

Enabling High Fidelity Neutron Transport Simulations on Petascale Architectures

Dinesh Kaushik*, Micheal Smith, Allan Wollaber, Barry Smith, Andrew Siegel, Won Sik Yang

Argonne National Laboratory

Argonne, IL 60439 USA

* corresponding author:kaushik@mcs.anl.gov

Abstract

The UNIC code is being developed as part of the DOE's Nuclear Energy Advanced Modeling and Simulation (NEAMS) program. UNIC is an unstructured, deterministic neutron transport code that allows a highly detailed description of a nuclear reactor core in our numerical simulations. The goal of our simulation efforts is to reduce the uncertainties and biases in reactor design calculations by progressively replacing existing multi-level averaging (homogenization) techniques with more direct solution methods based on first principles. Since the neutron transport equation is seven dimensional (three in space, two in angle, one in energy, and one in time), these simulations are among the most memory and computationally intensive in all of computational science. To model the complex geometry of a reactor core, billions of spatial elements, hundreds of angles, and thousands of energy groups are necessary, which leads to problem sizes with petascale degrees of freedom. Therefore, these calculations exhaust memory resources on current and even next-generation architectures. In this paper, we present UNIC simulation results for two important representative problems in reactor design/analysis - PHENIX and ZPR. In each case, UNIC shows excellent weak scalability on up to 163,840 cores of BlueGene/P (Argonne) and 131,072 cores of XT5 (ORNL). While our current per processor performance is not ideal, we demonstrate a clear ability to effectively utilize the leadership computing platforms. Over the coming months, we aim to improve the per-processor performance while maintaining the high parallel efficiency by employing better algorithms (such as spatial p-refinement, optimized matrix-tensor operations, and weighted partitioning for load balancing). Combining these additional algorithmic improvements with larger parallel machines in the near future should allow us to realize our long term goal of explicit geometry coupled multiphysics reactor simulations. In the long run, these high fidelity simulations will be able to replace expensive mockup experiments and reduce the uncertainty in crucial reactor design and operational parameters.

I. Introduction

Nuclear engineering has a rich history of simulation-based design following sound economical and safety-driven principles. However, many of the modern reactor modeling codes were developed in the seventies and eighties and targeted serial platforms because of the high computational costs of explicit geometry approximations. In this paper, we describe the development of a new reactor analysis code that bridges the gap between the approximation-based legacy tools and a first-principles approach. The code we discuss in this paper is specifically targeted for applications where the legacy tools are least reliable, and its development is only possible given the availability of the large scale parallel machines.

The performance of nuclear power reactors is governed by the fission rate of the uranium based fuel. A predictive analysis capability is generally required to optimize the safety characteristics of the reactor and minimize the costs associated with operating the reactor. This analysis capability is derived from the solution of a Boltzmann integro-differential transport equation for the neutron density. This equation is widely used in industrial and scientific industry and appears in atmospheric modeling, astrophysical and nuclear weapons research, medical physics, and industrial applications such as mineral assaying and oil-well logging. Among these fields, the most significant parallelization efforts to date have been applied to the thermal radiative transport equation (gamma and x-rays) used in weapons related research, and researchers have utilized several “top” supercomputers to perform simulations. Unfortunately, many of the modeling challenges that arise in the thermal radiative and the neutron transport equations are sufficiently different that direct technology transferability between the codes is impractical.

Our focus is on nuclear reactor systems that consider a large distributed fission source, where the primary unknown of the Boltzmann transport equation is the neutron density or, in nuclear engineering vernacular, the neutron “flux” (density multiplied by velocity). This equation has seven independent variables: three in space, two in angle, one in energy, and time. Because an accurate, first-principles discretization of these variables is untenable, legacy solvers are typically based on approximations that reduce this dimensionality. The neutron transport equation can also be shown to asymptotically limit to the canonical hyperbolic, elliptic, and parabolic partial differential equation forms under simple changes in material properties that may all occur in a nuclear reactor. In thick, highly-scattering regions, the transport equation limits to a (parabolic) time-dependent diffusion equation, which, in steady-state, is elliptic. In

“free-streaming” regions, the limiting behavior is hyperbolic. Thus, the large dimensionality and many-faceted solution behaviors for this equation present the greatest challenges to the code developer.

Our focus is on immediate improvement to areas where legacy solvers are insufficient: nuclear reactor dynamics. These problems require the solution of the time-dependent Boltzmann transport equation and the simultaneous solution of the thermal-hydraulic and structural-mechanics equations [1]. Two years ago we started a multi-year development project to create a dynamics solver capability using the open science high performance computing resources at Argonne National Laboratory (IBM BlueGene/P) and Oak Ridge National Laboratory (Cray XT5) [2-4]. The initial condition for this formulation requires the solution of a time-independent k-eigenvalue equation [1] that is the focus of this manuscript. We note that with a linear implicit time formulation, all subsequent solutions at the end of each time step exhibit very similar requirements to that needed to solve the initial k-eigenvalue problem. We focus on the recent success we have had with the SN2ND solver [5], which solves the second-order even-parity formulation of the neutron transport equation.

II. Neutron Transport Simulation Complexity

The primary issue in nuclear reactor analysis is the sheer scale of the problem to be solved. In this regard, we have thus far limited our dynamics solver development to sodium cooled, fast reactor designs, as these reactors have been proposed as an alternative to reduce the volume of spent fuel disposition (i.e. fission the fuel rather than store it indefinitely) and the nuclear industry has insufficient engineering experience. With time, we will also apply our reactor analysis tools to more prevalent pressurized water reactor (PWR), boiling water reactor (BWR), and the Canadian deuterium uranium (CANDU) reactors [6] to help address the smaller pool of unanswered questions that these thermal reactor designs pose.

We begin with the spatial domain of a typical sodium cooled fast reactor, some examples of which are shown in Figure 1. For neutron transport, we can limit our focus to just the “core” of the reactor shown in the center picture of Figure 1, the scale of which, relative to the plant, can be inferred from the rightmost picture in Figure 1. The core is typically made up of about two to five hundred ducted fuel assemblies similar to those depicted in the left picture of Figure 1. The assemblies are composed of many (60-300) fuel pins. Radially, the core is built of fuel assemblies that form a rough cylinder, leading to a total modeling diameter of between two to six meters and a height of three to five meters. This core size and the spatial

heterogeneity of the fuel assemblies require approximately half a billion to a billion finite elements to accurately represent the spatial heterogeneity and the associated spatial gradients in the neutron density.

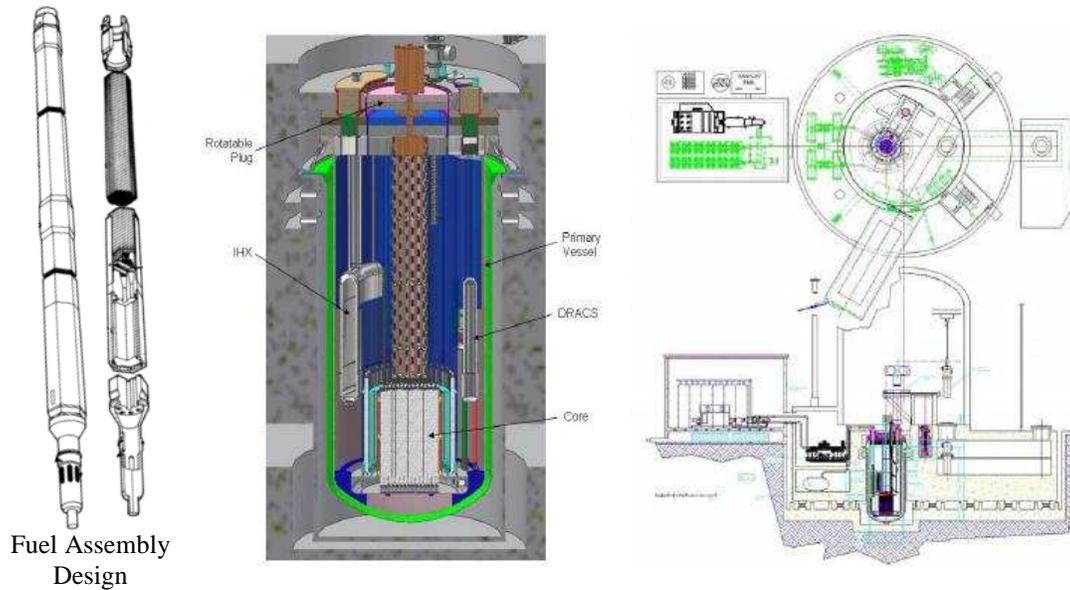
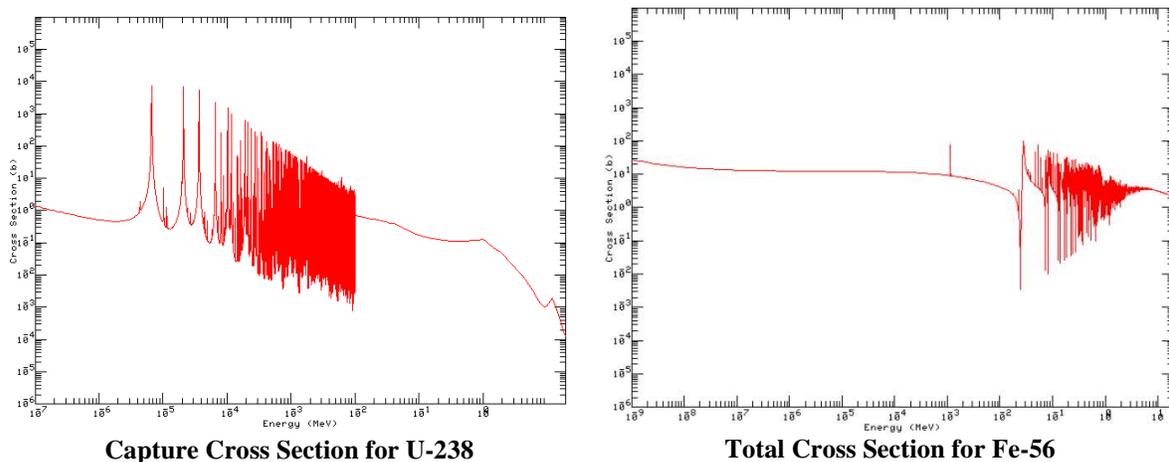


Figure 1. Fuel Assembly, Reactor, and Plant Schematics of Sample Sodium-Cooled Fast Reactor

Next we consider the energy and angular requirements because they are tightly coupled. Neutrons lose fractions of their energy by scattering with materials, and the amount of energy loss per scattering depends on the scattering material and angle. Figure 2 shows the “cross section” data (roughly, the probabilities of interaction versus neutron energy) for the Uranium 238 and Iron 56 isotopes that constitute two of the largest components of a sodium cooled fast reactor. Note that these are on a log-log scale. Most other isotopes present in a nuclear reactor have cross section representations with similar complexity with widely-varying energy dependencies.



Capture Cross Section for U-238

Total Cross Section for Fe-56

Figure 2. Cross section Data for U-238 and Fe-56.

The large amount of material heterogeneity in the geometry combined with the severity of the energy dependence in the cross section data leads to flux distributions of comparable complexity to the space-energy distribution of the cross section data. As such, it is entirely impractical to use a smooth polynomial functional representation in energy, and all historical and modern energy discretizations employ a “multi-group” (0th order finite element in energy) flux representation [1] by utilizing “effective” multi-group constants. Additionally, because neutron scattering couples the energy and angle terms, with the rapid changes in the energy dependence of the cross sections seen in Figure 2 come rapid variations in the magnitude of the flux in the angular variable. When combined, we estimate that a first-principles approach will require 100,000 energy groups and 1,000 angles (collocation or 0th order finite elements on the sphere), which leads to approximately 10^{17} degrees of freedom in space, energy, and angle for each time step. Thus, even on today’s supercomputers, some form of approximation is necessary to obtain solutions.

III. Neutron Transport Equation for Reactor Analysis

Fortunately, we only need a solution that meets the requirements of the engineering analysis, and thus simplifications can be implemented. The first and most important simplification reduces the demands of the energy representation. To do this we implement multi-level modeling and simplification steps, the details of which are beyond the scope of this paper, but whose purpose is to produce a set of coarse group cross section data that preserve key neutron reaction rates in each energy group [1]. These approximations rely upon substantial experience on particular reactor systems and experimental validations of the predictive abilities of the legacy tools. As we develop our code, the significance of these approximations will diminish relative to the legacy approaches, as we will enable a better matching of the “reference” configuration used in the coarse group cross section data to the system at hand. The end result is that we reduce the need for 100,000 groups to much less than 2,000 groups, with a general ability to use less than 100 groups for most analyses (2 groups are typically used in most industry reactor analysis codes). In this work, we use a 33 group approximation that pushes our current solver to the limits of the available memory on BlueGene/P.

We next consider the time discretization. The average speed and multiplication time of neutrons in a reactor core is such that time steps on the order of milliseconds are needed for rapid transient scenarios such as a control rod ejection accident, making the neutronics component the “stiff” part of the overall multi-physics system. The duration of the simulated accident varies from hours to days, which makes the

time spent in the neutron transport solver the limiting simulation factor. Most modern legacy tools avert this problem by using a point kinetics (space-angle-energy independent) model or a few energy-group diffusion theory methodology on a structured geometry grid [1]. With the improvements in the energy approximation and the use of transport rather than diffusion theory, we expect to significantly improve the accuracy and fidelity of the safety analysis modeling for these simulations.

III.A. Overview of the Development of Parallel Tools

Unstructured mesh deterministic methods use the multi-group approximation in energy combined with either a hybrid finite element or a continuous finite element decomposition in space. Historical angle discretization schemes include spherical harmonic (polynomial) expansions, finite element, or angle collocation (0th order finite element known as discrete ordinates) [1]. To date, most parallelization efforts in neutron transport have focused on improvement of the structured geometry discrete ordinates solvers [7-9] with moderate to good success on small to medium-range parallel machines, although it is difficult to find performance data for these tools on more than 2048 processors. Unfortunately, these tools are not useful given the unstructured geometries required in a coupled physics environment. While some might argue that we can impose local homogenization rules, this merely substitutes one problematic legacy approach with another (albeit better) one. Even if we were to take that approach, it would seem wiser to just use the legacy transport tools based upon assembly homogenization [10], as those tools can easily execute on serial process machines. In addition to our own work, there has been substantial research on unstructured methodologies [11-13], but these codes are either not obtainable, not set up for the specific needs of reactor analysis, or not proven for large geometry heterogeneous applications on massively parallel machines.

We also remark briefly that large scale parallelization of the Monte Carlo method for “embarrassingly parallel” calculations has been particularly effective. However, the dynamics problems that we are targeting include massive memory requirements that prevent each processor from accessing the full space-energy representation of the problem in the Monte Carlo method. The solution is to use domain decomposition in the Monte Carlo algorithm which severely impacts its scalability. Also, Monte Carlo solutions can contain stochastic uncertainties on the order of the expected perturbations from the thermal-structural feedback effects. Thus, a massively parallel deterministic solver for dynamics problems truly fills a gap in the available predictive capabilities of modern neutron transport tools.

IV. UNIC: Modernization and Development of Reactor Analysis Tools

In this section we present an overview of our solution algorithm to the time-independent (the initial condition) neutron transport equation. We also discuss algorithmic choices we have made to reduce execution times by cutting down on extraneous floating point work and maintaining parallel scalability.

IV.A Neutron Transport Equation

The multigroup form of the neutron transport equation consists of G equations with $1 < g < G$:

$$\hat{\Omega} \cdot \vec{\nabla} \psi_g(\vec{r}, \hat{\Omega}) + \Sigma_{t,g}(\vec{r}) \psi_g(\vec{r}, \hat{\Omega}) = S_g(\vec{r}, \hat{\Omega}). \quad (1)$$

$\psi_g(\vec{r}, \hat{\Omega})$ is the group neutron angular flux and $\Sigma_{t,g}(\vec{r})$ is the total cross section (sum of all reaction probabilities). Thus, the first term is a streaming/leakage term, and the second is a collision removal term.

The system of equations is coupled via the source $S(\vec{r}, \hat{\Omega})$ which we expand in terms of group-to-group scattering and fission as

$$S_g(\vec{r}, \hat{\Omega}) = \sum_{g'=1}^G \int \Sigma_{s,g \rightarrow g}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') \psi_{g'}(\vec{r}, \hat{\Omega}') d\hat{\Omega}' + \frac{1}{k} \chi_g \sum_{g'=1}^G \nu_{g'}(\vec{r}) \Sigma_{f,g'}(\vec{r}) \int \psi_{g'}(\vec{r}, \hat{\Omega}') d\hat{\Omega}'. \quad (2)$$

Here, k is the system eigenvalue, also known as its effective multiplication factor. The scattering source in Eq. (2) redistributes neutron energies and angles in an anisotropic way, while the fission source redistributes neutrons into the isotropic fission spectrum χ .

Based upon the parallelization successes of other authors with the Poisson equation, we focused part of our initial development on second-order methodologies that implement continuous spatial finite element approximations such that we can take advantage of parallel conjugate gradient methods. To obtain the second-order discrete ordinates formulation used in SN2ND, we expand the angular flux in Eq. (1) into even-parity, $\psi_g^+(\vec{r}, \hat{\Omega})$, and odd-parity, $\psi_g^-(\vec{r}, \hat{\Omega})$, components

$$\psi_g(\vec{r}, \hat{\Omega}) = \psi_g^+(\vec{r}, \hat{\Omega}) + \psi_g^-(\vec{r}, \hat{\Omega}). \quad (3)$$

We then rewrite Eq. (1) using Eq. (3) to get the first order even-parity and odd-parity equations

$$\hat{\Omega} \cdot \vec{\nabla} \psi_g^\pm(\vec{r}, \hat{\Omega}) + \Sigma_{t,g}(\vec{r}) \psi_g^\pm(\vec{r}, \hat{\Omega}) = S_g^\pm(\vec{r}, \hat{\Omega}). \quad (4)$$

Assuming a discontinuous finite element approximation in the odd-parity flux, we solve for the odd-parity flux and substitute it into the even-parity equation to obtain the second-order even-parity transport equation.

$$-\hat{\Omega} \cdot \vec{\nabla} \frac{1}{\Sigma_{t,g}(\vec{r})} \hat{\Omega} \cdot \vec{\nabla} \psi_g^+(\vec{r}, \hat{\Omega}) + \Sigma_{t,g}(\vec{r}) \psi_g^+(\vec{r}, \hat{\Omega}) = S_g^+(\vec{r}, \hat{\Omega}) - \hat{\Omega} \cdot \vec{\nabla} \frac{1}{\Sigma_{t,g}(\vec{r})} S_g^-(\vec{r}, \hat{\Omega}) \quad (5)$$

We next weight Eq. (5) with a set of spatial trial functions $f(\vec{r})$, integrate over volume, and apply the divergence theorem to the first term such that we obtain the natural vacuum boundary condition term on the surfaces of the domain boundary.

$$\int dV \hat{\Omega} \cdot \vec{\nabla} f(\vec{r}) \frac{1}{\Sigma_{t,g}(\vec{r})} \hat{\Omega} \cdot \vec{\nabla} \psi_g^+(\vec{r}, \hat{\Omega}) + \int dV f(\vec{r}) \Sigma_{t,g}(\vec{r}) \psi_g^+(\vec{r}, \hat{\Omega}) + \int d\Gamma f(\vec{r} \in \Gamma) \left| \hat{\Omega} \cdot \hat{n} \right| \psi_g^+(\vec{r} \in \Gamma, \hat{\Omega}) = \int dV f(\vec{r}) Q_g^+(\vec{r}, \hat{\Omega}) \quad (6)$$

Finally, we implement a continuous finite element formulation for the even-parity flux. The coefficient matrix produced from the terms on the left side of Eq. (6) can be shown to be symmetric positive definite and thus suitable for the conjugate gradient methodology, provided that the terms on the right side of Eq. (6) are “lagged” in an iterative approach to be discussed below.

IV.B. SN2ND Solver Implementation

As mentioned, the spatial approximation is treated via a standard continuous finite element method, and we employ classic domain decomposition where weights can be applied to the vertices to balance the local work with the communication costs required to connect the domain. In angle, we chose the discrete ordinates approximation, which requires us to define a set of directions on the unit sphere. With regard to parallelization, we employ the generic decomposition of (S)pace, (A)ngle, and (G)roup shown in Figure 3.

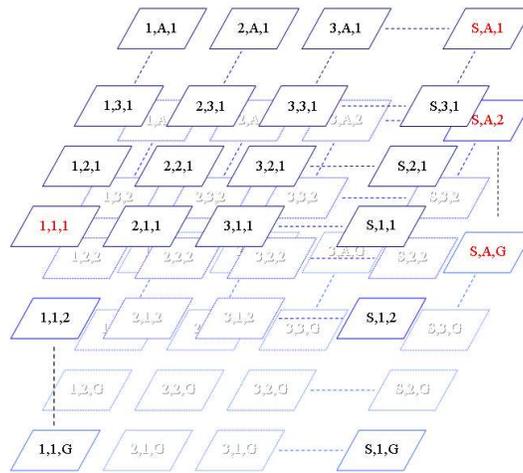


Figure 3. Space, Angle, Group Decomposition for a Parallel Machine.

In this approach, each MPI process sees four communicators: space, angle, group and the global

communicator. The advantage of this approach is that the group and angle communication does not overlap with respect to space, and thus the communication in these two directions can be done simultaneously on the parallel machine.

When a discrete ordinates approximation is applied to Eq. (6), we find that, for each group, the set of angular equations are only coupled via the source term on the right side of Eq. (6) termed the within group source. This equation is typically cast into scattering iterations where the scattering source is lagged in iteration [1,2]. The equation is accelerated by solving a synthetic diffusion equation for the angle-integrated (scalar) flux, which is essentially a multi-grid preconditioner in angle [5]. Thus, a “scattering iteration” of SN2ND involves solving 100 diffusion-like equations (assuming 100 directions in the angular cubature) simultaneously to obtain the angular discrete ordinates flux for each group. These equations are currently solved using a parallel SOR methodology available in PETSc [17], although we are now moving towards developing a multi-grid preconditioner. To update the source (or perform a synthetic acceleration step) on the right side of Eq. (6), we then collect the information on the angular communicator of each process. This requires a global reduce operation for the locally visible spatial mesh partition for each group (simultaneous communication on group and space communicators if fully partitioned in energy).

In our current implementation, we do not consider parallelization by group because we can already saturate the available parallel machines with our space-angle parallelization scheme. However, this means our memory requirements are linear with respect to the number of energy groups, which can be problematic on low memory machines like BlueGene/P [3]. In our current solver, we can distribute any number of angles on a given process and generally have found that two to three angles per process works best. With regard to solving the synthetic acceleration equation, we have currently assigned the first process on each angular communicator to again utilize the parallel SOR algorithm in PETSc, which introduces a load imbalance by angle parallelization.

The steady state transport equation shown in Eqs. (1) and (2) requires an eigenvalue search procedure to obtain k and the associated flux vector. The gold standard for all modern neutron transport codes is to use inverse power iteration [1] as it minimizes the amount of effort required to find the dominant eigenvalue. Assembling all of the group and direction equations derived from Eq. (6), we write

$$A\psi^+ = B\psi^+ + \frac{1}{k}F\psi^+ \rightarrow \psi^+ = \frac{1}{k}T^{-1}F\psi^+ \rightarrow F\psi^+ = \theta = \frac{1}{k}FT^{-1}\theta, \quad (7)$$

where A is the coefficient matrix, B is the scattering source operator, and F is the fission source operator. The power (or outer) iteration methodology finds the dominant k eigenvalue using the following recurrence relationships

$$\theta^{(i)} = \frac{1}{k^{(i-1)}}FT^{-1}\theta^{(i-1)}, \quad k^{(i)} = k^{(i-1)} \frac{\langle w, \theta^{(i)} \rangle}{\langle w, \theta^{(i-1)} \rangle}. \quad (8)$$

In SN2ND, we currently use the Gauss-Seidel method to iteratively invert T during each outer iteration in Eq. (8), because for fast reactors only a single iteration is required for convergence (because neutrons only lose energy during scattering events over the energy range of interest in fast reactors, the energy coupling is lower triangular). For time dependent problems and thermal reactor calculations, a Gauss-Seidel scheme is not expected to be as efficient, and we intend to use a more general Krylov method with our current Gauss-Seidel scheme as a preconditioner. We note that a Krylov solver will also assist in making the above methodology scalable in energy for time dependent problems.

We note that this approach does not require T to be exactly inverted at each outer iteration. Instead, we only require that the error in the flux vector in Eq. (7) be slightly lower than the error in the fission source vector. We implemented an optimized scheme to account for this behavior and combined it with conventional Tchebychev acceleration [1,2]. Together, these approaches have allowed us to significantly reduce the overall time to solution. Figure 4 shows the impact of making these optimization changes on the C5G7 benchmark [18] where the outer iteration eigenvalue, fission source, and flux vector are plotted in addition to the within group flux error for each energy group. In Figure 4, the un-accelerated approach takes roughly twice the number of outer iterations as the Tchebychev accelerated one. More importantly, the effort spent on solving the within group flux equations for each outer iteration is substantially reduced in the optimized version (the targeted flux error obtained for each group flux at each iteration is relaxed). In practice, this makes the effort spent on each outer iteration nearly constant, although the dynamic error adjustments can introduce variability in the total solution time from problem to problem by requiring slightly different numbers of outer iterations as will be seen in the numerical results.

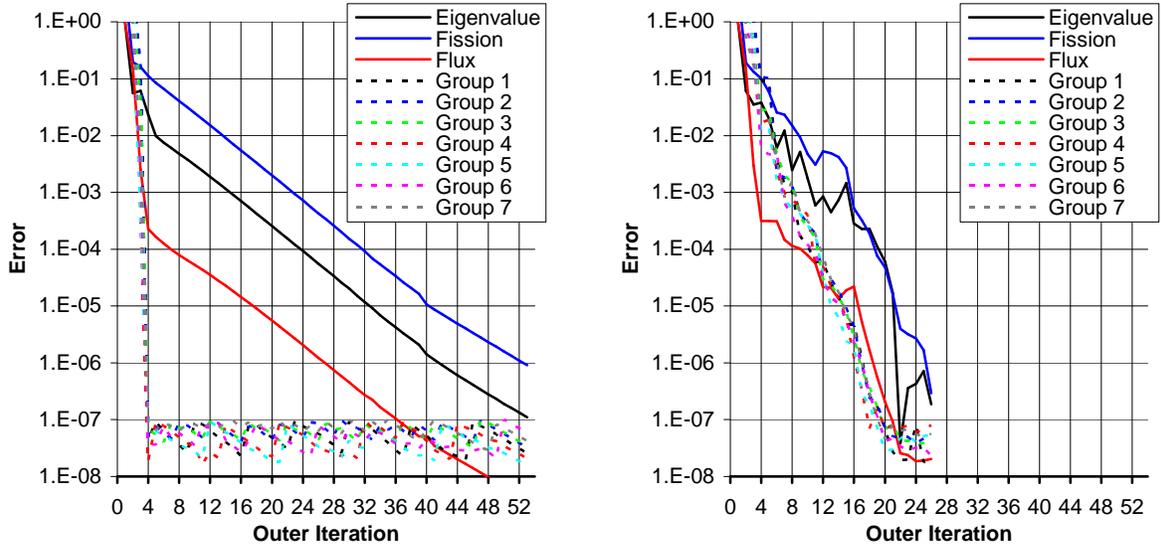


Figure 4. Optimization and Tchebychev Acceleration Impact on the C5G7 Benchmark.

V. Problems Chosen For Study

Two reactor problems have been chosen to demonstrate the performance of the SN2ND solver. Both problems consider the steady state eigenvalue solution, the initial condition for the time dependent problems that we will be studying in the near future. Also, both problems cannot be solved well using existing homogenization methodologies. For both problems, we present weak scalability as we increase the number of angles on the entire BlueGene/P machine at Argonne and most of the XT5 machine at ORNL (no issues are expected in using all 150,000 processors of this machine, but time constraints prevented an attempt at this). These weak scaling studies are highly relevant to our work to assess the impact of the angular discretization on the accuracy of the eigenvalue (as well as the flux solution, although the eigenvalue is more convenient for reporting). As mentioned, the SN2ND solver allows for parallelization by direction, and we generally have found it best to use two or three directions per process to balance the communication costs with the computational burden. However, with the memory limitations on BlueGene/P and our desire to obtain comparable results on XT5, we executed the SN2ND solver with only a single angle per process. As will be shown, the time required to update the scattering source in Eq. (6) is a substantial amount of computational effort and we have not yet optimized it for performance. We emphasize that the SN2ND solver (as part of UNIC) is in the early stages of development, and its focus has been on demonstrating the feasibility of higher fidelity reactor core simulations on large-scale parallel machines. These simulations have helped identify several performance issues that, when addressed, will

substantially improve the per-processor performance of the SN2ND solver (potentially an order of magnitude or more).

V.A PHENIX End of Life Experiments

The first problem is taken from the end of life experiments of the PHENIX reactor [19]. A solution using UNIC is desired because the legacy solvers (based on conventional homogeneous approaches) have difficulty in representing various control rod configurations accurately. In this benchmark, only the control rod assemblies are represented heterogeneously. This type of spatial representation is directly relevant in that our initial time dependent calculations will also focus on representing only part of the geometry heterogeneously. Figure 5 depicts a slice of the PHENIX core center along with a typical unstructured mesh (prisms) and the flux solution at two important energy groups in the lower part of the control rod assembly created using VISIT [20]. We note that the solutions obtained with SN2ND are, to the best of our knowledge, the most reliable means of obtaining the correct solution compared with all other modern deterministic solution methods.

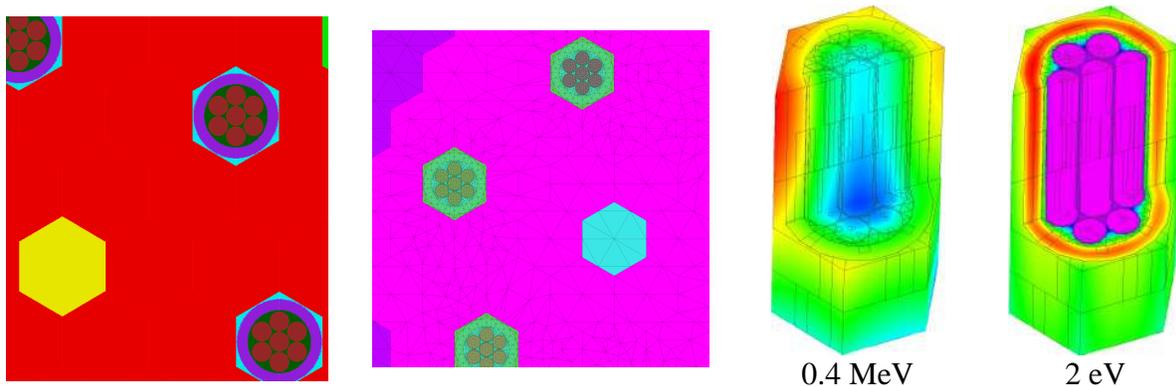


Figure 5. Planar Configuration of PHENIX Geometry Model and Flux Solution

At present, we have not performed all of the requested calculations desired by the benchmark authors, nor have we done an essential multi-group accuracy assessment; only space and angle studies have been performed. We chose a standard 33-group cross section set with a P_3 expansion of the scattering kernel which we typically use for homogenous problems. Because the problem specification was originally intended for homogeneous problems (which is indicative of the simulation capabilities of most available tools), we had to construct a specification appropriate for heterogeneous geometries. Unfortunately, we made a geometric error in the thermal expansion of the control rods that compromises the accuracy of our results, although the code performance is unaffected. We intend to resolve any remaining geometry and

cross section uncertainties for the final paper or a later publication. Using CUBIT [21], we created meshes considering different degrees of radial mesh refinement (three levels) and axial mesh refinement (three levels), leading to a total of nine meshes. Our simulations demonstrated that the medium level approach for both the radial and axial directions was sufficiently accurate. This mesh contains 284,682 quadratic LaGrangian prismatic elements and 1,741,833 spatial vertices.

In Table 1, we present the weak scaling results we achieved using SN2ND on BlueGene/P. We partitioned the mesh using MeTiS [22] into 2,048 pieces leading to ~850 vertices per process, which is near the minimum that we can use with the parallel SOR algorithm in PETSc (below this, communication overhead increases substantially on both machines). As we increase the number of angles (note that the even-parity formulation only requires the half-sphere set of angles or 2π), we make a corresponding increase in the number of processors. The eigenvalue rapidly converges as the number of directions is increased, which is expected given that a majority of the domain is homogenized. An initial glance indicates a drop in weak scaling to 75% on the entire machine; however, the number of “Fission” (outer) iterations needed to solve Eq. (8) is correspondingly seen to increase as well. It is important to understand that the number of fission source iterations required for convergence varies naturally depending upon both the space-angle-group approximation, the effectiveness of Tchebychev acceleration on the fission iterations, and the peculiarities of our optimization scheme (for adapting tolerances on each fission source iteration), and that none of these are tied to the overall parallelization of the algorithm, which is primarily focused on space-angle distributed work of the scattering iterations. If we normalize the time to solution based upon the number of fission iterations, then we can claim to achieve 97% weak scalability with 128 angles on 131,072 cores of XT5 and 88% weak scalability with 160 angles on 163,840 cores of BlueGene/P.

Table 1. Weak Scaling Study by Angle for the PHENIX Problem on BlueGene/P

Cores	4π Angles	k_{eff}	Fission Iters. / Time	Total Time (sec)	Source Update (sec)	Weak Scaling
32,768	32	0.96006	23 / 152	3493	2934	100%
49,152	48	0.96004	23 / 152	3510	2933	100%
65,536	64	0.96007	23 / 153	3526	2934	99%
73,728	72	0.96015	23 / 156	3593	2934	97%
131,072	128	0.96019	27 / 156	4209	3437	83%*
163,840	160	0.96019	27 / 173	4676	3436	75%*

* See text for discussion on “effective” weak scalability

Table 2 shows the weak scaling results on up to 131,072 cores of XT5, as well as the “effective” weak scaling numbers that are generated by normalizing to 25 fission iterations. The larger memory of the XT5 machine also allowed us to use our finest mesh, which contains 833,280 quadratic LaGrangian prismatic elements and 4,017,189 vertices.

Table 2. Weak Scaling Study by Angle for the PHENIX Problem on XT5

Cores	4π Angles	k_{eff}	Fission Iters. / Time	Total Time (sec)	Source Update (sec)	Weak Scaling	Effective Weak Scaling
32,768	32	0.96017	25 / 63	1574	851	100%	100%
49,152	48	0.96014	22 / 64	1399	746	112%	99%
65,536	64	0.96017	22 / 64	1402	745	112%	99%
98,304	96	0.96017	25 / 65	1623	847	97%	97%
114,688	112	0.96017	26 / 65	1687	882	93%	97%
131,072	128	0.96029	28 / 68	1902	948	83%	93%

Similar to the previous case, the number of fission source iterations varies as we change the space-angle-group approximation (previously we saw between 23 to 27 on BlueGene/P, here we see 22 to 28 iterations). We note that the total time to solution drops substantially for XT5 (which is expected since a XT5 processor is about three times faster in clock frequency), and the ratio of updating the source to total time drops from ~80% on BlueGene/P to ~50%. As a final note, the largest problem we executed on BlueGene/P in Table 1 solved 9.2 billion degrees of freedom, while the largest problem on XT5 in Table 2 solved 17 billion degrees of freedom.

V.B Zero Power Reactor 6 Experiment 6A

The other problems we chose to simulate for our project this year are the Zero Power Reactor (ZPR) experiments 6a and 7 [23]. The ZPR experiments were performed to acquire fundamental data on nuclear reactor designs of interest. These particular experiments focused on uranium fueled (ZPR-6/6a) and plutonium fueled (ZPR-6/7) sodium cooled fast reactor systems. Eventually, the SN2ND simulations of these experiments will be used to help validate the code and better ascertain the approximation errors in legacy approaches by enabling direct comparisons of computed results. While the experiment ZPR-6/7 has more data and is our preference, we have only set up the 6a experiment at present. For scoping studies, we also reduced the number of unique fuel drawer types and introduced asymmetries into the geometry to investigate local flux heterogeneities. In essence, these results provide a preliminary assessment of the

space-angle requirements of these experiments and have allowed us to identify the parts of the solver that need improvement before the entire problem can be simulated.

Figure 6 provides two pictures of the explicit geometry model (left, center) and the power distribution for the enriched U-235 plates in the back matrix assembly (right). A gray color in Figure 6 is used for the matrix tube and drawer fronts that are loaded into each tube position. The solid green squares are two inch depleted uranium blocks directly loaded into the tubes surrounding the main core and act as a blanket. We separated the matrix assemblies, withdrew one of the drawers from the front matrix assembly, and pulled a section of the plates out to give a better perspective on the overall geometry.

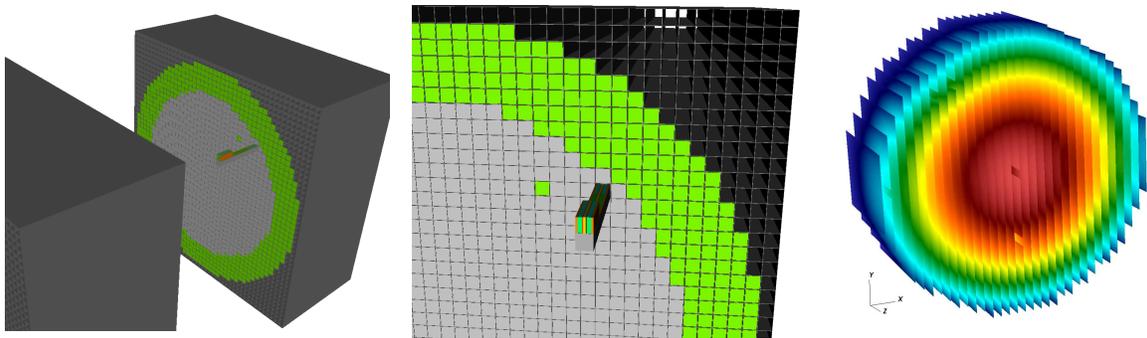


Figure 6. ZPR-6 Experiment 6A Geometry and U-235 Plate Power Solution.

The fully-explicit geometry is rather difficult to solve because of the large number of material boundaries as indicated by the left picture in Figure 7. Also, for even-parity methods, the extremely small voids separating the plates and various other components are unacceptable, and we have made the geometric simplifications shown in Figure 7 (center), which we expect to refine with time.

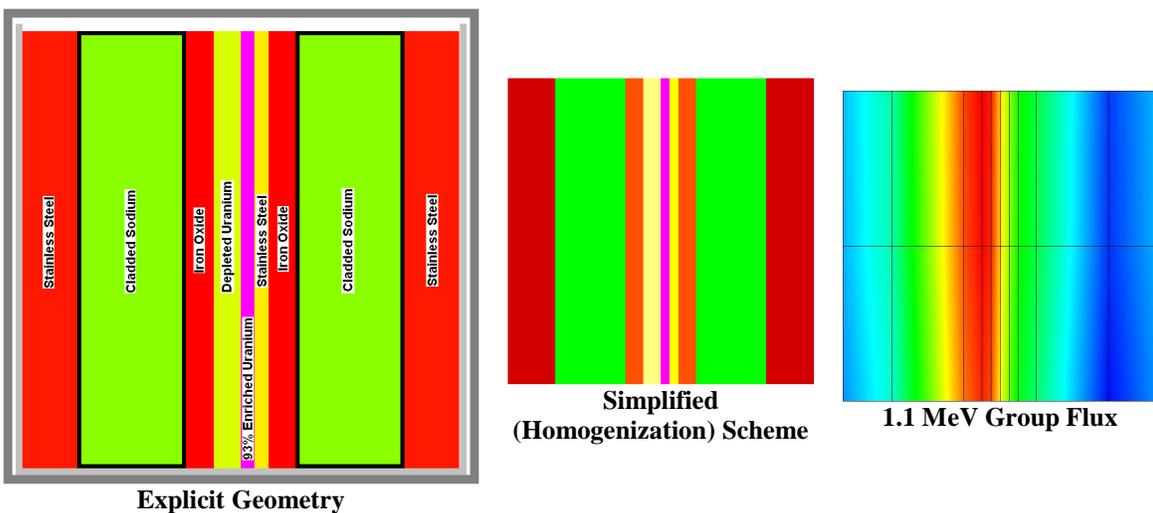


Figure 7. Fuel Drawer Model for Initial ZPR-6 Assembly 6A Benchmark

We note that in all legacy homogenization approaches, the plates in each drawer are effectively “mixed together”, which makes the observed plate-power distribution generated using SN2ND in Figure 6 (right) and the 1.1 MeV flux distribution in an axial slice of the center-most drawer shown in Figure 7 (right) impossible to obtain.

Even with the simplification shown in Figure 7 (from left to center), we generate quadratic finite element meshes with upwards of 20 million vertices that are currently beyond the memory on BlueGene/P. Thus we employed linear meshes for our initial scoping and parallel scalability studies, which we feel are indicative of the computational effort in the preconditioning phase that we will expect to perform once spatial p-refinement is in place. We constructed a coarse mesh with 545,664 hexahedral elements and 575,168 vertices to use on BlueGene/P, and a more refined mesh with 2,058,264 hexahedral elements and 2,119,175 vertices to use on XT5. Switching these meshes to quadratic typically increases the number of vertices by a factor of 8 to 9, and the more refined mesh is what we expect to need for the homogenization approach shown in Figure 7.

For brevity, we only provide weak angle scaling results for the XT5 machine as Table 3, although we note that we also successfully simulated this problem on 163,840 cores of BlueGene/P. We split the mesh into 512 pieces with ~4139 vertices per process.

Table 3. Weak Scaling by Angle and Flop Rate for the ZPR 6/6a Problem on XT5

Cores	4π Angles	k_{eff}	Fission Iters. / Time	Total Time (sec)	Eff. Weak Scaling	Aggregate Tflop/s	% of Machine Peak	% of Memory Bandwidth Bound
8,192	32	1.00542	20 / 37	745.0	100%	5.3	7.1	43
16,384	64	1.00633	17 / 38	647.9	98%	10.5	6.9	42
32,768	128	1.00632	23 / 42	968.7	88%	19.2	6.4	39
65,536	256	1.00761	27 / 45	1204.6	83%	36.6	6.1	37
131,072	512	1.00761	29 / 50	1441.3	75%	67.9	5.6	34

While it appears as though the eigenvalue is inconsistently converging, this is an artifact of the Legendre-Tchebychev cubature. With the Legendre-Tchebychev cubature, the number of directions defined perpendicular to the plate geometry can be set independently of those defined parallel to the plate geometry. Because of the monolithic plate orientation we see a correspondingly strong heterogeneity when we increase the number of directions perpendicular to the plates. Even going from 512 to 640 angles on BlueGene/P provided a change of 0.03% in the eigenvalue (which is significant in the reactor physics

community) indicating this explicit geometry problem has a solution with a much stronger angular dependence than the preceding PHENIX problem.

In Table 3, we again have good weak scaling, although we do see a drop in performance as the number of angles is increased, which is consistent with the behavior seen in PHENIX on the preceding results for both BlueGene/P and XT5. At present we are not certain whether this is a load imbalance in angle due to the heterogeneity (some directions can require more effort to solve than others) or whether this is an artifact of the machine network (we are performing a global reduce operation of a significantly sized vector on the angle communicator which appears to lose scalability on XT5).

In addition to the scaling data in Table 3, we also indicate the floating point performance. Our current preconditioners heavily rely upon a sparse-matrix conjugate gradient solver in PETSc. It is well known that the performance of such sparse-matrix vector multiplications is limited by the available memory bandwidth. On XT5, the STREAM Triad operation achieves about 2.27 GB/s per core on XT5 when all eight cores are used (consistent with the approach used for all of our calculations). Following the methodology in Gropp et al [24], we estimate the memory bandwidth limited performance bound to be ~17% of the theoretical machine peak for sparse matrix-vector multiplication (about 6 bytes per flop). In Table 3, we show the percentage of peak machine flop performance that SN2ND is currently achieving in the second-to-last column, and the percentage of memory bandwidth bound in the last column. These data show that UNIC achieves about 5-7% of theoretical peak and 34-43% of the memory bandwidth bound on this machine. Therefore, per-processor performance, though low, appears to be scaling well over a sixteen fold increase in the processor core count. Given the known issues were identified with the scattering source update routines recently and the relatively small amount of work that we were able to invest in the preconditioner development until now, low floating-point performance was not unexpected. Over the coming months, we will work on optimizing the floating point performance of the code further.

VII. Conclusions and Future Work

The preceding calculations give us much hope that we will actually be able to meet the time dependent neutronics requirements of the multi-year development and analysis project we are working on for sodium cooled, fast reactors. With relatively little manpower invested into the SN2ND solver, we were able to combine several “off the shelf” computing packages and rapidly produce a neutronics solver that can

reliably and justifiably utilize large scale parallel machines. This new tool already provides accurate and reliable solutions to several difficult numerical benchmark problems for neutron transport, and we anticipate obtaining accurate solutions to the problems discussed in this manuscript. We hope to have made clear the potential of SN2ND to effectively use available large scale machines in addition to future, larger sized machines. We have demonstrated weak scaling of over 75% on 163,840 cores of BlueGene/P and over 130,000 cores of XT5, and completed a space-angle convergence study of two reactor problems. In both problems we presented flux and/or power solutions that are not achievable using legacy deterministic tools. We discussed how the SN2ND algorithm is scalable and, despite its nascent development status, has been largely successful at producing new science results using BlueGene/P and XT5.

In the near future, we will be carrying out extensive performance optimizations at the algorithmic and implementation levels while adapting to the hostile memory hierarchy limitations. First, we need to improve our per-processor performance on the spatial discretization. This will improve our overall performance in the above weak scaling results by reducing communication overhead on the angular communicator. We are currently initiating work on two fronts to accomplish this: spatial p-refinement and the utilization of tensor-matrix vector multiplication techniques that have previously demonstrated excellent flop performance [25]. With multi-grid p-refinement, we expect to vastly increase the amount of local work we perform without being concerned with the low availability of memory. Combined with better-optimized routines, we expect to replace the memory-bandwidth limited sparse-matrix operations we are currently relying upon with more efficient tensor-matrix vector operations [25] that are compatible with our method.

Second, we need to improve our scaling performance in angle. We have already identified a poorly performing sparse matrix-vector algorithm in the scattering source update process. We can again utilize tensor-matrix vector operations in addition to parallelizing the work on the angle communicator (we currently duplicate this work on each process). In addition to these changes, we need to implement a fixed iteration algorithm by angle to impose proper load balancing and use more than one angle per process. This will reduce the communication costs on the angle communicator that become important at high angle parallelization. A final change we need to make is to partition the synthetic equation over more processors of the angular communicator (currently assigned to just the first process). This should allow us to further reduce the time to solution and reduce the current load imbalance we see in angle due to the synthetic

equation.

Finally, we must implement a Krylov methodology using the existing algorithm we have built as a preconditioner. This is essential because with time dependent problems the inclusion of the fission source will greatly degrade the performance of the Gauss-Seidel algorithm. This will also allow us to do parallelization by group which will allow us to effectively utilize larger future machines with millions of processors. Our long term goal is to be able to perform calculations with 200+ groups, 400+ angles, and 100+ million vertices efficiently so that we can start doing time-dependent calculations with thousands of time steps in a coupled multi-physics simulation framework that resolves the tight coupling among physics modules with the same fidelity as is done in each physics.

References

1. E. E. Lewis and Jr. W. F. Miller. Computational Methods of Neutron Transport, John Wiley & Sons, New York, 1984.
2. G. Palmiotti, M. A. Smith, C. Rabiti, M. Leclere, D. Kaushik, A. Siegel, B. Smith, E.E. Lewis, "UNIC: Ultimate Neutronic Investigation Code," Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications, Monterey, California, April 15-19, 2007.
3. Argonne National Laboratory, Argonne Leadership Computing Facility, <http://www.alcf.anl.gov>.
4. Oak Ridge National Laboratory, National Center for Computational Sciences, <http://www.nccs.gov>.
5. C. Rabiti, E. Wolters, M. A. Smith, G. Palmiotti, "Spherical Quadratures for the Discrete Ordinates Method," Trans. Am. Nucl. Soc 96, Boston, MA, June 24-28, 2007.
6. J. Lamarsh, Introduction to Nuclear Engineering, Addison-Wesley Publishing Co, Massachusetts, 1983.
7. R. E. Alcouffe, R. S. Baker, J. A. Dahl, S.A. Turner, and Robert Ward: PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System, LA-UR-05-3925, May 2005.
8. W. A. Rhoades, R. L. Childs, M. B. Emmett, and S. N. Cramer, "Application of the Three-Dimensional Oak Ridge Transport Code," *Proc. Am. Nucl. Soc. Topical Meeting on Reactor Physics and Shielding*, pp. 225-238, Chicago, September 17-19, 1984.
9. Sjoden, G. and Haghghat, A., "PENTRANTM: Parallel Environment Neutral particle TRANsport in 3-D Cartesian Geometry", Proc. Int. Conf. on Mathematical Methods and Supercomputing for Nuclear Applications, pp. 232-234. Saratoga Springs, NY (1997). Performance of PENTRANTM 3-D Parallel Particle Transport Code.
10. G. Palmiotti, E. E. Lewis and C. B. Carrico, VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation, ANL-95/40 Argonne National Laboratory, 1995.
11. C. R. E. de Oliveira, A. J. H. Goddard, "EVENT - A Multidimensional Finite Element-Spherical Harmonics Radiation Transport Code", Proceedings of the OECD International Seminar on 3D Deterministic Radiation Transport Codes, Paris, December 01-02, 1996.
12. John M. McGhee, Randy M. Roberts, Jim E. Morel, "The DANTE Boltzmann Transport Solver: An Unstructured Mesh, 3-D, Spherical Harmonics Algorithm Compatible with Parallel Computer Architectures," Int. Int. Conf. on Math. Meth. and super. For Nuc. Apps., Saratoga Springs, New York, Oct. 5-10, 1997.
13. T. A. Wareing, J. M. McGhee, J. E. Morel, and S. D. Pautz, "Discontinuous Finite Element Sn Methods on 3-D Unstructured Meshes," Nuclear Science and Engineering, 138:1-13, 2001.
14. A. Siegel, D. Kaushik, P. Fischer, G. Palmiotti, M. A. Smith, C. Rabiti, J. Ragusa, "Software Design of SHARP," Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications, Monterey, California, April 15-19, 2007.
15. C. Rabiti, M. A. Smith, G. Palmiotti, "A Three-dimensional Method of Characteristics on Unstructured

- Tetrahedral Meshes,” *Trans. Am. Nucl. Soc.* 96, 470, 2007.
16. M. A. Smith, G. Palmiotti, C. Rabiti, D. Kaushik, A. Siegel, B. Smith, E.E. Lewis, “PNFE Component of the UNIC Code,” Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications, Monterey, California, April 15-19, 2007.
 17. S. Balay, K. R. Buschelman, W. D. Gropp, D. K. Kaushik, M. G. Knepley, L. C. McInnes, and B. F. Smith, “PETSc home page,” www.mcs.anl.gov/petsc.
 18. M. A. Smith, G. Palmiotti, et al., “Benchmark on Deterministic Transport Calculations Without Spatial Homogenization (MOX Fuel Assembly 3-D Extension Case),” OECD/NEA document, NEA/NSC/DOC(2005)16, October 2005.
 19. Personal Communication, Frederic Varaine, CEA, IAEA CRP on Phenix end of life tests; control rod withdrawal, 2009.
 20. Lawrence Livermore National Laboratory, Visit: A distributed, parallel, visualization tool, <https://wci.llnl.gov/codes/visit/home.html>.
 21. Sandia National Laboratory, CUBIT: Geometry and Mesh Generation Toolkit, <http://cubit.sandia.gov/>.
 22. Amine Abou-Rjeili and George Karypis, “Multilevel Algorithms for Partitioning Power-Law Graphs,” IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2006.
 23. NEA Nuclear Science Committee, “International Handbook of Evaluated Criticality Safety Benchmark Experiments,” NEA/NSC/DOC(95)03, September 2007.
 24. W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith. Toward realistic performance bounds for implicit CFD codes. In D. Keyes, A. Ecer, J. Periaux, N. Satofuka, and P. Fox, editors, *Proceedings of Parallel CFD’99*, pages 233–240. Elsevier, 1999.
 25. D. Kaushik, W. Gropp, M. Minkoff, B. Smith, “Improving the Performance of Tensor Matrix Vector Multiplication in Cumulative Reaction Probability Based Quantum Chemistry Codes,” *Proceedings of the 15th International Conference on High Performance Computing (HiPC 2008)*, Bangalore, India, Springer LNCS5374:120-130, December, 2008.

Acknowledgements

This work has been supported by DOE Office of Nuclear Energy, Science and Technology through the Advanced Fuel Cycle Initiative – B&R AF58 and in part by U.S. DOE under Contract DE-AC02-06CH11357. For BlueGene/P time, this research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357. Also, this research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

The submitted manuscript has been created in part by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Dept. of Energy 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, by or on behalf of the Government.