LA-UR-07-5462

Approved for public release;
distribution is unlimited.

Title: Implementing the Coarse Mesh Method into MCNP for Dominance Ratio Calculation

Author(s): Brian Nease & Forrest Brown

Intended for: MCNP reference material

Implementing the Coarse Mesh Method into MCNP for Dominance Ratio Calculation

Brian R Nease
Forrest Brown

Abstract

Recently, a robust method of determining the dominance ratio (Coarse Mesh Method) was developed for iterated source Monte Carlo (MC) methods. The objective of this method was to correctly evaluate the dominance ratio using time series techniques in a completely automated fashion, without requiring specific knowledge of transport theory or time series analysis from the user. In this work, CMM was integrated into MCNP and tested against several benchmarked problems. Since MCNP is a production code, there is a strong desire to reduce memory usage and improve algorithm efficiency with any new methods introduced. With these constraints in mind, CMM was introduced using a two-bin scheme. CMM performed very well (due to the symmetric nature of most problems), though a bin scheme matching the problem dimensions would be more advantageous to ensure that the largest eigenmode of the noise propagation matrix does not cancel out for any particular problem. The difficulties with implementing this larger bin scheme will be discussed, followed by the results of CMM against several test problems in MCNP.

1 Introduction

In the past, calculating dominant eigenvalues of criticality problems has generally been left to deterministic power iteration methods. Performing these calculations using Monte Carlo methods usually involved calculating the fission matrix, which is generally regarded as an inefficient method. It requires a very fine bin overlaying the problem in which the fission contributions from every other bin in the system are stored. Improving the accuracy of the calculation using the fission matrix method thus meant increasing the number of bins n, which in turn meant storing n^2 values.

Recently, a robust method of determining the dominance ratio (i.e. the ratio of the first to fundamental eigenmode of the fission source distribution) was developed for iterated source Monte Carlo (MC) methods [1]. This method was independent of bin structure and used time series analysis of a particular projection of the fission source vector to compute DR, which was a significant improvement. However, this method relied on an autoregressive moving average (ARMA(2,1)) fit, which could be complicated and require fine tuning from the user.

An improvement to this method was sought that retained the strengths of using time series analysis, but that did not rely on the complicated ARMA fitting. This became the Coarse Mesh Method (CMM) [2]. This method was much the same as the previous, but improved on the particular projection vector used. It was found that when using a certain projection vector, DR could be computed reliably using a simple autoregressive order one (AR(1)) fitting. Using this method, the calculation of DR in continuous Monte Carlo calculations became viable.

In this work, CMM was integrated into the MCNP Monte Carlo production code and tested against several benchmark problems where DR was known by other means. A number of factors had to be taken into consideration when integrating this into MCNP's framework. First, since DR was computed on-the-fly instead of by a separate post processing program, the fission sources for every bin and every cycle could not be stored. At most, the fission source values could only be saved for a small number of cycles (three or less) to save on memory and processing time. Second, MCNP could not rely on the commercial libraries of functions (such as IMSL) for routines to solve for eigenvalues/vectors of non-symmetric matrices or invert matrices. New routines had to be written to perform these functions.

A review of CMM theory is presented in the first section. The actual implementation of CMM into MCNP is described next, including the advantages and limitations of this particular implementation. This is followed by the results of testing CMM against several benchmark problems. Lastly, a discussion is given for extending this method to a Fine Mesh Method (FMM), which could potentially be used to generate other higher order ratios, beyond DR.

2 Coarse Mesh Method

2.1 Theory

2.1.1 Regression Analysis of Time Series

Since the source distribution $\hat{S}(\vec{r})$ is a stochastic realization, it can be written as

$$\hat{S}^{(m)}(\vec{r}) \equiv NS(\vec{r}) + \sqrt{N}\hat{e}^{(m)}(\vec{r}), \quad m \geq 0, \tag{1}$$

after simulating the m^{th} stationary cycle in an MC iterated source calculation, where N is the number of particles (neutrons) per cycle, $S(\vec{r})$ is the ensemble average of the source fluctuation (i.e. $S(\vec{r}) = E[\hat{S}^{(m)}(\vec{r})]/N$ with $E[\cdot]$ denoting expectation), and $\hat{e}^{(m)}(\vec{r})$ is the fluctuating part of the source distribution ($E[\hat{e}^{(m)}(\vec{r})] = 0$). The hat above the symbols is used to imply a stochastic realization and the N terms are scaling terms used to clarify the asymptotic behavior of the integrals of individual terms as $N \to \infty$.

The fluctuating part of the source distribution is governed by a linear Markov process

$$\hat{e}^{(m)}(\vec{r}) = \mathbf{A}_0\hat{e}^{(m-1)}(\vec{r}) + \hat{\varepsilon}^{(m)}(\vec{r}) + O(N^{-1/2}), \tag{2}$$

where the following relationships hold:

$$
\begin{gathered}
E\left[\hat{\varepsilon}^{(m1)}\hat{e}^{(m2)}=0\right], \quad m1>m2, \\
E\left[\hat{\varepsilon}^{(m)}=0\right], \\
E\left[\hat{\varepsilon}^{(m1)}\hat{\varepsilon}^{(m2)}=0\right], \quad m1>m2.
\end{gathered}
\tag{3}
$$

The operator $\mathbf{A}_0$ introduced in Eq. (2) is defined via the operation on function $f(\vec{r})$ as

$$
\mathbf{A}_0(f) \equiv \frac{1}{k_0}\int\left[F\left(\vec{r}' \to r'\right) - S_0(\vec{r})\right] f(\vec{r}')dV', \tag{4}
$$

with the requirement that

$$
\int S_0(\vec{r})dV = k_0 . \tag{5}
$$

This operator is also known as the noise propagation matrix and is key to relating how the error is propagated from cycle-to-cycle. It is from $\mathbf{A}_0$ that the particular projection vector necessary for CMM will be obtained.

From the definition of the operator in Eq. (4), the following relationships can be easily derived through substitution:

$$
\mathbf{A}_0^i S_0(\vec{r}) = 0, \quad i \geq 1, \tag{6}
$$

$$
\mathbf{A}_0^i S_j(\vec{r}') = \begin{cases} \left(\dfrac{k_j}{k_0}\right)^i \left[S_j(\vec{r}) - S_0(\vec{r})\right] & \text{when } \int S_j(\vec{r})dr \neq 0 \\ \left(\dfrac{k_j}{k_0}\right)^i S_j(\vec{r}) & \text{when } \int S_j(\vec{r})dr = 0 \end{cases}, \quad i \geq 1,\ j \geq 1. \tag{7}
$$

If we consider a discrete form of $\mathbf{A}_0$, Eq. (7) would allow us to assume that the eigenvalues of $\mathbf{A}_0$ approach k_i/k_0, $i = 1,2,\ldots,$ if the number of bins were large and each bin was sufficiently fine spatially. Of course, this implies that DR could be solved for directly from $\mathbf{A}_0$ using matrix solvers in mathematics libraries if it were statistically computed. This would be computationally prohibitive, however, due to the large number of bins needed to sufficiently reduce the discretization error. For this reason, we use the time series approach, guided by the eigenvectors of $\mathbf{A}_0$, so that the DR calculation is free of discretization error. The exact mechanics of the projection process and time series analysis are described next.

2.1.2 Projection Process

Consider a discrete form of Eqs. (2) and (3):

$$\vec{e}^{(m)} = \mathbf{A}_0 \vec{e}^{(m-1)} + \vec{\varepsilon}^{(m)}, \tag{8}$$

$$\begin{gathered} E\left[\vec{\varepsilon}^{(m1)} \otimes \vec{e}^{(m2)} = 0\right], \quad m1 > m2, \\ E\left[\vec{\varepsilon}^{(m)} = 0\right], \\ E\left[\vec{\varepsilon}^{(m1)} \otimes \vec{\varepsilon}^{(m2)} = 0\right], \quad m1 > m2, \end{gathered} \tag{9}$$

where $\vec{e}^{(m)}$ and $\vec{\varepsilon}^{(m)}$ are $p \times 1$ column vectors, $\mathbf{A}_0$ is the corresponding $p \times p$ matrix for a discrete version of Eq. (4), $\otimes$ signifies an outer product, and p is the number of bins. By multiplying Eq. (8) by $\vec{e}^{(m-1)}$ from the right and taking the outer product, we can obtain

$$\vec{e}^{(m+1)} \otimes \vec{e}^{(m)} = \mathbf{A}_0 \left(\vec{e}^{(m)} \otimes \vec{e}^{(m)}\right) + \vec{\varepsilon}^{(m+1)} \otimes \vec{e}^{(m)}. \tag{10}$$

Taking the expectation of Eq. (10) and applying Eq. (9) results in

$$E\left[\vec{e}^{(m+1)} \otimes \vec{e}^{(m)}\right] = \mathbf{A}_0 E\left[\vec{e}^{(m)} \otimes \vec{e}^{(m)}\right]. \tag{11}$$

Defining

$$L_0 = E\left[\vec{e}^{(m)} \otimes \vec{e}^{(m)}\right], \tag{12}$$

$$L_1 = E\left[\vec{e}^{(m+1)} \otimes \vec{e}^{(m)}\right], \tag{13}$$

the matrix $\mathbf{A}_0$ can be expressed and solved for as

$$\mathbf{A}_0 = L_1 L_0^{-1}. \tag{14}$$

The eigenvalue problem of $\mathbf{A}_0$ is

$$\begin{gathered} \mathbf{A}_0 \vec{b}_i = \lambda_i \vec{b}_i, \quad i, \ldots, p \\ \mathbf{A}_0^T \vec{d}_i = \lambda_i \vec{d}_i, \quad i, \ldots, p \end{gathered} \tag{15}$$

Theoretically, the noise propagation matrix $\mathbf{A}_0$ should be positive definite, ensuring that all eigenvalues are real and positive. In reality, statistical fluctuations can often result in $\mathbf{A}_0$ having eigenvectors with complex parts. As the number of particles per cycle and number of cycles increase, however, the statistical fluctuations will decrease in magnitude, leading to smaller complex pieces. Since the complex parts are generally small, they are usually just ignored in this method.

Eigenvectors of the matrix $\mathbf{A}_0^T$ are used in the projection process to compute DR. The transpose of $\mathbf{A}_0$ is used to ensure that all of the eigenvectors are independent. The eigenvector $\vec{d}_1$ corresponding to the dominance ratio is selected as a projection vector and applied to Eq. (8), taking the inner product, which is signified by $\langle \cdot, \cdot \rangle$:

$$\left\langle \vec{d}_1, \vec{e}^{\,(m)} \right\rangle = \left\langle \vec{d}_1, \mathbf{A}_0 \vec{e}^{\,(m-1)} \right\rangle + \left\langle \vec{d}_1, \vec{\varepsilon}^{\,(m)} \right\rangle. \tag{16}$$

Defining a scalar time series as

$$y^{(m)} \equiv \left\langle \vec{d}_1, \vec{e}^{\,(m)} \right\rangle \tag{17}$$

and a scalar noise component as

$$z^{(m)} \equiv \left\langle \vec{d}_1, \vec{\varepsilon}^{\,(m)} \right\rangle, \tag{18}$$

we can rewrite Eq. (16) in the form of an autoregressive (AR) process of order one [AR(1)] with the dominance ratio λ_1 as the autocorrelation coefficient.

$$y^{(m)} = \lambda_1 y^{(m-1)} + z^{(m)}. \tag{19}$$

This is a remarkable result for several reasons. First, the bin dependence for the calculation of DR is removed by the time series process. The calculation of $\mathbf{A}_0$ still requires a certain binning scheme, but (as will be shown) a coarse mesh works well. Second, an AR(1) process is the most simple type of time series. The coefficient is guaranteed to be less than one and is straightforward to calculate.

To solve for the autocorrelation coefficient (DR), the autocovariance function is formed by multiplying Eq. (19) by $y^{(m-1)}$, $m > 1$ and taking the expectation:

$$E\left[y^{(m-1)} y^{(m)} \right] = \lambda_1 E\left[y^{(m-1)} y^{(m-1)} \right] + E\left[y^{(m-1)} z^{(m)} \right], \quad m > 1\,. \tag{20}$$

The time series $y^{(m)}$ at a particular time m is only dependent on the noise components from the past values, not the future, so $E\left[y^{(m-1)} \varepsilon^{(m)} \right] = 0,$. Eq. (9) with Eq. (18) yield

$$\begin{aligned} E\left[z^{(m)} \right] &= 0 \\ E\left[z^{(m1)} z^{(m2)} \right] &= \left\langle \vec{b}_1, E\left[\vec{\varepsilon}^{\,(m1)} \otimes \vec{\varepsilon}^{\,(m2)} \right] \vec{b}_1 \right\rangle = 0, \quad m1 > m2, \end{aligned} \tag{21}$$

guaranteeing that Eq. (20) follows similarly and can be rewritten as

$$\gamma^{(1)} = \lambda_1 \gamma^{(0)}, \tag{22}$$

where

$$\gamma^{(k)} = E\left[y^{(m-k)}y^{(m)}\right]. \tag{23}$$

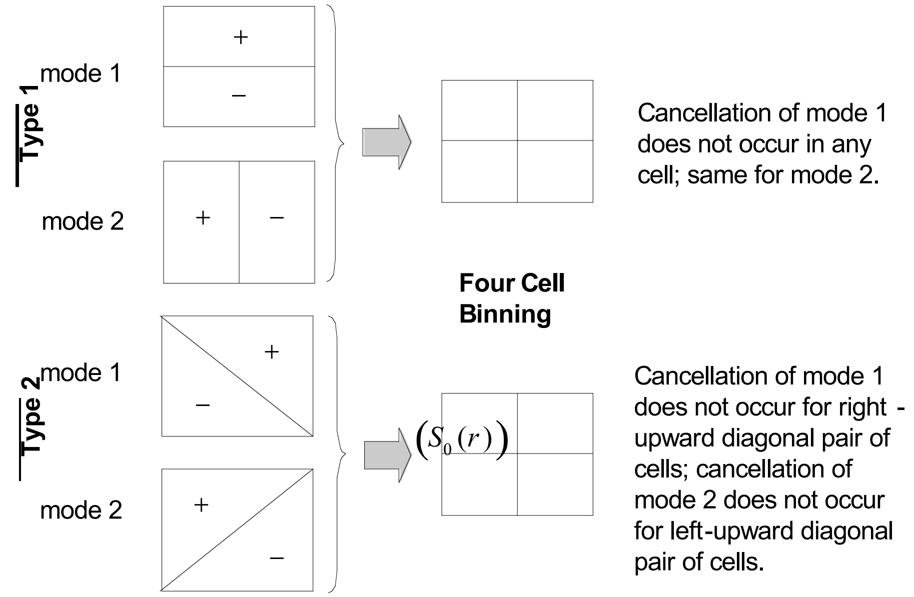
The dominance ratio λ_1 can then be solved for as

$$\lambda_1 = \frac{\gamma^{(1)}}{\gamma^{(0)}}. \tag{19}$$

2.1.3 Coarse-Mesh Projection

We next develop the idea of the coarse-mesh projection based on the relation between source binning and the first-mode eigenfunction $S_1(\vec{r})$. Previous work has shown [3] that if the complete cancellation of $S_1(\vec{r})$ occurs at a bin, the fluctuation mode associated with k_1/k_0 disappears for the source at the bin. The goal then is to find a binning scheme of general nature that always prevents the complete cancellation of $S_1(\vec{r})$ from occurring.

The fundamental-mode eigenfunction $S_0(\vec{r})$ is everywhere non-negative, i.e. does not change sign. The first-mode eigenfunction $S_1(\vec{r})$ is of anti-symmetric nature, i.e. changes sign once if simply put. The simplest way to ensure that $S_1(\vec{r})$ never completely cancels out is to put a two-bin mesh along each possible Cartesian direction. Two bins are sufficient for a one-dimensional problem, four bins for two-dimensions, etc. Figure 1 illustrates this in a two-dimensional case. This bin arrangement even works for problems with diagonal symmetry, as shown in Type 2 of the figure. Of course, more bins could be used along each direction, but this would only add to the complexity and storage requirements of the problem.



Cancellation of mode 1 does not occur in any cell; same for mode 2.

Cancellation of mode 1 does not occur for right - upward diagonal pair of cells; cancellation of mode 2 does not occur for left-upward diagonal pair of cells.

Figure 1: Four Cell Binning & the Dominant Eigenmodes
Next to Fundamental Eigenmode in Two-Dimensional Problems

2.2 Implementation into MCNP

The CMM algorithm implemented into MCNP was written primarily with memory conservation in mind. There was a significant obstacle to this, however. Since the projection vector is the eigenvalue of $\mathbf{A}_0$, it cannot be computed until after the run is completed (when $\mathbf{A}_0$ is formed). According to Eq. 16, however, the projection vector must be applied to the fission source vector every cycle. It would be completely counterproductive to save the fission source for every bin and every cycle, just to apply the projection vector subsequently. Instead, the terms in Eq. 19 are rearranged so that only the necessary terms are saved each cycle. Writing Eq. 19 in terms of the definition from Eq. 17 yields

$$\lambda_1 = \frac{\gamma^{(1)}}{\gamma^{(0)}} = \frac{E\left[y^{(m-1)} \cdot y^{(m)}\right]}{E\left[y^{(m-1)} \cdot y^{(m-1)}\right]}. \tag{20}$$

The individual terms can be expanded as

$$\frac{E\left[y^{(m-1)} \cdot y^{(m)}\right]}{E\left[y^{(m-1)} \cdot y^{(m-1)}\right]} = \frac{E\left[\left\langle \vec{d}_1, \vec{e}^{\,(m-1)} \right\rangle \cdot \left\langle \vec{d}_1, \vec{e}^{\,(m)} \right\rangle\right]}{E\left[\left\langle \vec{d}_1, \vec{e}^{\,(m-1)} \right\rangle \cdot \left\langle \vec{d}_1, \vec{e}^{\,(m-1)} \right\rangle\right]} \tag{21}$$

$$= \frac{\left\langle \vec{d}_1 \otimes \vec{d}_1, E\left[\vec{e}^{(m-1)} \otimes \vec{e}^{(m)}\right]\right\rangle}{\left\langle \vec{d}_1 \otimes \vec{d}_1, E\left[\vec{e}^{(m-1)} \otimes \vec{e}^{(m-1)}\right]\right\rangle}. \tag{22}$$

In this way, the outer product of the source fluctuation term and the projection vector can be computed separately and combined after the run is completed. The only values that need to be stored throughout the run are the outer product of the source fluctuation, which is a $p \times p$ matrix (*p* being the number of bins). This p^2 storage requirement is similar to that of the fission matrix method, the difference being that CMM actually prefers a coarser mesh than a finer one.

The error inherent in the autocorrelation coefficient calculation is measured by a technique described by Box and Jenkins [4]. The variance of an estimated autocorrelation coefficient of a stationary Normal process is given as

$$\operatorname{var}\left[r_k\right] \approx \frac{1}{N} \sum_{\upsilon=-\infty}^{+\infty} \left\{\rho_\upsilon^2 + \rho_{\upsilon+k}\rho_{\upsilon-k} - 4\rho_k\rho_{\upsilon-k} + 2\rho_\upsilon^2\rho_k^2\right\}. \tag{23}$$

For autocorrelation functions that damp out exponentially, such as in the AR(k) case where $\rho_k = \phi^{|k|}$, $(-1 < \phi < 1)$, this becomes

$$\operatorname{var}\left[r_k\right] \approx \frac{1}{N}\left[\frac{\left(1+\phi^2\right)\left(1-\phi^{2k}\right)}{1-\phi^2} - 2k\phi^{2k}\right]. \tag{24}$$

In particular, for an AR(1) case, this reduces to

$$\operatorname{var}\left[r_1\right] \approx \frac{1}{N}\left(1-\phi^2\right). \tag{25}$$

This is the expression implemented into MCNP for estimating the variance of the autocorrelation coefficient, i.e. DR.

It is important to note that the current implementation of CMM can only find coefficients for an AR(1) fitting. To expand the calculation to fitting orders greater than one, MCNP would have to save the fission data from more cycles, thus increasing the storage requirement. Specifically, to perform an AR(k) fit of the fission source MCNP must save data from *k+1* cycles, thus increasing the storage requirement to p^{k+1}. While this is not a concern for estimating DR using CMM (since it only requires an order one fit), it might become a problem if this work is extended to compute higher order eigenvalue ratios.

The other difficulty encountered when implementing CMM into MCNP was the need for several matrix manipulation routines, namely a matrix inverter and an eigenvalue/vector solver. While these are matrix operations are fairly commonplace, the computer routines necessary to perform them are not trivial, especially when dealing with non-symmetric matrices with complex eigenvalues/vectors.

3 Results

Once implemented, CMM was benchmarked against several problems for validation. The first problem (Figure 2) is a two-dimensional inhomogeneous problem with a DR of 0.9993 ± 0.0004 (2σ) taken from previous work [XX]. Figure 3 shows the DRs computed with different numbers of cycles and particles/cycle. Figure 4 shows the DRs computed using different bin arrangements. All results include a 95% confidence interval around the estimate.

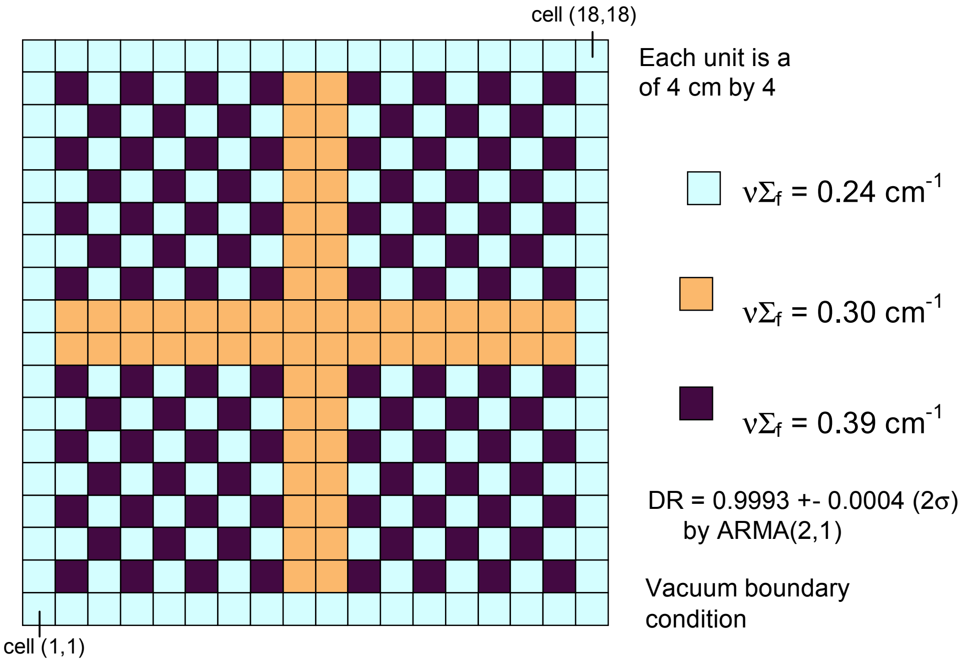

cell (18,18)
Each unit is a
of 4 cm by 4
$\nu\Sigma_f$ = 0.24 cm^{-1}
$\nu\Sigma_f$ = 0.30 cm^{-1}
$\nu\Sigma_f$ = 0.39 cm^{-1}
DR = 0.9993 +- 0.0004 (2σ)
by ARMA(2,1)
Vacuum boundary
condition
cell (1,1)

Figure 2: Description of Problem 1

Figure 3 shows the DR calculation using a $2\times2\times2$ mesh (two bins in the x, y, z directions, respectively) on Problem 1 using 10,000 particles per cycle and a varying number of cycles. In all cases, the 95% CI contains the benchmark DR. It is interesting to note that the error bounds actually grow larger when using 2,000 cycles as opposed to 1,000 cycles. This is because of the particular error formula used. Problems with DRs near unity will have a very small variance, especially when a particular estimate fluctuates close to unity.

Figure 4 shows DR for Problem 1 using varying mesh sizes, with 10,000 particles per cycle and 500 cycles. Again, in all cases the 95% CI contains the benchmark DR. The values and variances are all roughly the same, which is expected. Even when a $3\times3\times2$ mesh is reduced to $2\times2\times2$, DR is computed correctly. This shows that even when there are small cancellation effects caused by an odd bin combination, CMM is still able to accurately determine DR from the remaining portion of the eigenmode projection vector. This is important to ensuring the method works ‘black-box’ since users will be able to setup their own mesh. It is only when the integral of the eigenmode completely cancels out over the domain that CMM would not be able to extract DR, and this should not occur with the $2\times2\times2$ scheme introduced.

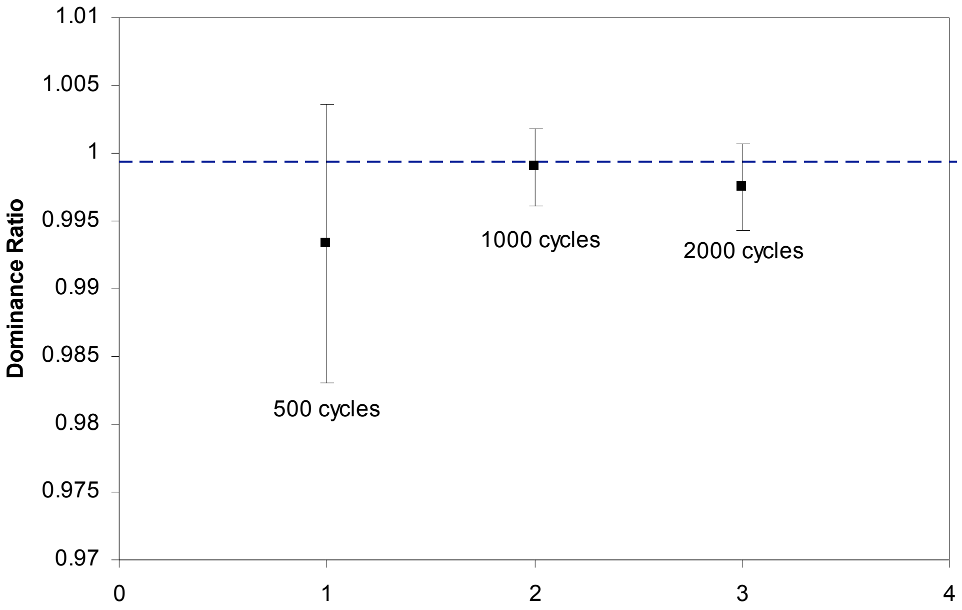


Figure 3: Dominance Ratio of Problem 1 using varying a number of cycles

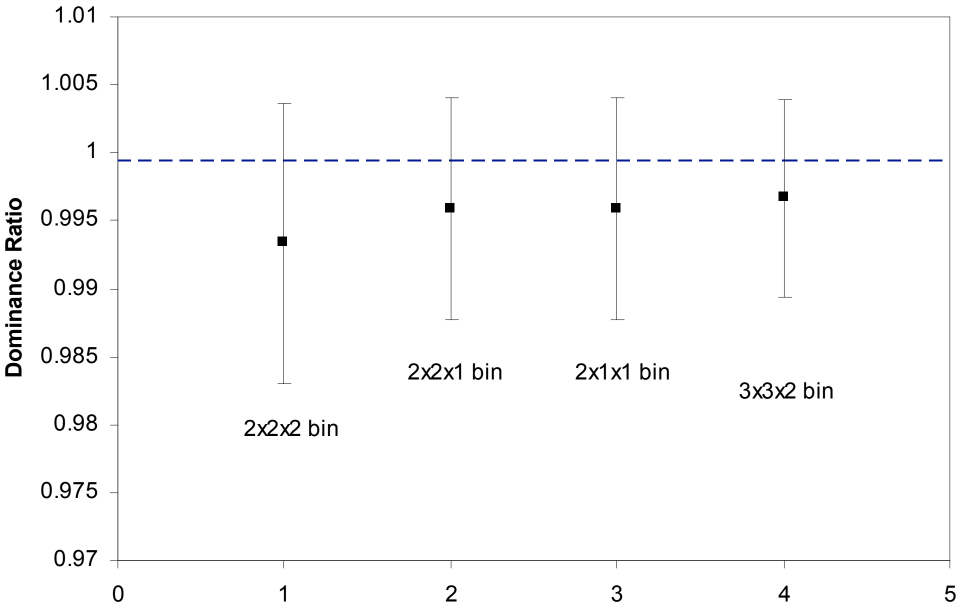


Figure 4: Dominance Ratio of Problem 1 using varying bin schemes

The second problem is a one-dimensional criticality safety problem with fissionable material at both ends of slab, separated by scattering and absorbing material. The regions are almost completely decoupled, raising DR significantly and skewing the shapes of the eigenmodes. This particular problem has two variations, which are labeled 2-1 and 2-2 in Figure 5. The only difference between the two variations is that 2-2 is slightly asymmetric, the right most material being 1.01 cm thick.

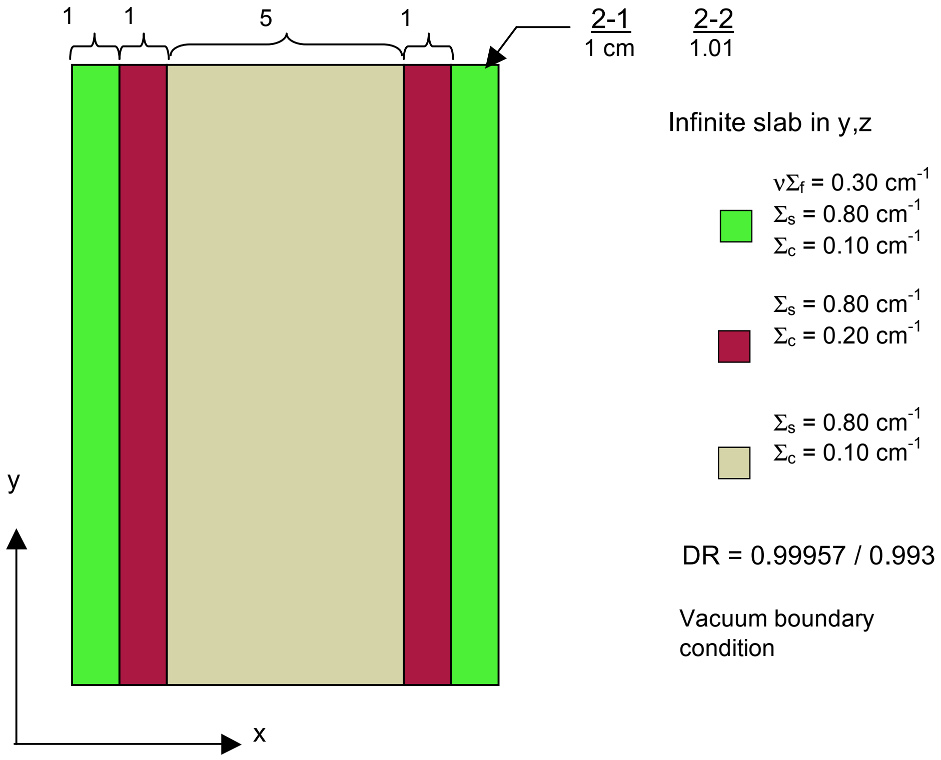

1
1
5
1
2-1
1 cm
2-2
1.01
Infinite slab in y,z
$\nu\Sigma_f$ = 0.30 cm^{-1}
Σ_s = 0.80 cm^{-1}
Σ_c = 0.10 cm^{-1}
Σ_s = 0.80 cm^{-1}
Σ_c = 0.20 cm^{-1}
Σ_s = 0.80 cm^{-1}
Σ_c = 0.10 cm^{-1}
y
DR = 0.99957 / 0.993
Vacuum boundary condition
x

Figure 5: Description of Problem 2-1 and 2-2

Figures 6 & 8 show the DR calculation for Problems 2-1 and 2-2, respectively, using a $2\times2\times2$ mesh using 10,000 particles per cycle and a varying number of cycles. The benchmark is contained within all 95% CIs. Figures 7 & 9 show DR for Problems 2-1 and 2-2 using varying mesh sizes with 10,000 particles per cycle and 500 cycles. The benchmark is contained within the 95% CI in all cases except 2,000 cycles, though it is within the 99% CI (3σ).

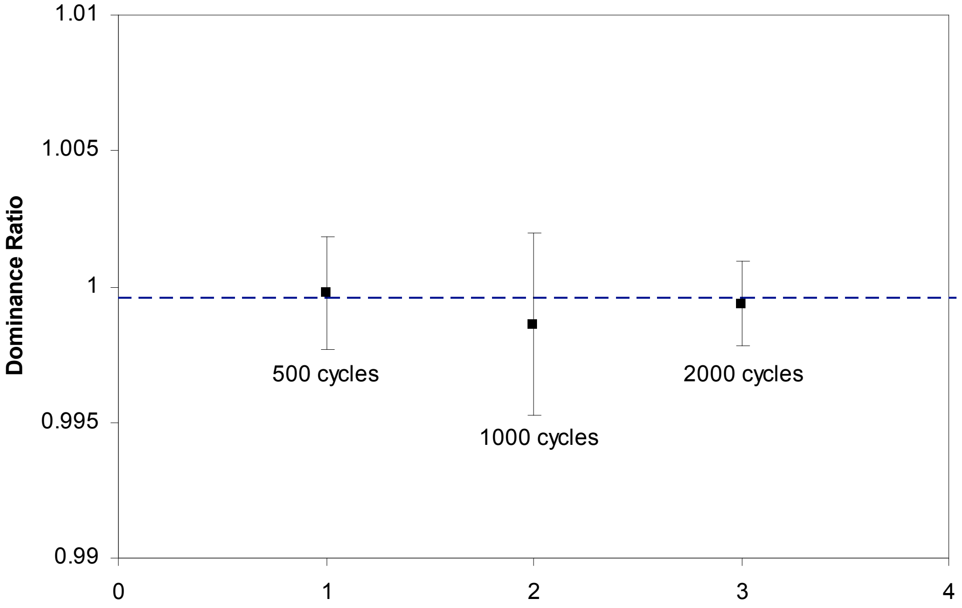


Figure 6: Dominance Ratio of Problem 2-1 using a varying number of cycles

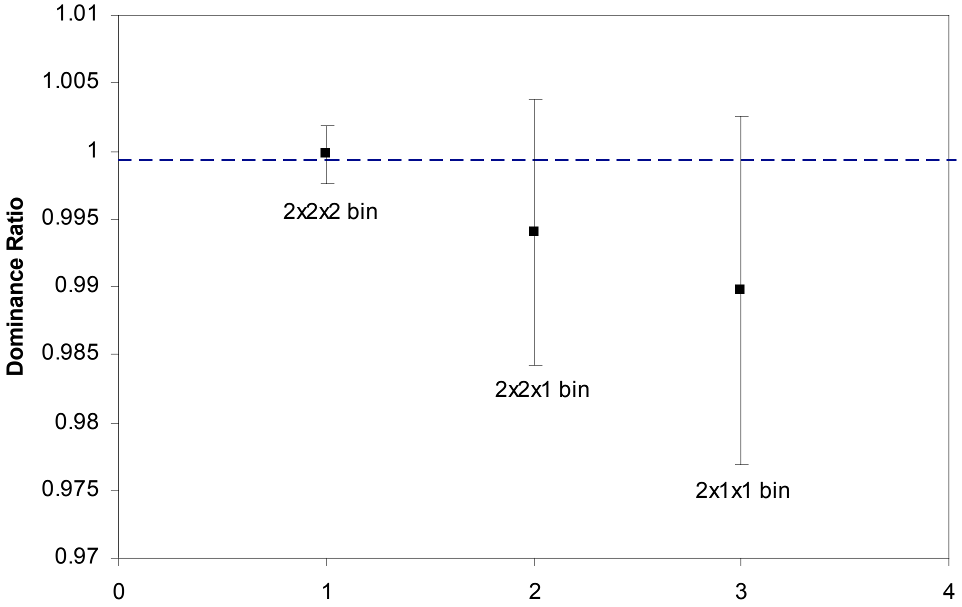


Figure 7: Dominance Ratio of Problem 2-1 using varying bin schemes

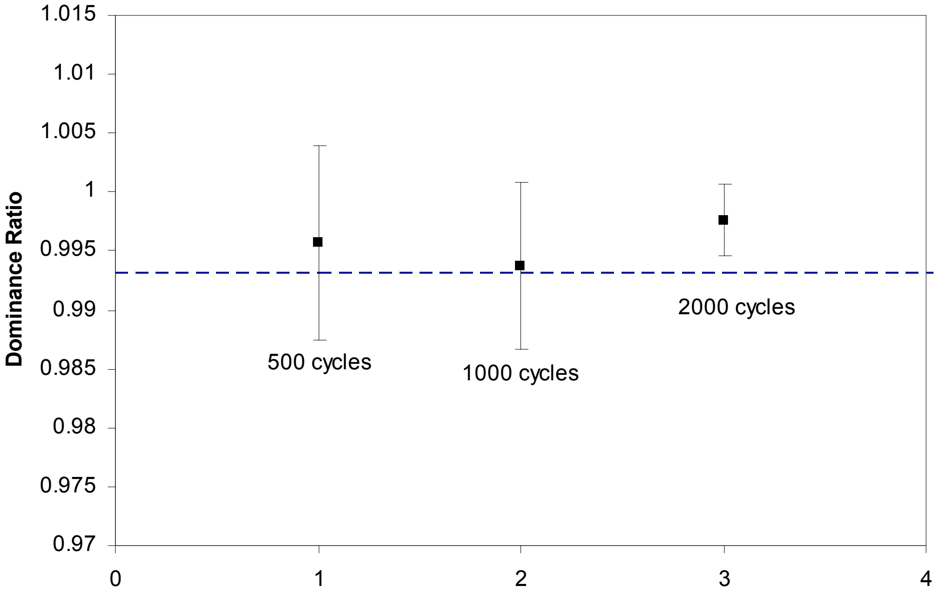


Figure 8: Dominance Ratio of Problem 2-2 using a varying number of cycles

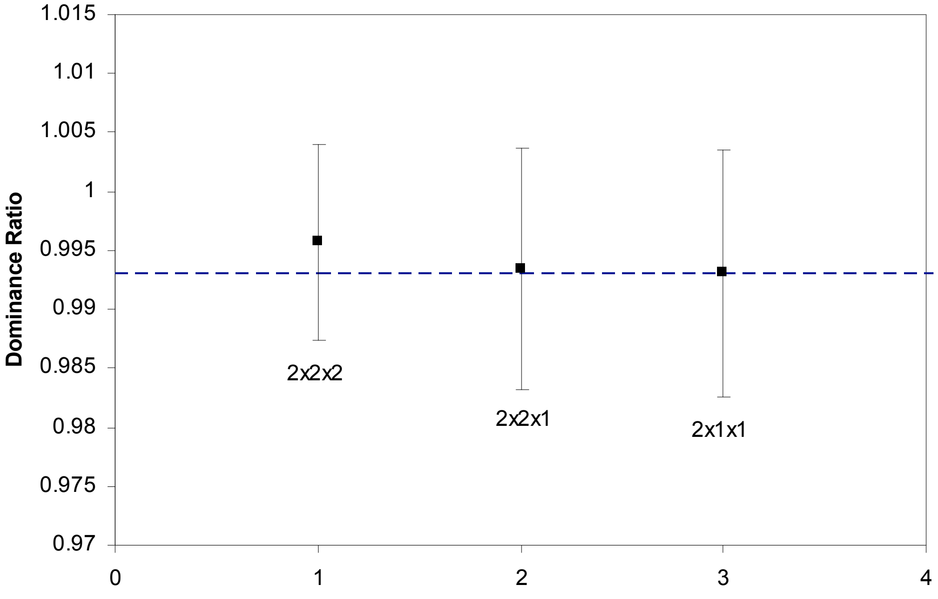


Figure 9: Dominance Ratio of Problem 2-2 using varying bin schemes

4 Conclusions

The implementation of CMM into MCNP was successful using a $2\times2\times2$ bin scheme. The coarse mesh utilized by this method is preferred to methods that require a fine mesh because of memory storage concerns and easier matrix manipulations. The results are positive for the problems tested, including criticality safety problems with distorted fission source distributions. From this point, a natural extension of this work would be to calculate the higher eigenmode ratios. This might require a finer mesh (depending on the eigenvalue ratio desired) as well as the capability for higher autoregressive order fittings.

References

1. T. Ueki, F.B. Brown, D.K. Parsons, and J.S. Warsa, “Time Series Analysis of Monte Carlo Fission Sources: I. Dominance Ratio Computation,” *Nuclear Sciences and Engineering*, **148**, 374-390 (2004).

2. B.R. Nease and T. Ueki, “Time Series Analysis of Monte Carlo Fission Sources – III: Coarse Mesh Projection,” *Nuclear Science and Engineering*, **157**, 51-64 (2007).

3. T. Ueki, F.B. Brown, D.K. Parsons, and D.E. Kornreich, “Autocorrelation and Dominance Ratio in Monte Carlo Criticality Calculations,” *Nuclear Science and Engineering*, **145**, 279-290 (2003).

4. G.E.P. Box, G.M. Jenkins and G.C. Reinsel, *Time Series Analysis: Forecasting and Control,* Prentice-Hall, Inc., Upper Saddle River, New Jersey (1994).