

A COMPARISON OF STOCHASTIC MESH CELL VOLUME COMPUTATION STRATEGIES FOR THE RANDOM RAY METHOD OF NEUTRAL PARTICLE TRANSPORT

John R. Tramm, Andrew R. Siegel, Amanda L. Lund, Paul K. Romano

Argonne National Laboratory
9700 S. Cass Avenue Lemont, IL, USA

{jtramm,siegela,alund,promano}@anl.gov

ABSTRACT

The random ray method is a recently developed neutron transport method that can be used to perform efficient full-core, general-purpose, high-fidelity 3D simulations of nuclear reactors. While Tramm et al. have so far documented the new random ray algorithm in several publications, one critical detail has not yet been published: how to best determine the volume of each source region (or cell) of the simulation. As the “true” analytical constructive solid geometry cell volumes are typically not known a priori they must be computed by the application at runtime, which is not straightforward in TRRM as different rays are used each power iteration such that the sampled volume of each cell also changes between iterations. In the present study, we analyze two different on-the-fly stochastic methods for computing the cell volumes and quantify their impacts on the accuracy of scalar flux estimates. We find that the “naïve” stochastic volume estimator (which arises naturally from the derivation of the Method of Characteristics), is highly biased and can result in over 1,000 pcm error in eigenvalue. Conversely, we find that the “simulation averaged” estimator is unbiased and is therefore equivalent to the use of analytical cell volumes even when using a coarse ray density. Thus, the new simulation averaged method is a critical (and as yet undocumented) component of the TRRM algorithm, and is therefore vital information for those in the reactor physics community working to implement random ray solvers of their own.

1. Introduction

The Random Ray Method (TRRM) of neutral particle transport is a technique for numerically estimating the solution to the Boltzmann neutron transport equation. It shares many similarities with the Method of Characteristics (MOC), though the key difference is that TRRM uses a continuous rather than discrete treatment for the spatial and angular dependencies of the angular flux. Instead of using a fixed deterministic quadrature as in traditional MOC, TRRM samples rays from a uniform random distribution in space and angle and tracks them through the simulation domain, solving the MOC equations along the way. While random ray does introduce the complexities of stochastic uncertainties, there are several key advantages of TRRM as compared to traditional deterministic MOC:

- Extreme reduction in memory usage requirements. This advantage stems from the fact that in TRRM, no storage of boundary angular fluxes is required.
- TRRM can reach an unbiased stationary distribution using only a small fraction of the track/ray density that would be required for a deterministic solver. The result is that a TRRM power iteration takes orders of magnitude less work than a deterministic one while achieving similar fidelity. This allows TRRM to converge solutions using naïve power iteration in time that is highly competitive with traditional deterministic MOC [1] that uses coarse mesh finite difference (CMFD) [2] acceleration.

TRRM has been implemented in the Advanced Random Ray Code (ARRC) [3], and has been demonstrated to be capable of converging a full core high-fidelity 3D benchmark problem [1]. More information on the theory and derivation of TRRM can be found in [4], [3], and [5].

In the present study, we will be looking deeper into one of the statistical issues that arises in the derivation of TRRM, which stems from TRRM's stochastic rather than deterministic basis. In particular, we will derive the scalar flux expression that involves estimation of the volume of each mesh cell, and show that direct use of this expression results in undersampling bias when randomness is introduced. We will then propose a new method to fix the bias and quantify its impact on simulation accuracy.

2. Derivation

While the MOC and random ray equations have been derived in many other works (e.g., [6], [7], [8]), we provide a cursory derivation here as their form is crucial for understanding how cell volumes are used in MOC and TRRM, and how errors might arise from their various estimation techniques. We begin with the steady-state Boltzmann Neutron Transport partial differential equation (PDE) (Equation 1), where all symbols have their usual meaning (for instance, as defined in [5]).

$$\vec{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, \vec{\Omega}, E) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, \vec{\Omega}, E) = \int_0^\infty dE' \int_{4\pi} d\Omega' \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E) \psi(\vec{r}, \vec{\Omega}', E') + \frac{\chi(\vec{r}, E)}{4\pi k_{\text{eff}}} \int_0^\infty dE' \nu \Sigma_f(\vec{r}, E') \int_{4\pi} d\Omega' \psi(\vec{r}, \vec{\Omega}', E') \quad (1)$$

If the right hand side of Equation 1 is condensed into a single term, represented by the total neutron source term $Q(\vec{r}, \vec{\Omega}, E)$, the form given in Equation 2 is reached.

$$\overbrace{\vec{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, \vec{\Omega}, E)}^{\text{streaming term}} + \overbrace{\Sigma_t(\vec{r}, E) \psi(\vec{r}, \vec{\Omega}, E)}^{\text{absorption term}} = \overbrace{Q(\vec{r}, \vec{\Omega}, E)}^{\text{total neutron source term}} \quad (2)$$

Fundamentally, MOC works by decomposing the Boltzmann PDE into a family of ordinary differential equations (ODEs), each of which solves for the angular flux only along a single

characteristic line, as in Equation 2, thus altering the full spatial and angular scope of the transport equation into something that holds true only for a particular linear path (also called a “track” or “ray”) through the reactor. These tracks are linear for neutrons as they are neutral particles and are therefore not subject to field effects. If enough of these characteristic ODEs are solved for, we can approximate the solution to the governing Boltzmann PDE. To accomplish this, we parameterize \vec{r} (the neutron’s spatial location) with respect to some reference location \vec{r}_0 such that $\vec{r} = \vec{r}_0 + s\vec{\Omega}$. In this manner, Equation 2 can be rewritten for a specific segment length s at a specific angle $\vec{\Omega}$ through a constant cross section region of the reactor geometry as in Equation 3.

$$\vec{\Omega} \cdot \vec{\nabla}\psi(\vec{r}_0 + s\vec{\Omega}, \vec{\Omega}, E) + \Sigma_t(\vec{r}_0 + s\vec{\Omega}, E)\psi(\vec{r}_0 + s\vec{\Omega}, \vec{\Omega}, E) = Q(\vec{r}_0 + s\vec{\Omega}, \vec{\Omega}, E) \quad (3)$$

As this equation holds along a 1D path, we can assume the dependence of s on \vec{r}_0 and $\vec{\Omega}$ such that $\vec{r}_0 + s\vec{\Omega}$ simplifies to s . When the differential operator is also applied to the angular flux ψ , we arrive at the characteristic ODE given in Equation 4.

$$\frac{d}{ds}\psi(s, \vec{\Omega}, E) + \Sigma_t(s, E)\psi(s, \vec{\Omega}, E) = Q(s, \vec{\Omega}, E) \quad (4)$$

An analytical solution to this characteristic ODE can be achieved with the use of the integrating factor $e^{\int_0^s ds' \Sigma_t(s', E)}$ to arrive at the final form of the characteristic equation shown in Equation 5.

$$\psi(s, \vec{\Omega}, E) = \psi(\vec{r}_0, \vec{\Omega}, E)e^{-\int_0^s ds' \Sigma_t(s', E)} + \int_0^s ds'' Q(s'', \vec{\Omega}, E)e^{-\int_{s''}^s ds' \Sigma_t(s', E)} \quad (5)$$

With this characteristic equation, we now have an analytical solution along a linear path through any constant cross section region of a reactor. Similar to many other solution approaches to the Boltzmann neutron transport equation, the MOC and random ray approaches also use a “multi-group” approximation in order to discretize the continuous energy spectrum of neutrons travelling through the reactor into fixed set of energy groups G , where each group $g \in G$ has its own specific cross section parameters.

Following the multi-group assumption, another assumption made is that a large and complex problem can be broken up into small constant cross section regions i [9], and that these regions have group dependent isotropic sources (fission + scattering), $Q_{i,g}$. Anisotropic sources are also possible with MOC based methods[10], but are not used here for simplicity. With these key assumptions, the multi-group characteristic equation can be written as in Equation 6.

$$\psi_g(s, \vec{\Omega}) = \psi_g(\vec{r}_0, \vec{\Omega})e^{-\int_0^s ds' \Sigma_{t,i,g}(s')} + \int_0^s ds'' Q_{i,g}(s'', \vec{\Omega})e^{-\int_{s''}^s ds' \Sigma_{t,i,g}(s')} \quad (6)$$

In ARRC, a constructive solid geometry (CSG) definition of the reactor is used to create spatially defined source regions. These neutron source regions are often approximated as being constant (flat) in intensity of source, but can also be defined using a higher order source (linear, quadratic, etc.) that allows for fewer source regions to be required to achieve a specified solution

fidelity[11,10]. In this study, we will use the normal approximation of a spatially constant isotropic fission and scattering source $Q_{i,g}$ in each flat source region (FSR) i and for energy group g , which allows us to solve Equation 6 and to arrive at Equation 7.

$$\psi_g(s) = \psi_g(0)e^{-\Sigma_{t,i,g}s} + \frac{Q_{i,g}}{\Sigma_{t,i,g}} (1 - e^{-\Sigma_{t,i,g}s}) \quad (7)$$

As we will be integrating each track (or ray) through one FSR mesh cell at a time, we can rewrite our equation into a form where we will be solving for the outgoing angular flux ψ_g^{out} given the incoming angular flux ψ_g^{in} as in Equation 8, where ℓ_r now represents the length of the ray's path through FSR mesh cell i .

$$\psi_g^{out} = \psi_g^{in} e^{-\Sigma_{t,i,g}\ell_r} + \frac{Q_{i,g}}{\Sigma_{t,i,g}} (1 - e^{-\Sigma_{t,i,g}\ell_r}) \quad (8)$$

Using Equation 8, we can now compute the angular flux for a track as it traverses through a reactor geometry and passes through different FSRs. However, to compute the scattering and fission sources for the next iteration, we need to be able to compute the average scalar flux for each cell, and to do that we need to be able to determine the total average angular flux in each cell. To track these quantities, we will begin by finding a solution for the average angular flux $\bar{\psi}_{r,i,g}$ for a single ray r passing through region i and for energy group g , as in Equation 9.

$$\bar{\psi}_{r,i,g} = \frac{1}{\ell_r} \int_0^{\ell_r} \psi_g(s) ds \quad (9)$$

By substituting Equation 7 into Equation 9 and solving, we can determine the average angular flux for a single ray through the region i using Equation 10.

$$\bar{\psi}_{r,i,g} = \frac{Q_{i,g}}{\Sigma_{t,i,g}} - \frac{\psi_{r,g}^{out} - \psi_{r,g}^{in}}{\ell_r \Sigma_{t,i,g}} \quad (10)$$

By re-arranging Equation 8, we can define $\Delta\psi_{r,g}$ as the change in angular flux for ray r passing through region i :

$$\psi_{r,g}^{in} - \psi_{r,g}^{out} = \Delta\psi_{r,g} = \left(\psi_{r,g}^{in} - \frac{Q_{i,g}}{\Sigma_{t,i,g}} \right) (1 - e^{-\Sigma_{t,i,g}\ell_r}). \quad (11)$$

If we substitute Equation 11 into Equation 10, we arrive at:

$$\bar{\psi}_{r,i,g} = \frac{Q_{i,g}}{\Sigma_{t,i,g}} + \frac{\Delta\psi_{r,g}}{\ell_r \Sigma_{t,i,g}}. \quad (12)$$

Because scalar flux is defined as:

$$\phi = \int_{4\pi} \psi(\Omega) d\Omega \quad (13)$$

we can estimate the scalar flux $\phi_{i,g}$ in each region i and energy group g by taking a weighted average (by distance travelled) over all N_i rays passing through that region in a power iteration, where r is the contribution from a single ray, as in Equation 14.

$$\phi_{i,g} \equiv \bar{\psi}_{i,g} = \frac{\sum_{r=1}^{N_i} \ell_r \bar{\psi}_{r,i,g}}{\sum_{r=1}^{N_i} \ell_r} \quad (14)$$

By substituting Equation 12 into Equation 14 and simplifying, we arrive at a (naïve) expression for each iteration’s scalar flux estimate:

$$\phi_{i,g}^{naive} = \frac{Q_{i,g}}{\Sigma_{t,i,g}} + \frac{\sum_{r=1}^{N_i} \Delta\psi_{r,g}}{\Sigma_{t,i,g} \sum_{r=1}^{N_i} \ell_r}. \quad (15)$$

The estimate of $\phi_{i,g}^{naive}$ defined in Equation 15 is computed once per power iteration by selecting rays randomly from a uniform distribution in space and angle and following them for a set distance through the problem geometry, with rays solving Equation 11 for all energy groups as they pass through each FSR and tallying $\Delta\psi_{r,g}$ and ℓ_r contributions along their way. The scalar flux estimates produced in each iteration’s stochastic transport “sweep” are then used in several ways in the power iteration process. Namely, to compute the neutron source $Q_{i,g}$ used in the next power iteration and to update the eigenvalue estimate. Full definitions and derivations of the entire power iteration, source update, and eigenvalue computation processes are given in [5], and are not included here for the sake of brevity.

3. Scalar Flux Ratio Estimators

While the naïve estimator defined in Equation 15 is numerically sound for a deterministic process (e.g., classical MOC), issues arise when used in a stochastic process like TRRM. Specifically, the second term of the equation (which is the ratio of the change in angular flux for all rays through the cell to the total distance travelled by all rays in the cell) features stochastic variables in both the numerator and denominator, making it a biased estimator [12].

We have several options for how to treat the naïve biased estimator defined in Equation 15, which will be presented in the following sections of this paper. To test the different methods, we will implement them in ARRC [3] and use them to simulate the standard 2D C5G7 light water reactor benchmark problem [13] that features four assemblies and a large moderator reflector region. While ARRC is capable of much larger 3D full core simulations, the relatively small 2D C5G7

problem is used in this analysis as it is fast to run, has defined 7 group macroscopic cross sections, and features a reference solution generated via multigroup Monte Carlo. For all tests, we used a mesh featuring 142,964 FSRs, ray lengths of 250 cm, and a ray dead zone [4] of 25 cm. For all tests, 800 inactive iterations were used to converge the source to a stationary distribution before accumulating statistics in the active region of the simulation. All runs were converged to a target uncertainty threshold of 0.003 source averaged relative error (SARE), which means that the total fission source (integrated across all 7 energy groups in the problem) in each FSR mesh cell will be converged to 0.3% (on average) for all FSR mesh cells.

3.1. Naïve Estimator

The first option is to simply use the estimator defined in Equation 15 verbatim and implement it directly in a random ray power iteration simulation. The results of testing the naïve estimator on the 2D C5G7 sample problem are shown in Table 1. Our findings show that the naïve estimator does indeed produce a biased eigenvalue solution when low ray densities are used, but the eigenvalue approaches the true reference solution as the ray density is increased. Increases in the ray density are directly proportional to the number of ray crossings in any given FSR (N). Thus, our naïve estimator can be said to be *asymptotically consistent*. As shown in Table 1, if we increase the ray density (and thus N) the bias eventually diminishes to acceptably low levels. This phenomenon is very similar to the undersampling biases found in Monte Carlo simulations when very few particles are used [14,15]. While it is good that the estimator is at least consistent, the downside is that we have to run a large number of rays per power iteration to achieve acceptable accuracy. A more ideal estimator with reduced bias might be able to get by with a much coarser ray density and therefore a much faster time to solution.

Table 1: Eigenvalue and standard deviations of different estimators. Bold indicates entries with error from the reference solution greater than 3 standard deviations.
Reference solution: 1.18655

Rays per Iteration	Analytical	Simulation Averaged	Naïve
1,250	1.18662 ± 0.00020	1.18698 ± 0.00019	1.17248 ± 0.00121
2,500	1.18646 ± 0.00020	1.18664 ± 0.00020	1.17911 ± 0.00114
5,000	1.18658 ± 0.00020	1.18692 ± 0.00019	1.18192 ± 0.00104
10,000	1.18667 ± 0.00017	1.18663 ± 0.00021	1.18473 ± 0.00094
20,000	1.18685 ± 0.00018	1.18651 ± 0.00021	1.18713 ± 0.00121
40,000	1.18647 ± 0.00021	1.18647 ± 0.00017	1.18669 ± 0.00097
80,000	1.18689 ± 0.00018	1.18663 ± 0.00022	1.18659 ± 0.00035
160,000	1.18676 ± 0.00013	1.18659 ± 0.00016	1.18614 ± 0.00048

3.2. Simulation Averaged Estimator

One method for reducing the bias in a ratio estimator is to increase the number of samples for the stochastic variable in the denominator. This approach has recently been deployed to reduce the

bias for ratio estimators that arise in the stochastic gradient descent algorithm from the field of machine learning, as discussed in [16].

Luckily, in the context of a random ray simulation, improvement in the number of samples for the denominator variable is easily accomplished. Instead of computing a new stochastic tally for ℓ (the distance estimator in the denominator) each power iteration, we can convert this to a “simulation average” quantity. As the average distance travelled by rays through a cell is a function only of geometry (and not a function of neutronics or the state of the neutron distribution in the system), we can begin tallying this quantity in the first power iteration and keep adding to the tally throughout the simulation. As more power iterations are run, the sample size of the distance tally will continue to grow, therefore continually reducing the overall bias of the scalar flux estimator. It is important to note that we could not use this simulation averaged strategy with the numerator tally for $\Delta\psi_{r,g}$, as that value is dependent on the source and neutron distribution of the system, which evolves between power iterations as it moves towards a stationary distribution over the course of the inactive phase of the simulation.

Thus, if we define M_i as the number of ray crossings in a given FSR i over the course of the entire simulation, we can write a new “simulation averaged” estimator as:

$$\phi_{i,g}^{simulation} = \frac{Q_{i,g}}{\Sigma_{t,i,g}} + \frac{\left[\sum_{r=1}^{N_i} \Delta\psi_{r,g} \right] / N_i}{\left[\Sigma_{t,i,g} \sum_{r=1}^{M_i} \ell_r \right] / M_i} \quad (16)$$

where each power iteration M_i grows by approximately N_i (which varies stochastically between iterations), and after many power iterations $M_i \gg N_i$.

We tested our new simulation averaged estimator on the 2D C5G7 problem and compared it against the original naïve estimator. The results, given in Table 1, show that the new simulation averaged estimator virtually eliminates the bias even for very low ray densities. The new estimator therefore allows us to run with the coarsest possible ray density, which is typically set as the lowest number of rays required to hit (almost) all FSRs at least once each power iteration*. Use of a coarse ray density is highly advantageous as it minimizes time spent in the inactive phase of the simulation.

*In the case of the 2D C5G7 benchmark problem, this minimal density occurs when approximately 1,250 rays per iteration are used, resulting in a mean N for each FSR of about 14. This value is arrived at empirically. If fewer rays are used, the iteration costs of building the source can become more costly than the transport sweep itself, resulting in an overall increase in runtime. Additionally, if very few rays are used per iteration (such that most FSRs are not being hit each iteration at all), the simulation can become unstable.

3.3. Analytical Volume Estimator

In the event that the true analytical volumes are known for each FSR cell V_i a priori, then a constant expression can be derived for the denominator of Equation 15. If the distance travelled by each ray through the global simulation domain is D , and R total rays are run in each power iteration, then the exact expected distance of rays travelling through a given FSR cell i becomes a constant value that can be computed analytically as:

$$\sum_{r=1}^N \ell_r = V_i R D. \quad (17)$$

In this manner, our estimator for the scalar flux is no longer a ratio estimator, as the denominator becomes a constant value:

$$\phi_{i,g}^{analytical} = \frac{Q_{i,g}}{\Sigma_{t,i,g}} + \frac{\sum_{r=1}^N \Delta\psi_{r,g}}{\sum_{t,i,g} V_i R D}. \quad (18)$$

Due to the complexities of user-defined constructive solid geometry (CSG) definitions of reactor geometry objects, we typically do not know the analytical volumes of each FSR cell analytically, which makes this estimator impractical for realistic use cases. However, for the purposes of this analysis, we were able to manually calculate the volumes of each FSR in our 2D C5G7 mesh in ARRC. Using these true analytical volumes, we performed a ray density convergence analysis (shown in Table 1) and found that the behavior of the analytical mode was nearly identical to that of the simulation averaged estimator.

3.4. Changes in Convergence Rate

In the previous section, we showed that the new simulation averaged estimator was able to virtually eliminate the bias from the simulation even with a very coarse ray density. However, we noticed that use of this estimator also increased the number of active iterations required to converge the solution below our target uncertainty threshold. These results, given in Table 2, show that approximately 5x more active iterations are required to converge statistics in the active phase of the simulation with equal ray densities, meaning that the unbiased simulation averaged estimator comes with a cost of significantly higher variance. This is likely due to the fact that the numerator and denominator in Equation 15 are not fully independent. As the change in angular flux (defined in Equation 11) is a function of the length travelled by the ray through the FSR, these two values will be naturally correlated. Figure 1 demonstrates this correlation by showing the numerator and denominator values tallied for Equation 15 for all iterations in the active region of a 2D C5G7 simulation, for a single (randomly chosen) FSR and energy group. Thus, by using the correct analytical value or simulation averaged estimator for the denominator, the correlation is ignored and the variance of the ratio as a whole is increased even though the bias is eliminated.

Given the differences in variance, it is therefore important to compare the overall runtimes required for the different methods to achieve a target convergence level and accuracy. To test this,

Table 2: Active iteration requirements to reach target source uncertainty level.

Rays per Iteration	Analytical	Simulation Averaged	Naïve
1,250	1,679	1,675	315
2,500	829	826	165
5,000	410	410	85
10,000	203	202	42
20,000	100	99	20
40,000	48	48	8
80,000	22	22	3
160,000	9	9	4

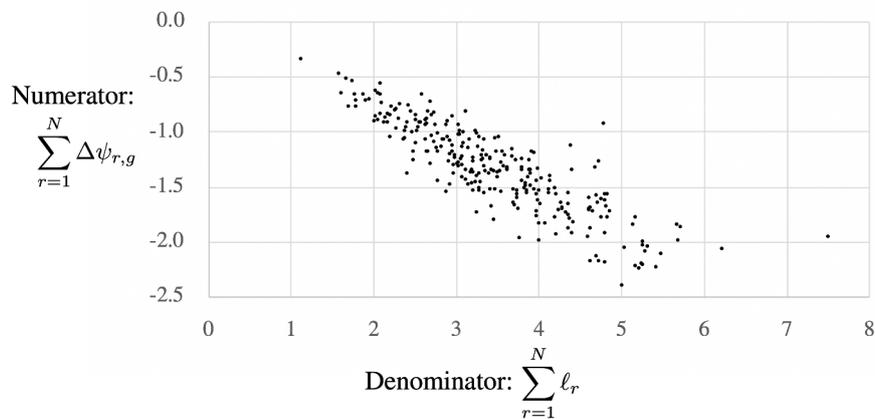


Figure 1: Relationship between the numerator and denominator of Equation 15 of a single FSR and energy group for each active iteration in a test simulation.

we measured the runtimes of the simulation when using each estimator in a configuration that was able to result in less than 25 pcm overall bias in eigenvalue. For the naïve estimator, this meant using 20,000 rays per iteration, while for the simulation averaged and analytical methods we only needed to use the minimum ray density of 1,250 rays per iteration. For all estimators, we used 800 inactive cycles to reach a stationary source distribution (as defined by convergence of Shannon Entropy [3]). Once stationary, the naïve estimator required 20 active iterations to converge statistics, resulting in an overall runtime of 128 seconds. The simulation averaged estimator required 1,675 active iterations to converge statistics, resulting in an overall runtime of 33 seconds. The analytical method behaved similarly in both numerical and runtime performance as the simulation averaged estimator. Thus, for comparable accuracy, the simulation averaged and analytical methods were about 4x faster in runtime even though they had higher variances, due to the fact that they could run with 16x fewer rays per iteration and therefore could get through the inactive iterations much faster. The error in pin power distributions was also improved. At 20,000 rays per iteration the naïve estimator achieved an absolute average percent pin power error (APE)

value of 0.454%, while at 1,250 rays per iteration the simulation averaged estimator and analytical methods achieved APE values of 0.367% and 0.394% respectively.

It is important to consider that the 2D C5G7 test case that was used for this analysis has a fairly low dominance ratio which only required 800 power iterations to reach source stationarity. Full core 3D reactor simulation problems can have much higher dominance ratios that can require many thousands of iterations to reach stationarity – meaning that the increased ray density requirements of the naïve estimator would greatly increase the numerical performance advantage of the simulation averaged estimator. For example, in [1], 11,763 inactive iterations and 1,570 active iterations (using the simulation averaged estimator) were required to converge a full core 3D light water reactor problem with a high-fidelity mesh. If the naïve estimator were used to cut the active cycle count by 85x, it would only reduce the total iteration count by approximately 12% while increasing the required ray density by 16x. Conversely, for very small problems with low dominance ratios (e.g., a 2D pincell), the naïve estimator may provide more competitive performance, though with the downside that a convergence study would have to be performed to ensure that the solution is not biased.

It is also important to consider the effect of the user’s target uncertainty on estimator performance. For instance, if a very low level of uncertainty is desired (e.g., less than 1 pcm in eigenvalue), then the number of active iterations can become much larger than the number of inactive iterations, even for high dominance ratio problems. In such cases, the lower variance of the naïve estimator may have a much larger benefit on overall numerical performance. However, it is usually not worth converging MOC and random ray simulations to such tight levels due to other sources of error inherent in the simulation that would not be eliminated by doing so (e.g., nuclear data uncertainties, multigroup cross section generation, transport correction errors), which can add up to 100 pcm in eigenvalue [17]. Thus, the transport convergence targets used in this analysis (in the range of 20 pcm in eigenvalue) represent the tighter bounds for real world engineering usage.

4. Conclusions

In this study, we derived the characteristic equations used in TRRM and used them to define an estimator for the scalar flux in each source region cell of a simulation. We found that our estimator was in fact a ratio estimator having stochastic variables in both the numerator (where the change in angular flux $\Delta\psi$ is tallied) and denominator (where the total distance travelled by all rays through the cell ℓ is tallied). We tested this naïve estimator out on the 2D C5G7 benchmark problem and found that the estimator exhibited clear undersampling bias, in one case resulting in over 1,200 pcm eigenvalue error. However, we found the estimator to be consistent, meaning that the undersampling bias could be mitigated if very high number of rays per iteration were used in a simulation.

We then investigated a method that reduces bias of ratio estimators by increasing the number of samples taken for the denominator stochastic variable, which has shown to be successful for other estimators commonly used in the field of machine learning. This strategy allowed us to develop a new “simulation averaged” scalar flux estimator, where distances travelled through each FSR are tallied continuously throughout the simulation rather than being discarded each power iteration. Empirical testing of the new estimator revealed no measurable bias, meaning that it could be used even when very few rays were run per power iteration. Additionally, we found that the simulation

averaged volume estimator produced results nearly identical to a method that used the true analytical volumes. However, the new simulation averaged estimator did result in an increase in variance compared to the naïve estimator. Overall, the new unbiased simulation averaged method resulted in 2D C5G7 runtimes that were overall about 4x faster than the naïve estimator for a given accuracy due to the naïve method requiring orders of magnitude more rays per iteration to eliminate bias.

All work to date published by the authors using TRRM has used the simulation averaged estimator, though this has not been documented anywhere previously as it was only discovered in this analysis that using the naïve estimator would result in undersampling bias. These findings are an important detail of random ray neutron transport; those implementing TRRM solvers of their own are highly encouraged to use the simulation averaged estimator so as to mitigate undersampling biases.

ACKNOWLEDGEMENTS

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. The material was based upon work supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357.

We gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory

REFERENCES

- [1] J. Tramm, B. Forget, and K. Smith. “Early experience in full core reactor simulation with The Random Ray Method.” In *PHYSOR 2018: Reactor Physics Paving The Way Towards More Efficient Systems*, pp. 486–497. Cancun (2018).
- [2] K. Smith. “Nodal Method Storage Reduction by Non-linear Iteration.” *Transactions of the American Nuclear Society (1983)*, **volume 44** (1983). URL https://inis.iaea.org/search/search.aspx?orig_q=RN:15010017.
- [3] J. R. Tramm, K. S. Smith, B. Forget, and A. R. Siegel. “ARRC: A random ray neutron transport code for nuclear reactor simulation.” *Annals of Nuclear Energy*, **volume 112**(Supplement C), pp. 693–714 (2018). URL <http://www.sciencedirect.com/science/article/pii/S0306454917303444>.
- [4] J. R. Tramm, K. S. Smith, B. Forget, and A. R. Siegel. “The Random Ray Method for neutral particle transport.” *Journal of Computational Physics*, **volume 342**, pp. 229 – 252 (2017). URL <http://www.sciencedirect.com/science/article/pii/S0021999117303170>.
- [5] J. R. Tramm. “Development of The Random Ray Method of neutral particle transport for high-fidelity nuclear reactor simulation.” Ph.D. thesis, Massachusetts Institute of Technology, Department of Nuclear Science and Engineering (2018). URL <http://hdl.handle.net/1721.1/119038>.
- [6] T. Mazumdar and S. B. Degweker. “Solution of neutron transport equation by Method of Characteristics.” *Annals of Nuclear Energy*, **volume 77**, pp. 522–535 (2015). URL <http://dx.doi.org/10.1016/j.anucene.2014.12.029>.

- [7] W. Boyd, S. Shaner, L. Li, B. Forget, and K. Smith. “The OpenMOC method of characteristics neutral particle transport code.” *Annals of Nuclear Energy*, **volume 68**, pp. 43–52 (2014). URL <http://dx.doi.org/10.1016/j.anucene.2013.12.012>.
- [8] D. Sciannandrone, S. Santandrea, and R. Sanchez. “Optimized tracking strategies for step MOC calculations in extruded 3D axial geometries.” *Annals of Nuclear Energy*, **volume 87**, pp. 49–60 (2016).
- [9] W. Boyd, N. Gibson, B. Forget, and K. Smith. “An analysis of condensation errors in multi-group cross section generation for fine-mesh neutron transport calculations.” *Annals of Nuclear Energy*, **volume 112**, pp. 267 – 276 (2018). URL <https://www.sciencedirect.com/science/article/pii/S0306454917303286>.
- [10] R. M. Ferrer and J. Rhodes. “Extension of linear source MOC methodology to anisotropic scattering in CASMO5.” In *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future* (2014).
- [11] R. M. Ferrer, J. Rhodes, and K. Smith. “Linear source approximation in CASMO5.” In *PHYSOR 2012 – Advances in Reactor Physics – Linking Research, Industry, and Education* (2012).
- [12] T. J. Rao. “Mean of ratios or ratio of means or both?” *Journal of Statistical Planning and Inference*, **volume 102**(1), pp. 129–138 (2002).
- [13] M. A. Smith and E. Lewis. “Benchmark on deterministic transport calculations without spatial homogenisation – A 2-D/3-D MOX fuel assembly benchmark.” Technical Report ISBN 92-64-02139-6, Nuclear Energy Agency: Organization for Economic Co-Operation and Development (2003).
- [14] R. J. Brissenden and A. R. Garlick. “Biases in the estimation of Keff and its error by Monte Carlo methods.” *Annals of Nuclear Energy*, **volume 13**(2), pp. 63–83 (1986).
- [15] F. B. BROWN. “‘K-effective of the world’ and other concerns for Monte Carlo eigenvalue calculations.” *Progress in Nuclear Science and Technology*, **volume 2**(0), pp. 738–742 (2011). URL <http://www.aesj.or.jp/publication/pnst002/data/738-742.pdf>.
- [16] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. “Mutual information neural estimation.” In *35th International Conference on Machine Learning (ICML 2018)*, volume 2, pp. 864–873 (2018). URL <https://arxiv.org/abs/1801.04062>.
- [17] K. Smith and B. Forget. “Challenges in the development of high-fidelity LWR core neutronics tools.” In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, pp. 1809–1825 (2013).

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department 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, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>.