LA-UR-12-22320
Approved for public release; distribution is unlimited.

Title: Fission Matrix Capability for MCNP Monte Carlo

Author(s): Carney, Sean E.
Brown, Forrest B.
Kiedrowski, Brian C.
Martin, William R.

Intended for: 2012 ANS Winter Meeting, 2012-11-11/2012-11-15 (San Diego, California, United States)
Web

Fission Matrix Capability for MCNP Monte Carlo

Sean Carney[1], Forrest Brown[2], Brian Kiedrowski[2], William Martin[1]

[1]*Dept. Nuclear Engineering & Radiological Science, Univ. of Michigan, Ann Arbor, MI, seanec@umich.edu*
[2]*Monte Carlo Codes Group, LANL, Los Alamos, NM, fbrown@lanl.gov*

INTRODUCTION

This summary describes the initial experience and results from implementing a fission matrix capability into the MCNP Monte Carlo code [1]. The fission matrix is obtained at essentially no cost during the normal simulation for criticality calculations. It can be used to provide estimates of the fundamental mode power distribution, the reactor dominance ratio, the eigenvalue spectrum, and higher mode spatial eigenfunctions. It can also be used to accelerate the convergence of the power method iterations. Past difficulties and limitations of the fission matrix approach are overcome with a new sparse representation of the matrix, permitting much larger and more accurate fission matrix representations.

Continuous-energy Monte Carlo codes simulate neutron behavior using the best available nuclear data, accurate physics models, and detailed geometry models. Reactor criticality calculations for k_{eff} and the power distribution are carried out iteratively, using the power method, where batches of neutrons are simulated for a single generation. The first-generation fission neutrons produced in a batch become the starting neutron sites for the next batch. A suitable number of "inactive" initial batches are required to converge to the fundamental mode eigenvalue and eigenfunction, and then succeeding iterations with "active" batches are used to accumulate Monte Carlo tallies for estimating desired reaction rate distributions.

Most Monte Carlo codes perform the power iteration without acceleration and can sometimes exhibit very slow convergence. Statistical noise for batch results precludes the use of common outer iteration acceleration methods (e.g., Chebyshev). Also, since production Monte Carlo codes restrict neutron statistical weights to be non-negative, higher eigenmodes cannot be evaluated directly from the Monte Carlo neutron simulation.

The fission matrix approach was proposed in the earliest works on Monte Carlo criticality calculations [2-5] and has been tried by many researchers over the years. The present work takes advantage of the very large computer memories available today and a new sparse matrix representation to overcome past difficulties.

THEORETICAL BASIS OF THE FISSION MATRIX

The neutron transport equation can be written as

$$M \cdot \Psi(\vec{r},E,\hat{\Omega}) = \tfrac{1}{K} \cdot \chi(E,\hat{\Omega}) \cdot S(\vec{r}), \qquad (1)$$

where M is the net loss operator defined by

$$M \cdot \Psi(\vec{r},E,\hat{\Omega}) = \hat{\Omega} \cdot \nabla\Psi(\vec{r},E,\hat{\Omega}) + \Sigma_T(\vec{r},E)\Psi(\vec{r},E,\hat{\Omega}) - \iint dE'\, d\hat{\Omega}' \Sigma_S(\vec{r},E' \to E,\hat{\Omega}' \to \hat{\Omega})\Psi(\vec{r},E',\hat{\Omega}'),$$

S is the fission source, defined by

$$S(\vec{r}) = \iint dE'\, d\hat{\Omega}' \nu\Sigma_F(\vec{r},E')\Psi(\vec{r},E',\hat{\Omega}'),$$

and χ is the emission energy spectrum of fission neutrons. All other terms are defined in the usual way.

The Green's function for this problem is defined by the equation

$$M \cdot G(\vec{r},E,\hat{\Omega};\, \vec{r}_0,E_0,\hat{\Omega}_0) = \delta(\vec{r}-\vec{r}_0, E-E_0, \hat{\Omega}-\hat{\Omega}_0),$$

where the "0" subscript denotes an initial point in phase space, and δ is the Dirac delta function. It then follows that

$$\Psi(\vec{r},E,\hat{\Omega}) = \tfrac{1}{K} \cdot \iiint d\vec{r}_0\, dE_0\, d\hat{\Omega}_0\, \chi(E_0,\hat{\Omega}_0) \cdot S(\vec{r}_0) \cdot G(\vec{r},E,\hat{\Omega};\, \vec{r}_0,E_0,\hat{\Omega}_0) \qquad (2)$$

Now perform the following steps on Eq. (2): 1) Multiply by $\nu\Sigma_F(\vec{r},E)$ and integrate over all E and $\hat{\Omega}$, and 2) segment the problem into N spatial regions, and integrate over the volumes of each region *I*. These steps lead to

$$S_I = \tfrac{1}{K} \cdot \sum_{J=1}^{N} F_{I,J} \cdot S_J \qquad (3)$$

where

$$F_{I,J} = \frac{1}{S_J} \cdot \int_{\vec{r}\in V_I} d\vec{r} \int_{\vec{r}_0\in V_J} d\vec{r}_0 \iint dE\, d\hat{\Omega} \nu\Sigma_F(\vec{r},E) \cdot \iint dE_0\, d\hat{\Omega}_0 \chi(E_0,\hat{\Omega}_0) \cdot S(\vec{r}_0) \cdot G(\vec{r},E,\hat{\Omega};\, \vec{r}_0,E_0,\hat{\Omega}_0) \qquad (4)$$

The kernel $F_{I,J}$ is equal to the number of fission neutrons born in region *I* due to one average fission neutron starting in region *J*, and is called the *fission matrix*. The fundamental mode eigenvalue of this matrix is identical to the eigenvalue *K* in Eq. (1), and the fundamental mode eigenvector is the regionwise fission neutron source distribution.

The elements of the fission matrix can be estimated at essentially no extra cost during the normal Monte Carlo simulation – simply remember the region a fission neutron was born in (*J*), determine the region a next-generation fission neutron is produced in (*I*), and tally the *(I,J)*-th element of the fission matrix. The tallies need to be normalized by dividing each *(I,J)*-th element by the total number of starters in region *J*. Thus, the tallies for fission matrix elements can be made using only the locations of fission neutron sources at the start and end of each batch, without incurring any overhead during the random walk simulation of the neutrons. This approach also eliminates any inter-process communications overhead during MPI parallel processing, since the entire fission matrix estimation can be performed on the master node, using only the existing "fission bank" information.

It is also important to note that the fission matrix elements can be estimated even during inactive batches in the iteration process. That is, fission matrix elements can be estimated based solely on fixed-source calculations, even without knowing the converged fundamental mode distribution. Doing so, however, relies on the assumption that the fission matrix elements $F_{I,J}$ are insensitive to the local distribution of $S(\mathbf{r_0})$ in Eq. 4. With a fine enough spatial mesh for tallying the fission matrix, this assumption is valid; with a coarse mesh this approximation will introduce inaccuracies. This raises the prospect of determining the fission matrix during inactive batches, assuming a fine-enough spatial resolution, and using it to accelerate convergence of the Monte Carlo history simulation.

The principal limitation on the accuracy of the fission matrix approach is, and always has been, the size of the regions for each fission matrix element. Typically, a regular 3D spatial mesh with $N = N_I \times N_J \times N_K$ elements is used, giving an NxN fission matrix, with N^2 entries. A 100x100x100 spatial mesh would give rise to a fission matrix with 10^{12} elements, which could not be stored even on today's computers.

SPARSE FISSION MATRIX REPRESENTATION

As noted in the Theory section above, the principal disadvantage of using a fission matrix is the matrix size, which grows as the square of the number of spatial mesh regions. For the 60x60x1 mesh case discussed above, the fission matrix requires about 0.1 GB of memory storage. Extending this case to 100 axial mesh increments would increase the storage requirements to about 1000 GB, which is excessive.

To overcome this limitation, we are investigating the use of a sparse fission matrix. Clearly, not every region in a large 3D problem is tightly coupled to every other region; fission neutrons induce most further fissions in neighboring regions, and few or none in distant regions. To investigate this, we have incorporated tallies into MCNP to diagnose the fractions of induced fission neutrons in neighboring regions, and have examined the structure of the fission matrix for a typical 2D PWR problem. Fig. 1 shows the structure of the fission matrix for the 15x15x1 mesh case, where each mesh element corresponds to an assembly-sized region. It is evident from the banded structure of the fission matrix that neutrons from one assembly cause nearly all of their fissions in that assembly and the nearest 2 neighboring assemblies in each direction. Only about 0.5% of the fission matrix tallies correspond to more distant fissions. Thus, storing a sparse, banded fission matrix (rather than the full matrix) offers a promising mechanism for mitigating the storage problem. Of course, as the spatial mesh is refined, more neighbor bands will need to be retained. For the testing described below, we have used the sparse fission matrix representation, with the number of bands in the matrix chosen to include spatial regions corresponding to the nearest 2 neighboring assemblies in each direction. That is, for a 15x15x1 spatial mesh, the fission matrix sparse storage is 225x25 elements, rather than the full 225x225 elements. For the 30x30x1 mesh, the sparse matrix representation is 900x81; for the 60x60x1 mesh, the matrix is 3600x289. Additionally, the very few tallies outside of the matrix bands were accumulated in the nearest banded fission matrix element in order to preserve overall neutron balance.

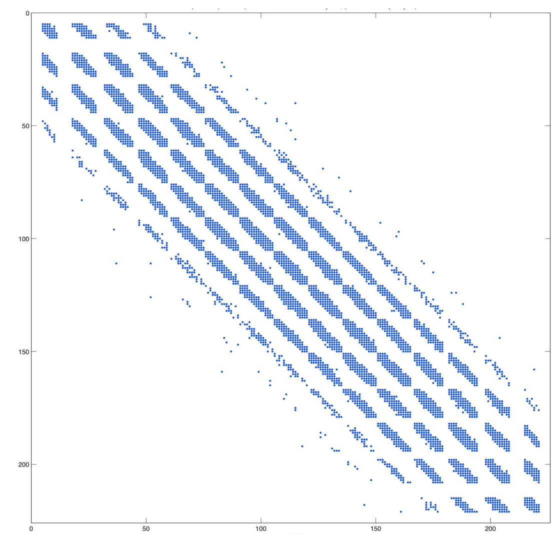

Fig. 1. Fission matrix structure for a 2D whole-core PWR model, for a 15 x 15 x 1 spatial mesh. Matrix dimensions are 225 x 225. Points in blue are non-zero elements of the fission matrix.

INITIAL TESTING RESULTS

The fission matrix capability was implemented in a local version of MCNP5, with both full and sparse matrix representations. All of the testing results described below were accomplished using a 2D whole-core PWR model shown in Fig. 2 (previously used in [6], based on [7]) with ENDF/B-VII.0 continuous-energy nuclear data. The fission matrix was accumulated during standard KCODE calculations with 500K neutrons/batch. Tallies for the fission matrix elements were made only for the 4th and successive batches. k_{eff}, the fundamental mode eigenfunction, and the dominance ratio from the fission matrix were determined via an iterative method. Higher-

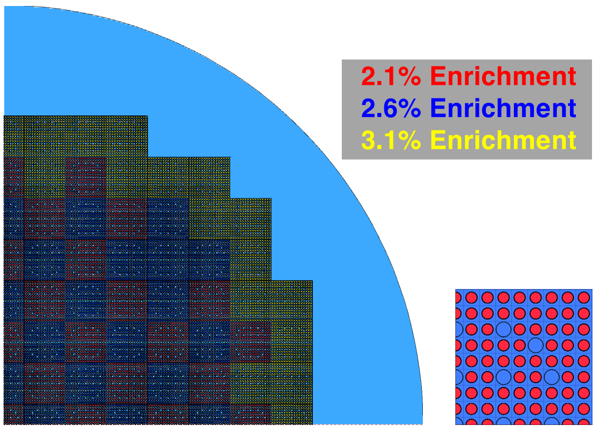

Fig. 2. 2D quarter-core PWR model, with detail shown for the center 1/4-assembly. For the current work, a whole-core model was used.

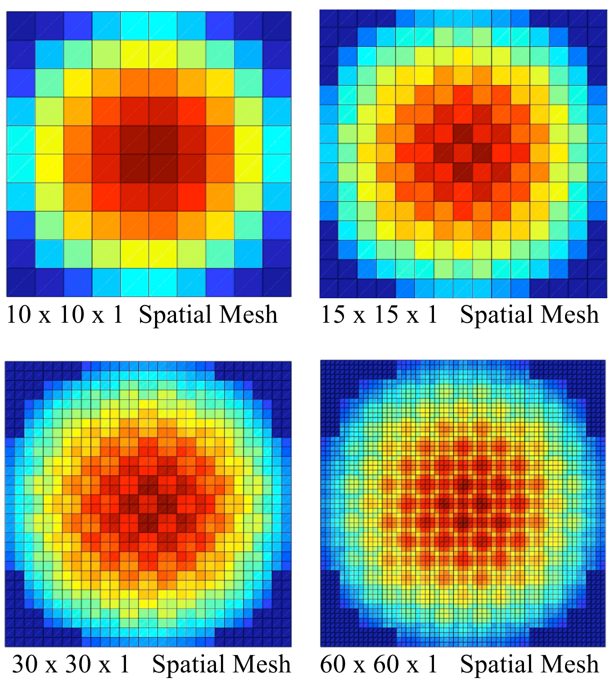

Fig. 3. Fundamental eigenmode of the fission matrix for a 2D whole-core PWR model, for various spatial meshes used to tally the fission matrix.

mode eigenvalues and eigenfunctions for the fission matrix were determined by either a direct non-symmetric matrix routine or by using Matlab.

Fig 3. shows the fundamental mode eigenfunction for various spatial resolutions used in tallying the fission matrix. Note that the 15x15x1 mesh corresponds to assembly-size elements, the 30x30x1 mesh corresponds to quarter-assembly-size elements, and the 60x60x1 mesh corresponds to 1/16th-assembly-size elements. The

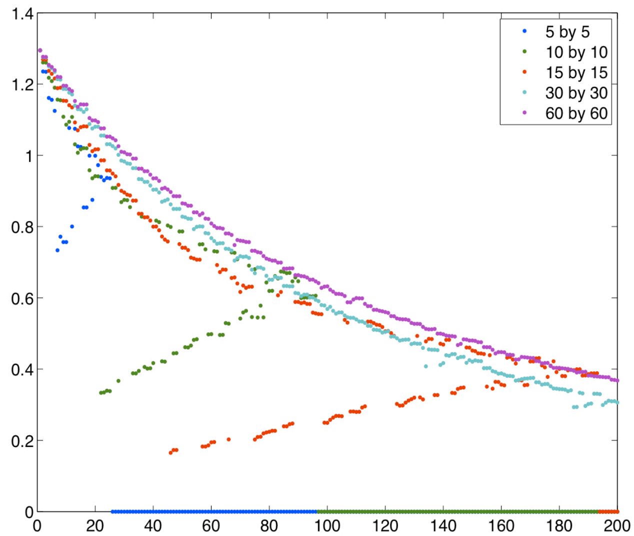

Fig. 4. Eigenvalue spectrum of the fission matrix for a 2D whole-core PWR model, for various spatial resolutions.

10x10x1 mesh does not match assembly geometry, and is shown as an example of a poor choice of meshes.

Fig. 4 shows the eigenvalue spectra for various choices of spatial resolution in determining the fission matrix. All of the eigenvalues are shown for the first 2 cases (25 or 100), and only the first 200 for the others. Conventional intuition suggests that all eigenvalues of the fission matrix should be real and discrete. Because the fission matrix is non-symmetric, complex eigenvalues can arise, and indeed are evident in the anomalous low values in the spectra plots for the 5x5x1 and 10x10x1 cases. (Only the real parts of the eigenvalues are plotted in Fig. 4. The anomalous low values correspond to the eigenvalues having a complex component.) For the 15x15x1 case, the first 40 eigenvalues are well-behaved, and complex eigenvalues only appear for the much higher modes. It appears that complex eigenvalues only arise if the spatial mesh for tallying the fission matrix is too coarse, and/or statistical noise from the Monte Carlo tallies leads to non-physical solutions for the eigenvalue solution. The 30x30x1 and 60x60x1 cases each behave as expected, and it is evident from the figure that at least the lowest-mode portion of the spectrum converges as the spatial mesh is refined.

Fig. 5 shows the fundamental eigenmode (i.e., the fission neutron source distribution) and 11 higher eigenmodes for the 60x60x1 mesh case. These plots are especially interesting, since the higher eigenmodes cannot normally be obtained directly from a Monte Carlo calculation. The 1st and 2nd higher modes show a slight tilt that is being investigated. It may be a result of incomplete convergence that was not apparent from standard diagnostics of the problem convergence, an artifact of

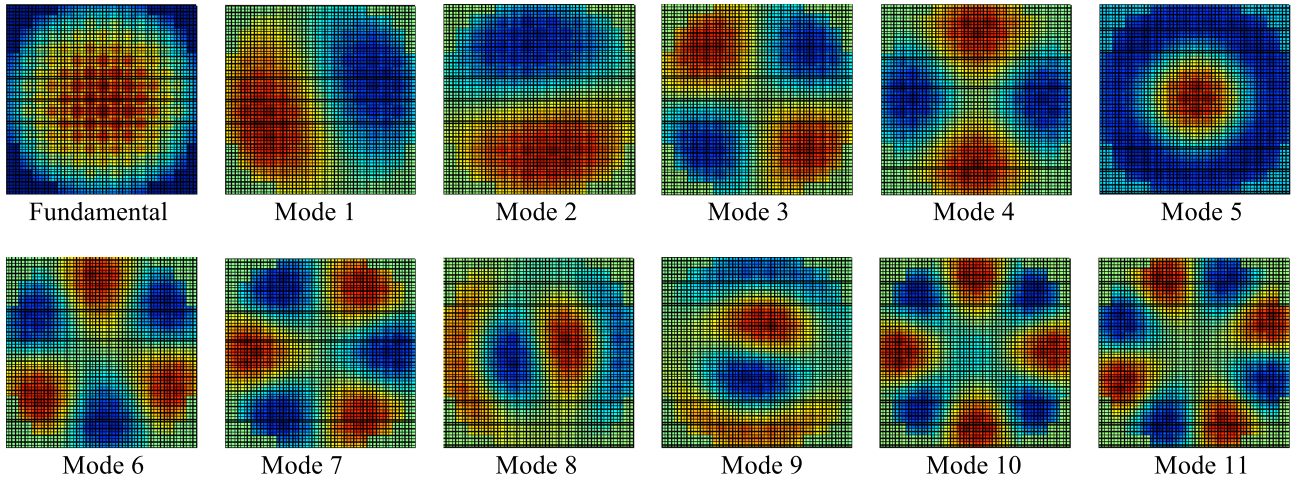

Fig. 5. Fundamental and higher eigenmodes for 2D whole-core PWR model obtained from a fission matrix using a 60x60x1 spatial mesh. Sparse matrix size was 3600x17.

either tallying or plotting, or the effect of statistical noise on the eigenmode calculation.

WORK IN PROGRESS

We are also investigating the use of the fission matrix to accelerate the power method convergence of Monte Carlo criticality calculations. Because the fission matrix can be determined accurately with only a few batches during the inactive portion of the calculation, the fundamental eigenmode can be used to bias the fission neutron source, forcing the source distribution based on Monte Carlo histories to converge more quickly. Initial testing of this method is encouraging, and further study and development are in progress.

SUMMARY & CONCLUSIONS

Initial implementation and testing of a fission matrix capability in MCNP5 has demonstrated that the method can be used to obtain interesting and valuable information for reactor physics analysis, including the higher mode eigenvalues and eigenfunctions. The fission matrix approach is well-founded in theory, does not significantly increase the cost of standard Monte Carlo criticality calculations, and does not significantly increase code complexity. The accuracy of the method improves as the spatial mesh is refined. Detailed 2D representations are performed easily on today's computers, and a sparse fission matrix representation is being investigated to permit scaling to detailed 3D problems.

REFERENCES

1. X-5 MONTE CARLO TEAM, "MCNP – A General N-Particle Transport Code, Version 5 – Volume I: Overview and Theory", LA-UR-03-1987, Los Alamos National Laboratory (April, 2003).
2. K.W. MORTON, "Criticality Calculations by Monte Carlo Methods", United Kingdom Atomic Energy Research Establishment, Harwell, Report T/R-1903 (1956).
3. E.L. KAPLAN, "Monte Carlo Methods for Equilibrium Solutions in Neutron Multiplication", Lawrence Radiation Laboratory, UCRL-5275-T (1958).
4. J.M. HAMMERSLEY & D.C. HANDSCOMB, Monte Carlo Methods, Chapter 8, Methuen & Co. (1964).
5. G.W. MORRISON, J.T. MIHALCZO, D.C. IRVING, "REACT and CONVRG – Fortran Subroutines for Determining Source Convergence for the O5R Monte Carlo Neutron Transport Code", Oak Ridge National Laboratory, ORNL-TM-1325 (1966).
6. F.B. BROWN, "A Review of Best Practices for Monte Carlo Criticality Calculations", ANS NCSD-2009, Richland, WA, Sept 13-17 (2009).
7. M. NAKAGAWA & T. MORI, "Whole Core Calculations of Power Reactors by use of Monte Carlo Method", *J. Nuc. Sci.and Tech.*, **30** [7], pp 692-701 (1993).