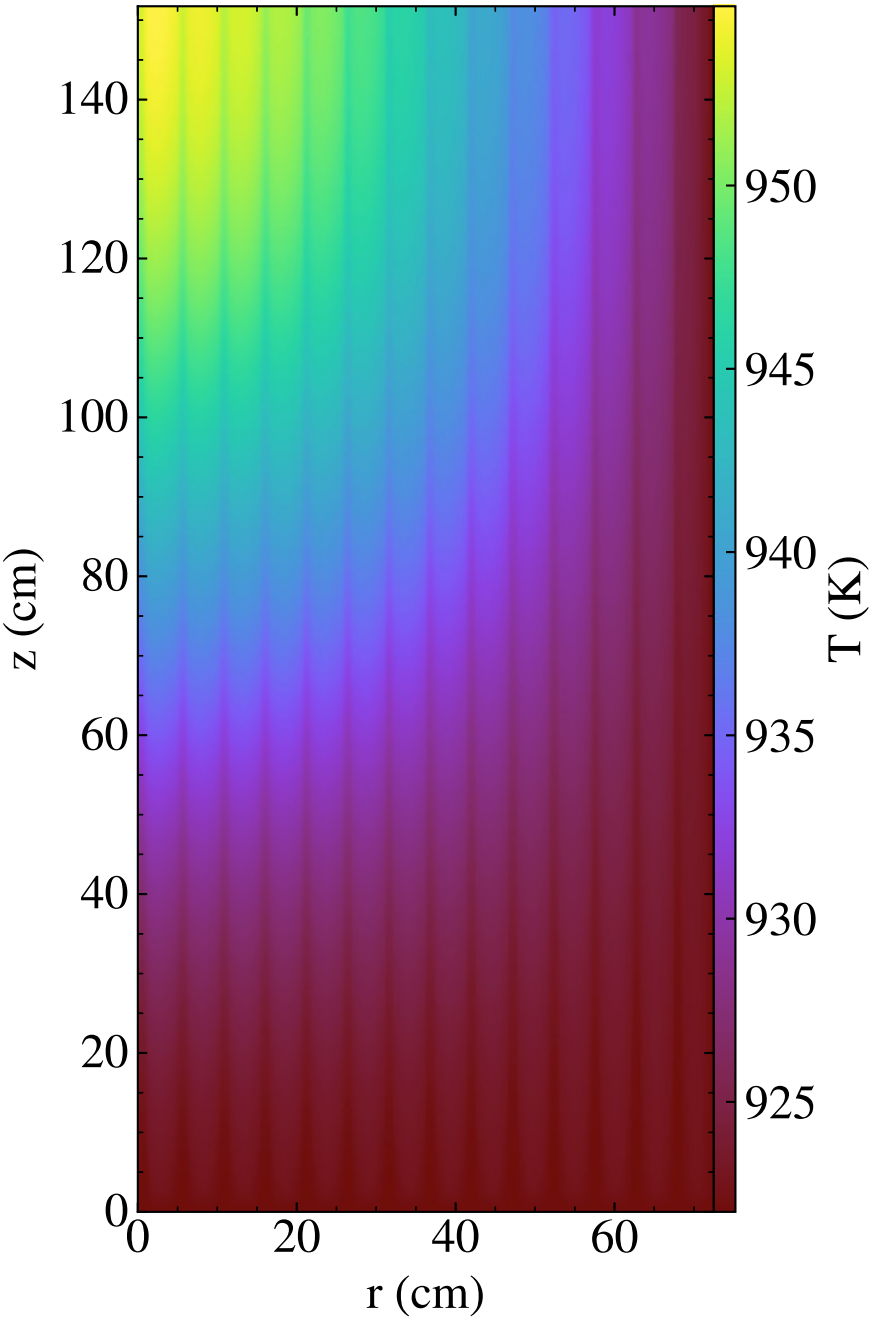

Figure 5: The reactor core temperature peaks near the reactor outlet in this 2-D axisymmetric model because of fuel advection ($r = 0$ is center of core). Because of heating by gammas and other radiation the moderator regions are hotter than the fuel, consistent with observations of the MSRE

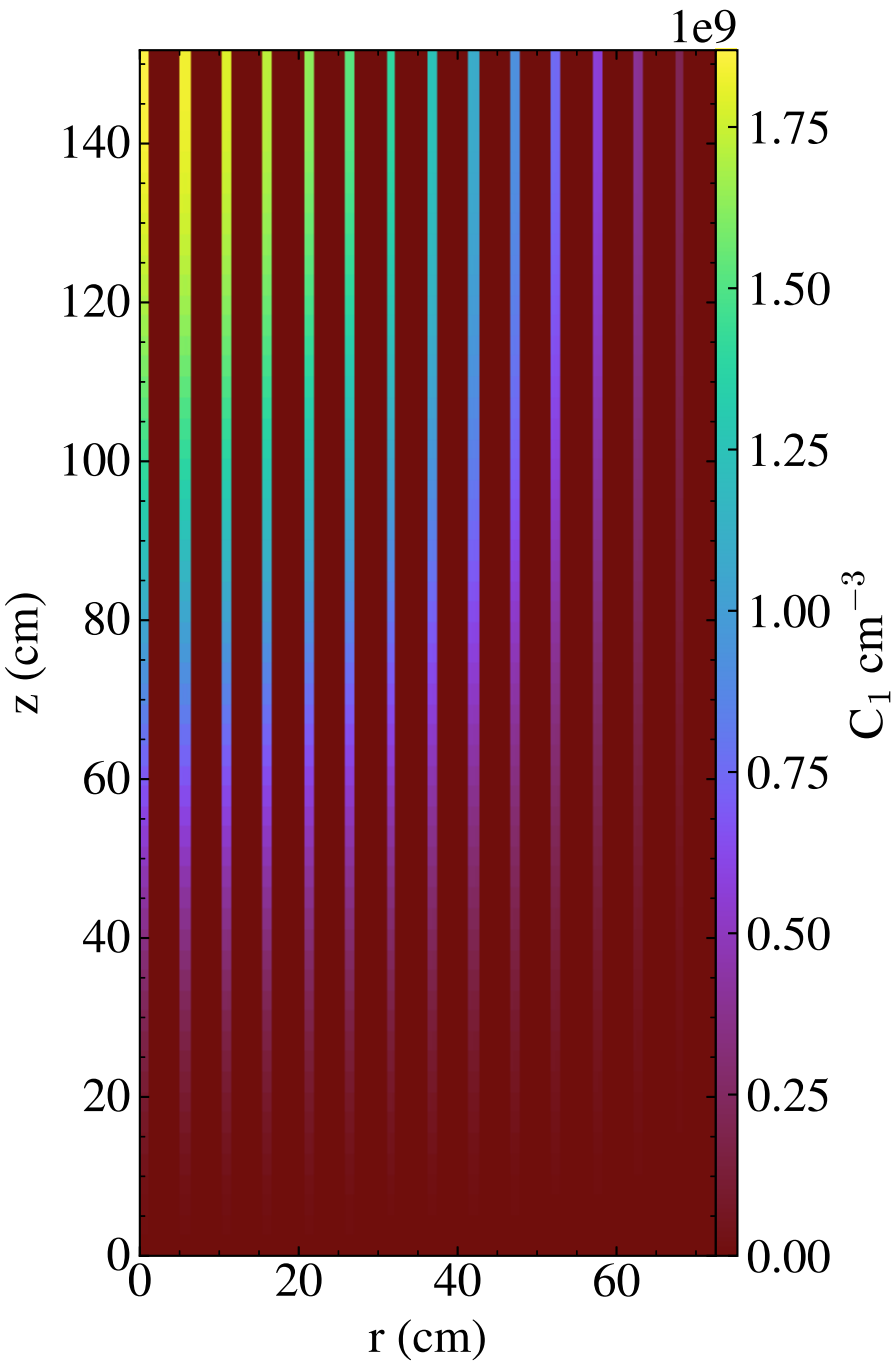

Figure 6: The concentration of the group of longest lived precursors ($\lambda = 1.24 \cdot 10^{-2} s^{-1}$) peaks near the reactor outlet in this 2-D axisymmetric model ($r = 0$ is center of core).

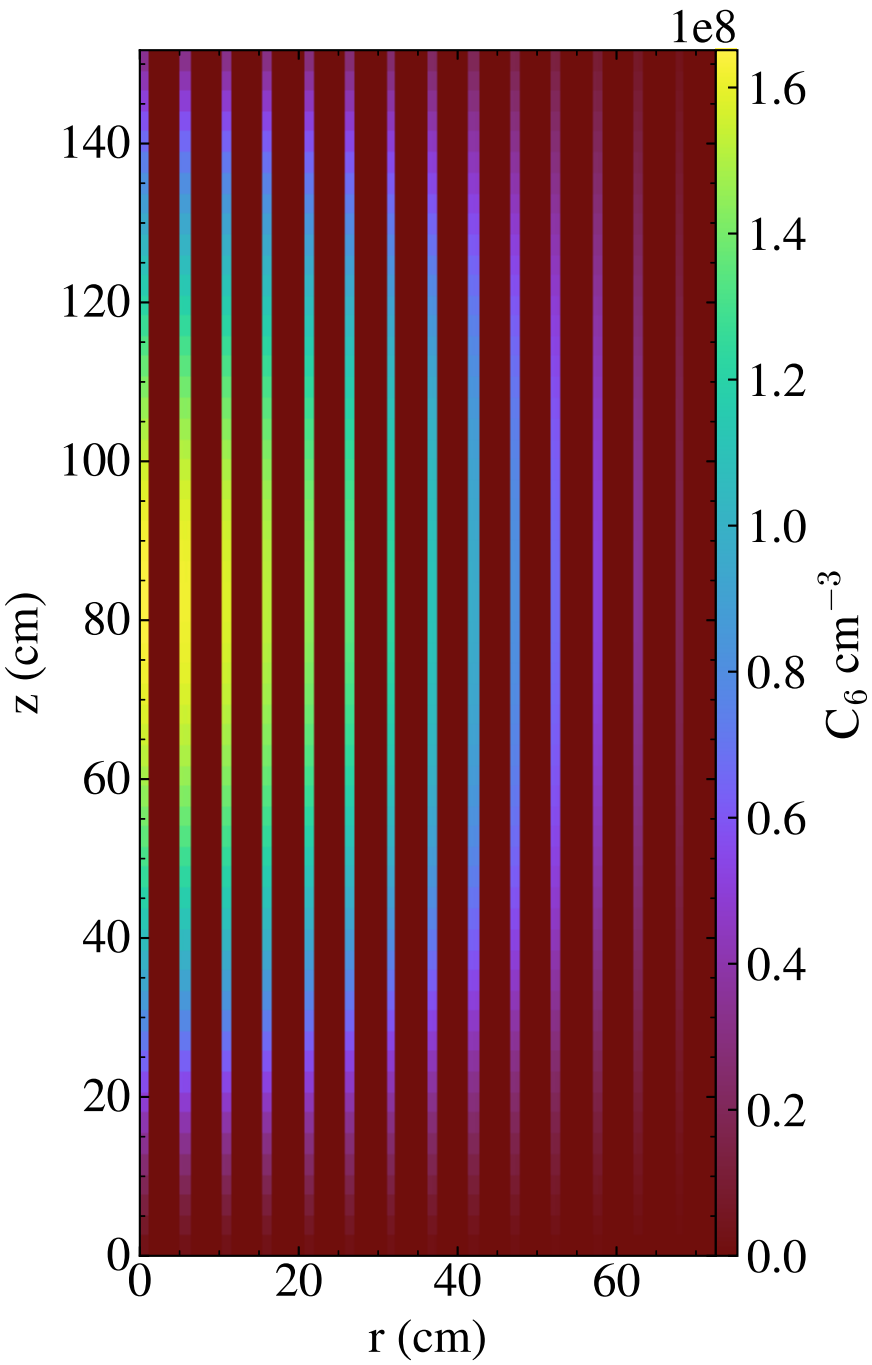

Figure 7: The concentration of the group of shortest lived precursors ($\lambda = 3.07s^{-1}$) peaks near the reactor center in this 2-D axisymmetric model ($r = 0$ is center of core).

4.1 Three dimensional simulation capability

Figures 8 and 9 show Moltres physics applied to a three dimensional geometry. The fast group flux (fig. 8) is in good qualitative agreement with the two dimensional axisymmetric case shown in fig. 3. Figure 9 is in similarly good agreement with fig. 5. Figure 10 shows the temperature profile at the outlet of the reactor (z = H). This three dimensional case contained 1,155,045 degrees of freedom and took only 2.5 hours to solve on 160 Blue Waters cores.

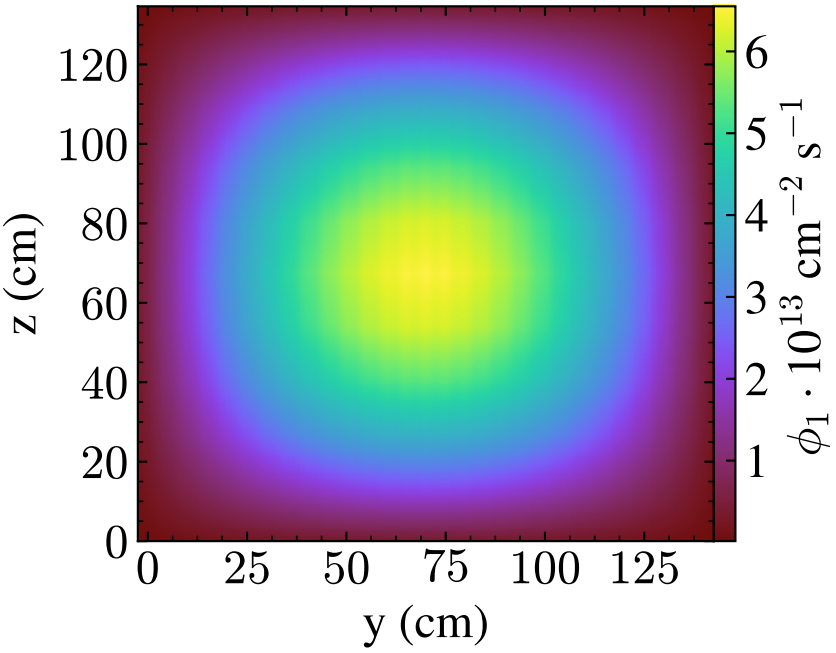

Figure 8: Fast flux for 3D MSRE-like model. Magnitude and shape in good agreement with 2D axisymmetric model (see fig. 3)

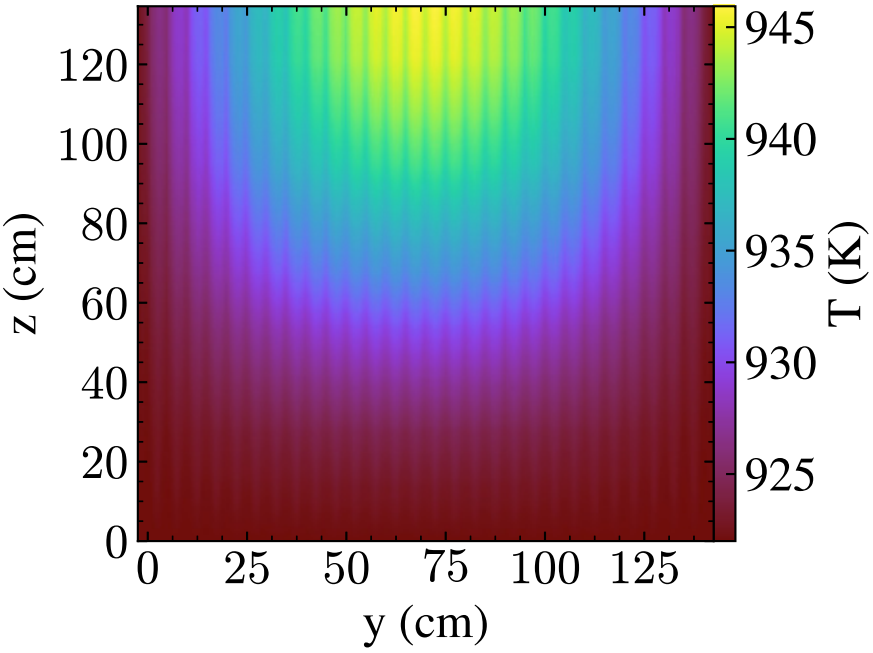

Figure 9: Temperature for 3D MSRE-like model. Magnitude and shape in good agreement with 2D axisymmetric model (see fig. 5)

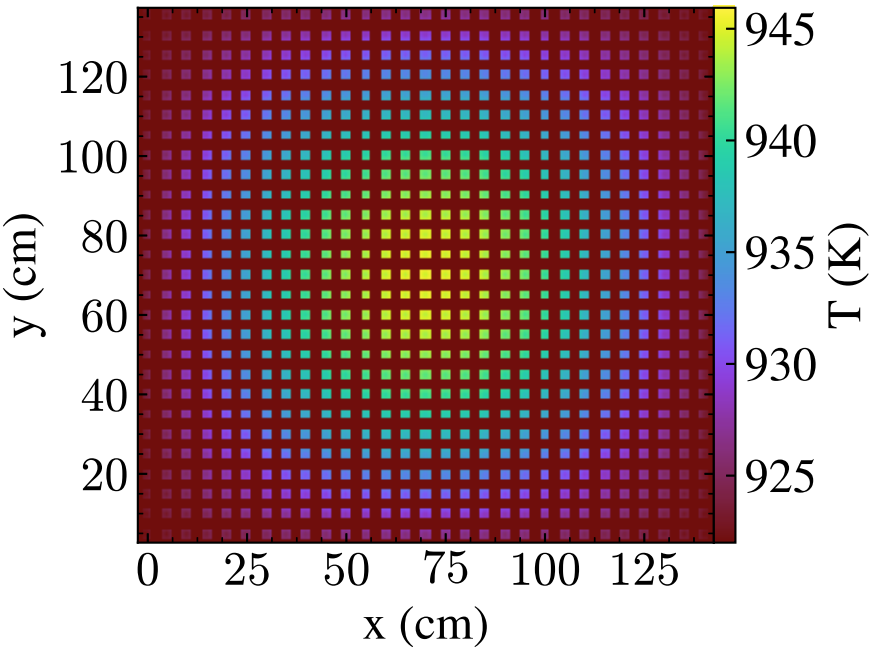

Figure 10: Temperature at the reactor outlet for 3D MSRE-like model.

4.2 Comparison with MSRE

Figure 11 shows a comparison between Moltres predicted temperature profiles with cosinusoidal gamma heating and MSRE design calculations [34] in the hottest channel and adjacent graphite. The ORNL MSRE design calculations, conducted in 1963-1964, were 32-group calculations using legacy computing tools (GAM-I, MODRIC, and EQUIPOSE, and THERMOS). Those calculations were conducted in two-dimensional R-Z geometry (a cylinder with angular symmetry), with 20 spatial regions. Notably, one limitation of the ORNL model was the control rod thimbles that, due to angular symmetry, were effectively a cylindrical shell of metal.

The profile shapes are in decent qualitative agreement with both Moltres and MSRE calculations showing a peak in graphite temperature before the reactor outlet. Fuel temperature increases monotonically in both Moltres and MSRE models. In the MSRE design, the moderator temperature at the reactor inlet is about 11 K larger than the fuel temperature, whereas the temperatures are about the same in the Moltres model. This difference is likely because the MSRE design model neglected axial heat conduction [34, p. 99].

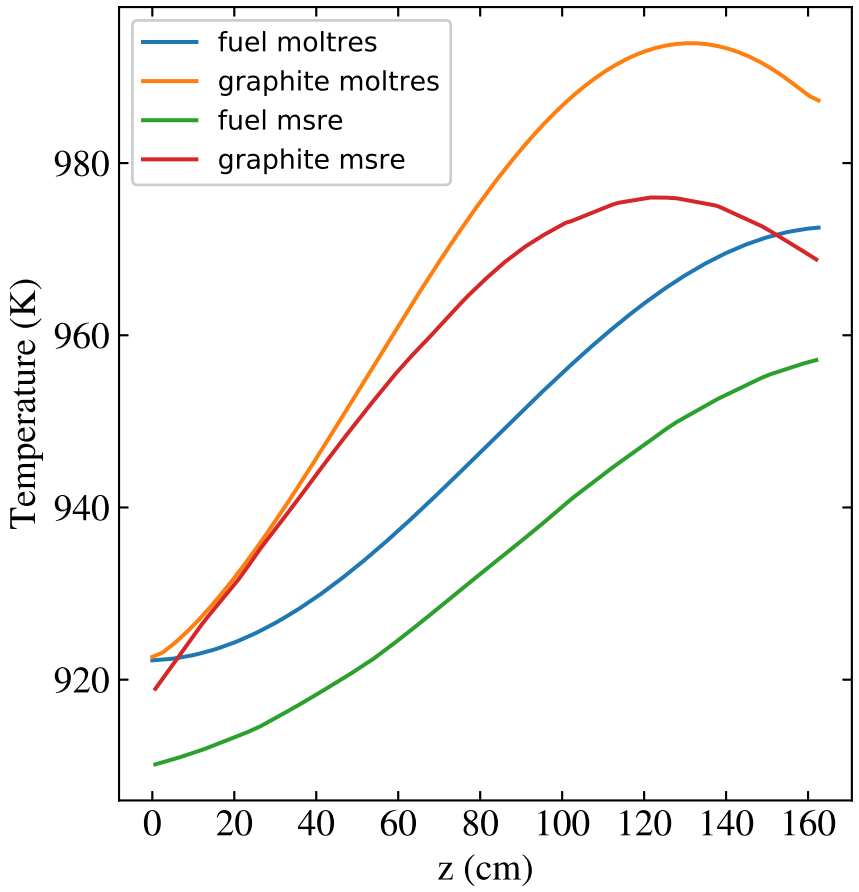

Figure 11: Moltres and MSRE design [34, p. 99] predicted axial temperature profiles in hottest channel and adjacent graphite

Figure 12 compares the fast and thermal neutron fluxes at the reactor mid-plane ($z = H/2$) for Moltres and MSRE design models. Local thermal flux growth and fast flux decay in moderator regions and visa versa in fuel regions are apparent in the Moltres calculation. The Moltres flux magnitudes are in good agreement with the magnitudes from the MSRE design calculations [34, p. 92]. The peak fast to thermal flux ratio

is approximately 3.5 in the MSRE design calculation as opposed to a ratio of 3 for the Moltres calculation. Control rod thimbles and an extra volume of surrounding fuel not included in the Moltres calculations cause the depression in the thermal flux in the MSRE profile.

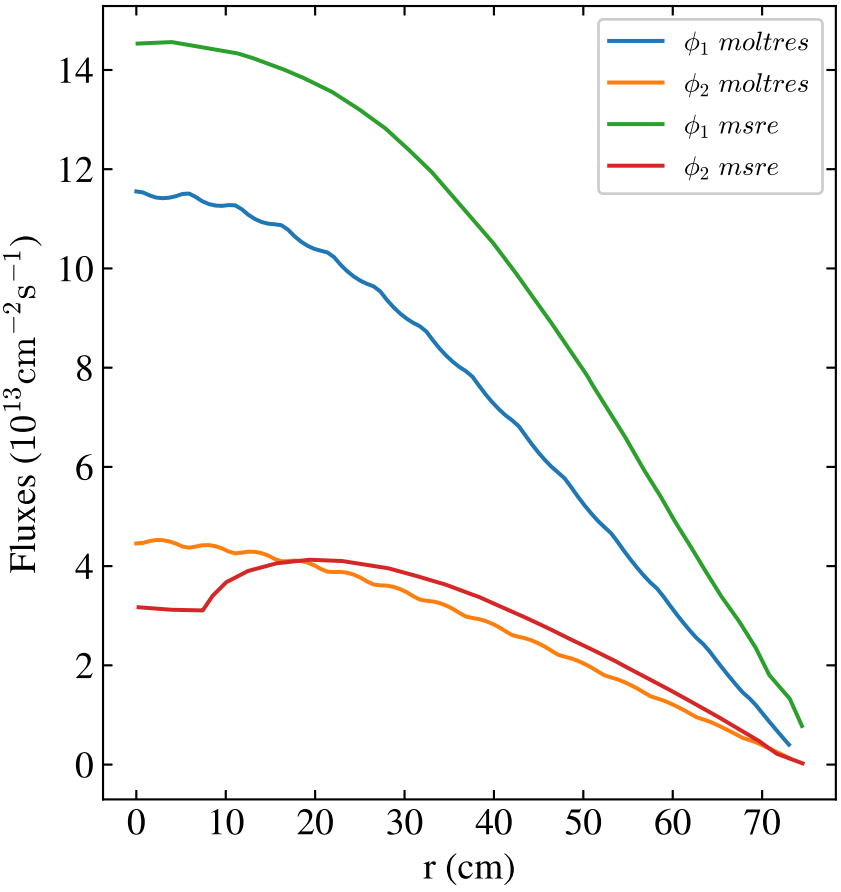

Figure 12: The thermal and fast flux profiles at the core mid-plane ($z = H/2$) for the Moltres 2-D cylindrical axisymmetric model and the MSRE design model [34, p. 92] ($r = 0$ is radial center of core).

Figure 13 compares the axial flux profiles calculated by Moltres and the ORNL MSRE design model. The radii for the plots are chosen to correspond to the peak of the thermal flux in both cases; for the ORNL calculations this is 21.336 cm (8.4 inches) from the core center-line because of the effect of the control rod thimbles and extra fuel along the center-line. Once again, the plots are in decent agreement. The ORNL calculations include the lower and upper plena which are not included in this report's Moltres model. Consequently, the MSRE lines extend to lower and higher z-values than the Moltres lines. Additionally, absorption in the plena cause deviation of the thermal flux from a sinusoidal shape in the MSRE design case. The peak power density from the MSRE calculation is 31 kW/L; the corresponding value for Moltres is 29 kW/L.

Although the qualitative agreement is decent, there are discrepancies between the MSRE and Moltres calculations. As outlined above there is an 11 K offset between MSRE and Moltres calculations for the fuel temperature in the hottest channel. The peak graphite temperature is around 14 K larger in the Moltres calculation. Fast fluxes are larger in the MSRE calculation by roughly 20%. These are not insignificant differences. However, given the differing nature of the two models–no axial conduction in the MSRE model, 2-group vs 32-group neutronics, exclusion of the control rod thimbles in the Moltres calculation–we believe this variation in quantitative behavior is acceptable for the purpose of this work, which is to introduce

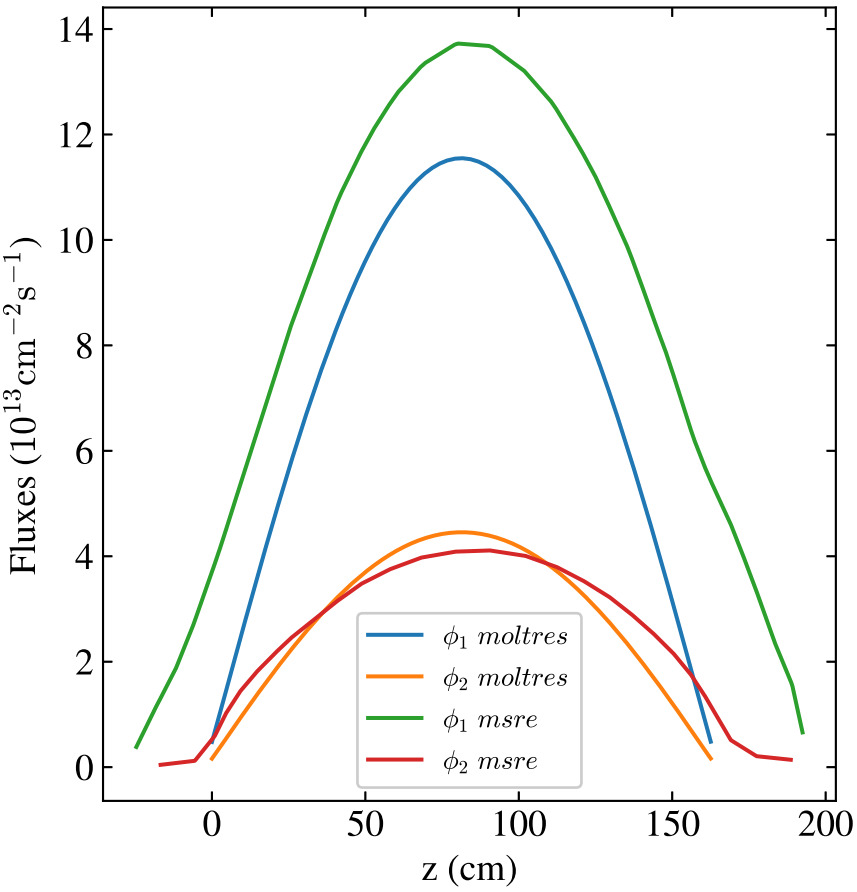

Figure 13: Moltres axial flux profiles along the core center line and MSRE design axial flux profiles 21.336 cm (8.4 inches) from the core center line [34, p. 91].

Moltres as a simulation tool. One additional feature that could influence Moltres results is inclusion of precursor decay heat; this is under active development. Trustworthy verification of Moltres results, such as spatial flux and temperature profiles, will require more detailed experimental measurements than those available in the ORNL technical reports, which contain only macroscopic data such as steady-state power and heat-exchanger temperature drops. Additionally, detailed comparison with other modern modeling efforts such as that conducted in [16, 18] is desirable. In that vein, a collaboration is under way with the primary author of [16] to try and reproduce calculation results from OpenFOAM. Further, Collins from [35] has expressed interest in comparing Moltres results against MSR-VERA in order to quantify errors resulting from usage of multigroup diffusion in comparison to 2D-1D method of characteristics fine group solutions. We hope to publish the results of these comparisons in a future work.

4.3 Scaling performance

Parallelization in Moltres is implemented via LibMesh, which includes a set of utilities for massively parallel finite element based computations, including mesh input/output, a finite element library, and an interface for connections with solver packages. Employing LibMesh provides Moltres with significant flexibility including the ability to swap out solver libraries such as PetSc, which includes an expanding suite of parallel linear and nonlinear solvers. Problem domain decomposition relies on LibMesh mesh adaptation capabilities for

running on a specific number of processors and can either be performed manually before the start of the simulation or automatically at the parameters of computation.

We conducted strong and weak scaling studies to characterize parallel performance in Moltres. In case of strong scaling, the problem size remains fixed but the number of processors is increased. Strong scaling studies seek to identify an optimal ratio between the number of processors and elements for the most rapid and power-efficient computation for a given problem. We measured Moltres strong scaling with a simple 2D axisymmetric case for various problem sizes separately for intra-node (2,820; 5,640; 11,280 and 28,200 elements) and extra-node (86,655; 173,310; 317,735; 664,355 elements) setup on Blue Waters' XK7 nodes (two AMD 6276 Interlagos CPU per node, 16 floating-point bulldozer core units per node or 32 "integer" cores per node, nominal clock speed is 2.45 GHz).

Figure 14 shows the simulation speed in seconds per element vs. the number of cores on 1 node (maximum 32 cores). Up to 8 cores, larger problems required considerably more time per element because of cache overhead. However, beyond 8 cores, scaling demonstrates asymptotic dependence on the number of processors due to increasing communication costs. The best parallel efficiency for the intra-node study is approximately 89% and has been achieved for the largest problem (28,200 elements).

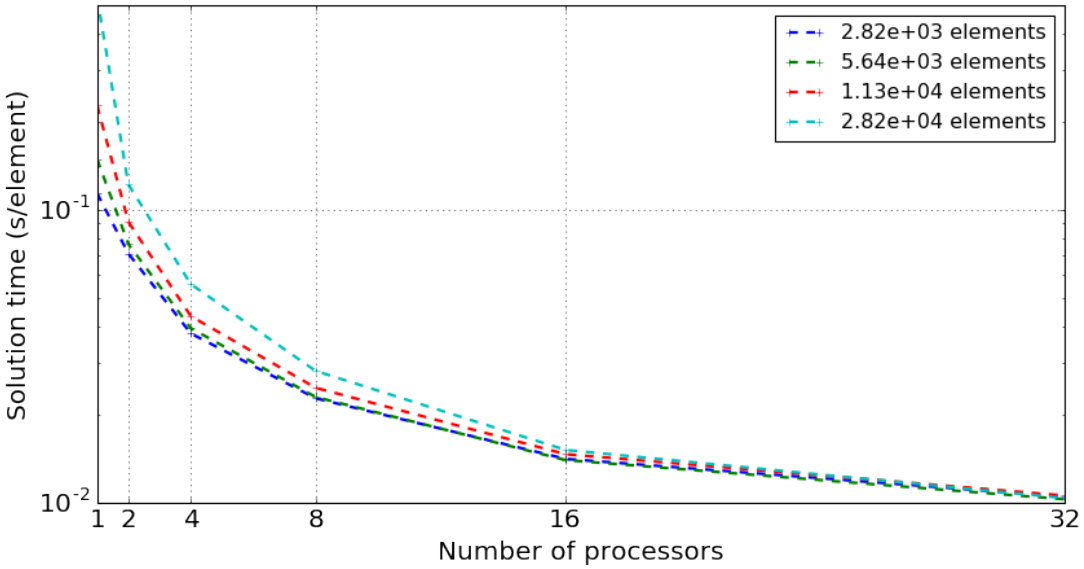

Figure 14: Moltres intra-node strong scaling efficiency for various problem sizes, for $n_{cores} \in [1, 32]$.

Figure 15 shows Moltres strong scaling up to 768 processors. This takes into account communication costs between nodes. Similar to the intra-node study, when fewer than 128 cores were used, cache overhead causes performance slow down for larger problems. However, beyond 256 cores, the simulation time per element remains almost constant for small cases (86,655 and 173,310 elements) and slighly decreases for the two larger problems. For extra-node scaling, parallel efficiency also grows with the problem size and reaches an optimal value of 73% for 664,355 elements.

For the weak scaling study, the part of the problem (workload) assigned to each processor stays constant and additional elements are used to solve a bigger problem which would not fit in memory on a single node. Thus, the weak scaling measurement is justification for memory-bound application such as multiphysics code. Linear weak scaling is achieved when the execution time stays constant while the workload increasing in direct proportion to the number of cores. We performed Moltres weak-scaling tests on Blue Waters, keeping the workload constant at 581, 985, 1970 and 3940 elements per core. Figure 16 shows Moltres weak scaling performance measured for $n_{cores} \in [1, 32]$ within one Blue Waters node and Figure 17 demonstrates performance for $n_{cores} \in [32, 128]$. As expected, the largest drop in performance occurs when the number of

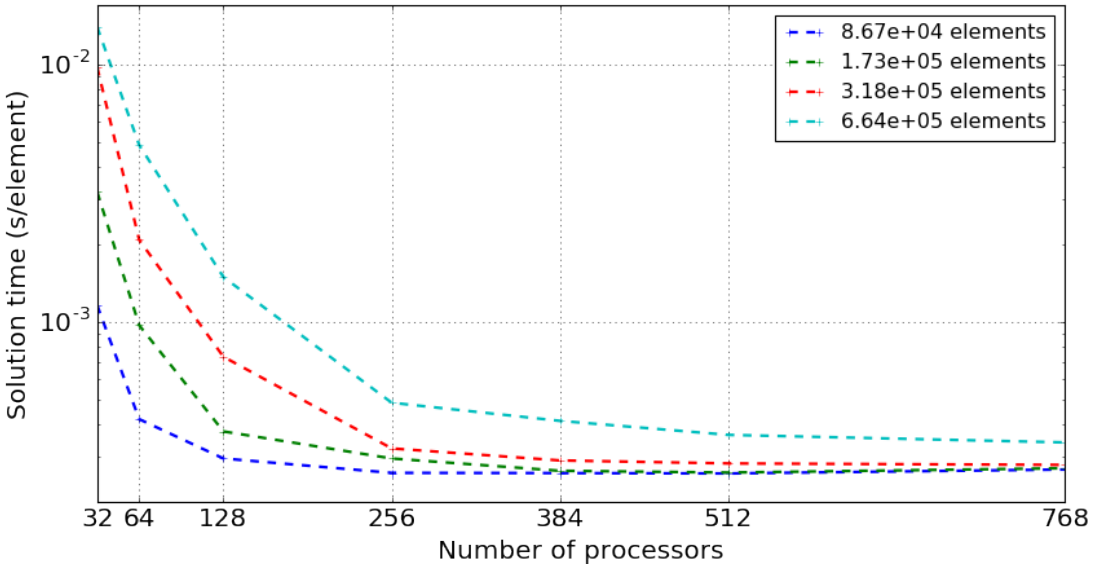

Figure 15: Moltres extra-node strong scaling efficiency for various problem size, for $n_{nodes} \in [1, 24]$.

cores increases from one to ≈ 8, which corresponds to switching from no communication to a 2-D domain decomposition. The further reduction in performance of only about 50% over a range of 32 cores is likely caused by increased communication latency appearing from collective Message Passing Interface (MPI) calls. In the extra-node case, the performance drops by a factor of three, which is most likely due to poor node selection by the Blue Waters job scheduler and significantly increased latency and bandwidth costs.

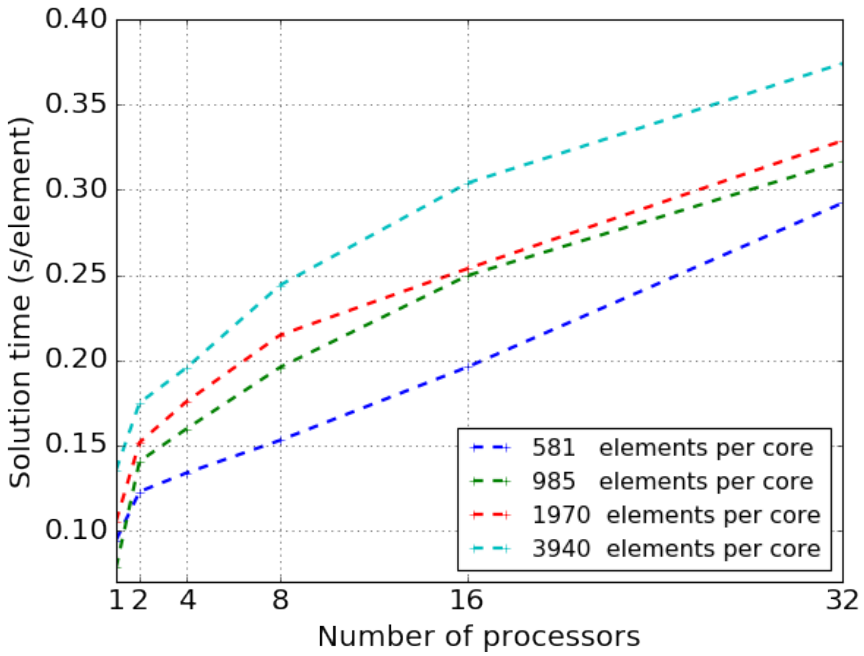

Figure 16: Weak scaling performance of Moltres on Blue Waters, in seconds per element vs. number of processors, for a constant number of elements per processor, and $n_{cores} \in [1, 32]$.

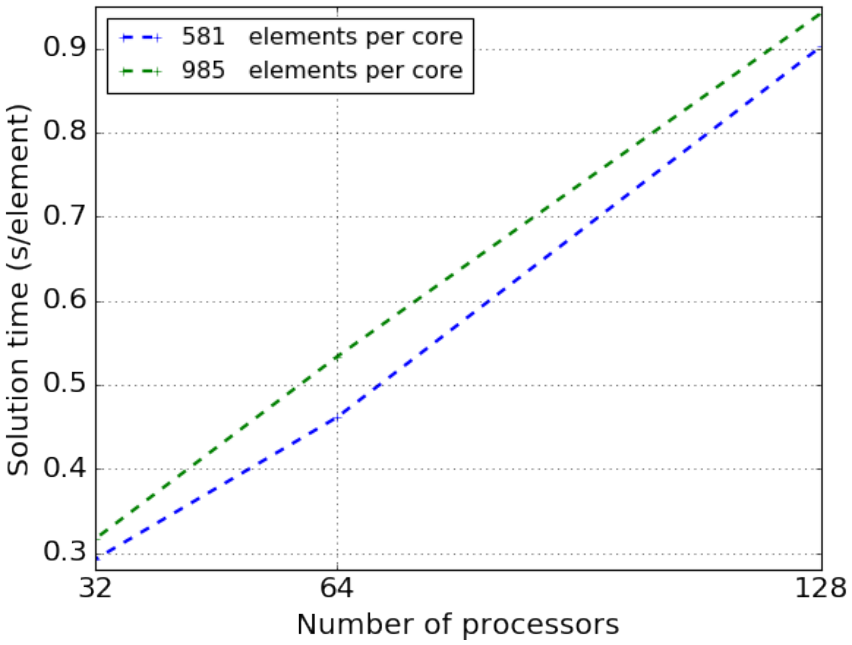

Figure 17: Weak scaling performance of Moltres on Blue Waters, in seconds per element vs. number of processors, for a constant number of elements per processor, for and $n_{cores} \in [32, 128]$.

Moltres scalability study results clearly indicate that parallelization using LibMesh's automatic domain decomposition is good, but not perfectly efficient. This scaling performance is satisfactory for MSR simulations approached thus far and improved parallel performance would require further optimization within LibMesh. Moreover, Moltres is memory-bound and therefore very sensitive to host memory and memory bandwidth. Consequently, if improved performance is needed, one could consider a transition from CPUs computing to GPU-accelerated computing because GPUs operate on the fly with global memory, avoiding CPU cache storage issues. Another way to improve parallel performance is to force the solver to use "older" information from previous iterations. However, this has been shown to slow convergence in terms of iterations and increased workload [27].

4.4 Monolithic vs. Segregated Solution

Relative performance of monolithic vs. segregated solution methods for large systems of equations is an active area of research. Monolithic solvers are generally regarded to be more robust than their segregated counterparts but are often perceived as being too expensive for large-scale problems. Work conducted by Heil et. al. [36] demonstrates that monolithic solvers are competitive with segregated techniques for weakly coupled physics; moreover, for strongly coupled problems the segregated methods struggle to achieve convergence. It is noted by the authors that use of monolithic solvers on large 3D problems requires efficient preconditioning to be effective.

Monolithic and segregated solution techniques were compared in Moltres using the 2D-axisymmetric problem but with delayed neutron precursors removed. In this case starting from arbitrary initial conditions the neutron multiplication exhibits damped oscillation around the critical state. Feedback between the power deposition by neutrons and temperature modification of neutron macroscopic group constants make this a very tightly coupled problem. Using adaptive time stepping based on the number of non-linear iterations at each time step, steady-state neutron fluxes and temperature profiles are achieved in 358 time steps and in 4 minutes of compute time on a single core using full-coupling (e.g. neutron fluxes and temperature in the

same matrix). Alternatively, a solution was also attempted splitting neutron and temperature solves and performing Picard iterations until convergence. Ultimately, a steady-state solution could not be achieved using the segregated method. After 670 time steps, the Picard solve hits the maximum number of allowed iterations (100) after only halving the neutronics residual. Before reaching the fail-point, the segregated solution required numerous Picard iterations to converge (68, 37, 22 for the previous three time-steps for example), resulting in a total compute time of 37 minutes before failure. These results suggest that for solving tight coupling between neutronics and temperature, a monololithic solve approach is more effective.

5 Conclusion

This work introduces the open source MSR simulation code Moltres. Moltres solves arbitrary-group neutron diffusion, temperature, and precursor governing equations in anywhere from one to three dimensions and can be deployed on an arbitrary number of processing units. The 2D-axisymmetric and 3D models presented here employ heterogeneous group constants for fuel and moderator regions generated with SCALE. Fuel volume fraction and fuel salt composition are based on the MSRE. Neutron fluxes show expected cosinusoidal shapes in radial and axial directions with visible striations between fuel and moderator regions. The fast group flux is enhanced in fuel regions while the thermal group flux in enhanced in moderator regions. Due to advection the temperature profile in the fuel increases monotonically in the direction of salt flow, while the moderator temperature exhibits a maximum between the mid-plane and outlet. The role of advection is also seen in precursor concentrations. Long lived precursors exhibit maximum concentrations at the core outlet. As the decay constant increases across precursor groups the maximum concentrations moves towards the reactor center where the precursor production rate is maximum. Results from 2D-axisymmetric and 3D models show good qualitative agreement. Moreover, Moltres results compare favorably with the actual design calculations of the MSRE. Moltres demonstrated strong parallel scaling on a typical model problem. Future Moltres publications will highlight transient simulation cases investigating control rod ejection, single channel blockage, loss of flow, and loss of secondary cooling.

6 Acknowledgments

All figures in this paper were generated using the python package yt [37]. The package engauge-digitizer [38] was used to convert rasterized MSRE line plots into point data for plotting alongside Moltres data.

This work was supported by the University of Illinois Department of Nuclear, Plasma, and Radiological Engineering, the National Center for Supercomputing Applications, and the University of Illinois Blue Waters Professors program. Undergraduate researcher Gavin Ridley was supported through NSF REU Site award 1659702, "INCLUSION - Incubating a New Community of Leaders Using Software, Inclusion, Innovation, Interdisciplinary and OpeN-Science."

Finally, the authors would like to thank Matthew Turk and the Data Exploration Lab for co-supporting Dr. Lindsay during his postdoctoral appointment.

Acronyms

ARE Aircraft Reactor Experiment. 1

BOL Beginning-of-Life. 5

BSD Berkeley Software Distribution. 2

CFD Computational Fluid Dynamics. 2

COMSOL COMmon SOLution. 2

LGPL Lesser GNU Public License. 2

LWR Light Water Reactor. 1

MOOSE Multiphysics Object-Oriented Simulation Environment. 2, 4

MPI Message Passing Interface. 20

MSBR Molten Salt Breeder Reactor. 1, 4

MSFR Molten Salt Fast Reactor. 2

MSR Molten Salt Reactor. 1, 2, 6, 21, 22

MSRE Molten Salt Reactor Experiment. 1, 4, 5, 7, 11, 14–18, 22

NCSA National Center for Supercomputing Applications. 4

ORNL Oak Ridge National Laboratory. 1, 16–18

References

[1] GIF. Generation IV International Forum 2008 Annual Report. Technical report, Generation IV International Forum, 2008.

[2] GIF. Generation IV International Forum 2015 Annual Report. Technical report, Generation IV International Forum, 2015.

[3] Roderick A. Hyde and Jon D. McWhirter. Liquid fuel nuclear fission reactor, November 2015.

[4] David Leblanc. Integral molten salt reactor, February 2015.

[5] ThorCon. The Do-able Molten Salt Reactor. Technical report, ThorCon USA Inc, Stevenson, WA, February 2017.

[6] Raluca O. Scarlat, Michael R. Laufer, Edward D. Blandford, Nicolas Zweibaum, David L. Krumwiede, Anselmo T. Cisneros, Charalampos Andreades, Charles W. Forsberg, Ehud Greenspan, Lin-Wen Hu, and Per F. Peterson. Design and licensing strategies for the fluoride-salt-cooled, high-temperature reactor (FHR) technology. *Progress in Nuclear Energy*, 77:406–420, November 2014.

[7] Transatomic Power Corporation. Neutronics Overview. White Paper 1.1, Transatomic Power Corporation, Cambridge, MA, United States, November 2016.

[8] J.R. Engel, H.F. Bauman, J.F. Dearing, W.R. Grimes, H.E. McCoy, and W.A. Rhoades. Conceptual design characteristics of a denatured molten-salt reactor with once-through fueling.

[9] Jiří Křepel, Ulrich Rohde, Ulrich Grundmann, and Frank-Peter Weiss. DYN3d-MSR spatial dynamics code for molten salt reactors. *Annals of Nuclear Energy*, 34(6):449–462, June 2007.

[10] J. Kópházi, D. Lathouwers, and J.L. Kloosterman. Development of a Three-Dimensional Time-Dependent Calculation Scheme for Molten Salt Reactors and Validation of the Measurement Data of the Molten Salt Reactor Experiment. *Nuclear Science and Engineering*, 163(2):118–131, 2009.

[11] Antonio Cammi, Valentino Di Marcello, Lelio Luzzi, Vito Memoli, and Marco Enrico Ricotti. A multi-physics modelling approach to the dynamics of Molten Salt Reactors. *Annals of Nuclear Energy*, 38(6):1356–1372, June 2011.

[12] Carlo Fiorina, Danny Lathouwers, Manuele Aufiero, Antonio Cammi, Claudia Guerrieri, Jan Leen Kloosterman, Lelio Luzzi, and Marco Enrico Ricotti. Modelling and analysis of the MSFR transient behaviour. *Annals of Nuclear Energy*, 64(Supplement C):485–498, February 2014.

[13] S. J. De Zwaan, B. Boer, D. Lathouwers, and J. L. Kloosterman. Static design of a liquid-salt-cooled pebble bed reactor (LSPBR). *Annals of Nuclear Energy*, 34(1):83–92, 2007.

[14] Erik van der Linden. Coupled neutronics and computational fluid dynamics for the molten salt fast reactor. *Delft University of Technology*, 2012.

[15] Matteo Zanetti, Antonio Cammi, Carlo Fiorina, and Lelio Luzzi. A Geometric Multiscale modelling approach to the analysis of MSR plant dynamics. *Progress in Nuclear Energy*, 83:82–98, August 2015.

[16] Manuele Aufiero, Antonio Cammi, Olivier Geoffroy, Mario Losa, Lelio Luzzi, Marco E. Ricotti, and Hervé Rouch. Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis. *Chemical Engineering Science*, 111:390–401, May 2014.

[17] Henry G. Weller, G. Tabor, Hrvoje Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.

[18] A. Laureau, D. Heuer, E. Merle-Lucotte, P. R. Rubiolo, M. Allibert, and M. Aufiero. Transient coupled calculations of the Molten Salt Fast Reactor using the Transient Fission Matrix approach. *Nuclear Engineering and Design*, 316(Supplement C):112–124, May 2017.

[19] Alexander Lindsay. Moltres, a code for simulating Molten Salt Reactors, 2017. https://github.com/arfc/moltres.

[20] GitHub. Build software better, together, 2017.

[21] Paul K. Romano, Nicholas E. Horelik, Bryan R. Herman, Adam G. Nelson, Benoit Forget, and Kord Smith. OpenMC: A state-of-the-art Monte Carlo code for research and development. *Annals of Nuclear Energy*, 82(Supplement C):90–97, August 2015.

[22] William Boyd, Samuel Shaner, Lulu Li, Benoit Forget, and Kord Smith. The OpenMOC method of characteristics neutral particle transport code. *Annals of Nuclear Energy*, 68(Supplement C):43–52, June 2014.

[23] Cameron R. Bates, Elliott Biondo, Kathryn Huff, Kalin Kiesling, Anthony Scopatz, Robert Carlsen, Andrew Davis, Matthew Gidden, Tim Haines, and Joshua Howland. PyNE progress report. *Transactions of the American Nuclear Society*, 111, 2014.

[24] Elliott Biondo, Anthony Scopatz, Matthew Gidden, Rachel Slaybaugh, Cameron Bates, and Paul PH Wilson. Quality assurance within the PyNE open source toolkit. *Transactions of the American Nuclear Society*, 111, 2014.

[25] Derek R. Gaston, Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andrš, Yaqi Wang, Michael P. Short, Danielle M. Perez, Michael R. Tonks, Javier Ortensi, Ling Zou, and Richard C. Martineau. Physics-based multiscale coupling for full core nuclear reactor simulation. *Annals of Nuclear Energy*, 84:45–54, October 2015.

[26] Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254, December 2006.

[27] Satish Balay, Shrirang Abhyankar, Mark Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William Grop, Dinesh Kaushik, Matthew Knepley, Lois Curfman McInnes, Karl Rupp, Barry Smith, Stefano Zampini, and Hong Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.

[28] R. C. Robertson. Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor. Technical Report ORNL–4541, comp.; Oak Ridge National Lab., Tenn., January 1971.

[29] D. L. Zhang, S. Z. Qiu, G. H. Su, and C. L. Liu. Development of a steady state analysis code for a molten salt reactor. 36(5):590–603.

[30] Jaakko Leppänen, Maria Pusa, Tuomas Viitanen, Ville Valtavirta, and Toni Kaltiaisenaho. The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150, August 2015.

[31] Mark D. DeHart and Stephen M. Bowman. Reactor Physics Methods and Analysis Capabilities in SCALE. *Nuclear Technology*, 174(2):196–213, May 2011.

[32] R. C. Robertson. Msre Design and Operations Report. Part I. Description of Reactor Design. Technical Report ORNL-TM-728, Oak Ridge National Lab., Tenn., January 1965.

[33] Paul N. Haubenreich and J. R. Engel. Experience with the Molten-Salt Reactor Experiment. *Nuclear Technology*, 8(2):118–136, February 1970.

[34] R. B. Briggs. Molten-Salt Reactor Program semiannual progress report for period ending July 31, 1964. Technical Report Archive and Image Library ORNL-3708, Oak Ridge National Laboratory, Oak Ridge, TN, United States, 1964.

[35] John A. Turner, Kevin Clarno, Matt Sieger, Roscoe Bartlett, Benjamin Collins, Roger Pawlowski, Rodney Schmidt, and Randall Summers. The virtual environment for reactor applications (VERA): Design and architecture. 326:544–568.

[36] Matthias Heil, Andrew L Hazel, and Jonathan Boyle. Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.

[37] Matthew J. Turk, Britton D. Smith, Jeffrey S. Oishi, Stephen Skory, Samuel W. Skillman, Tom Abel, and Michael L. Norman. yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data. *The Astrophysical Journal Supplement Series*, 192(1):9, January 2011.

[38] Mark Mitchell, Baurzhan Muftakhidinov, Tobias Winchen, Zbigniew Jędrzejewski-Szmek, The Gitter Badger, and badshah400. markummitchell/engauge-digitizer: Version 10.1 Export Improvements. Technical report, Zenodo, July 2017. DOI: 10.5281/zenodo.832444.