

## LA-UR-19-23887

Approved for public release; distribution is unlimited.

Title: Automated Acceleration and Convergence Testing for Monte Carlo  
Criticality Calculations

Author(s): Brown, Forrest B.  
Josey, Colin James  
Henderson, Shawn  
Martin, William R.

Intended for: ANS Mathematics & Computation - MCD 2019, 2019-08-25/2019-08-29  
(Portland, Oregon, United States)  
MCNP documentation

Issued: 2019-04-30

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# AUTOMATED ACCELERATION AND CONVERGENCE TESTING FOR MONTE CARLO CRITICALITY CALCULATIONS

Forrest Brown<sup>1</sup>, Colin Josey<sup>1</sup>, Shawn Henderson<sup>2</sup>, and William Martin<sup>3</sup>

<sup>1</sup>Monte Carlo Methods, Codes, & Applications, LANL  
PO Box 1663, MS A143, Los Alamos, NM, 87544

<sup>2</sup>Sandia National Laboratory, PO Box 5800, Albuquerque, NM, 87185

<sup>3</sup>Nuclear Engineering & Radiological Science, Univ. of Michigan  
2355 Bonisteel Blvd, Ann Arbor, MI, 48109

fbrown@lanl.gov, cjoyse@lanl.gov, sjhende@sandia.gov, wrm@umich.edu

## ABSTRACT

For over 60 years, Monte Carlo criticality calculations for  $k_{eff}$  and the fission distribution have been solved using the power iteration method. Significant burdens are placed on code users, however, to ensure that (1) a sufficient number of neutrons per cycle is used to prevent bias in  $k_{eff}$  and errors in the fission source shape, and (2) a sufficient number of initial cycles is discarded to ensure that  $k_{eff}$  and the fission source have converged to the steady-state, stationary distribution. Recent work has addressed those burdens, providing automated acceleration of the convergence process, statistical tests for automatically determining convergence, and additional tests to assess whether a sufficient number of neutrons per cycle was used. These automated methods provide adaptive meshing and a fission matrix reference solution, and provide quantitative evidence of convergence. Testing on a wide range of problems has demonstrated that the methods are robust and reliable.

KEYWORDS: MCNP, neutron transport, fission matrix

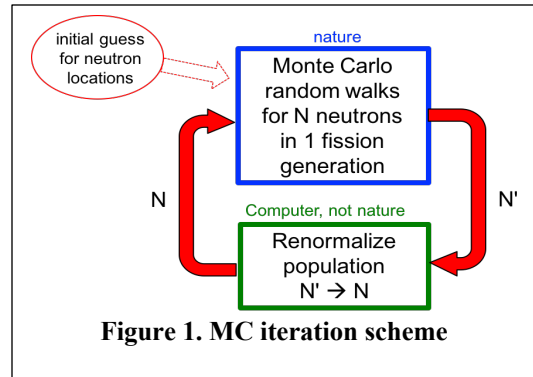
## 1. INTRODUCTION

For over 60 years, Monte Carlo (MC) criticality calculations for  $k_{eff}$  and the fission distribution have been solved using the power iteration method, also called the method of successive iterations [1,2].

- A generation-based iteration scheme is used, and the neutron population is renormalized between successive iterations, as shown in Figure 1. Each generation is started with the same total number of neutrons  $N$  (or equivalently for *mcnp6*, total weight) and a number of neutrons  $N'$  is produced.  $N'$  is a stochastic variable, hence the renormalization to  $N$  neutrons for the next generation is a biased process. Further, the renormalization reduces the number of independent neutron fission chains, introducing correlation among cycles [3,4]. The correlation manifests as clustering for very low  $N$ .
- The bias in  $k_{eff}$  introduced by the renormalization process is negative and proportional to  $1/N$  [5,6]. The eigenfunction (i.e., fission distribution) is also biased proportional to  $1/N$ , with

additional complications of being too low in high-importance regions and too high in low-importance regions.

- The MC iteration scheme begins with an initial guess for  $k_{eff}$  and the fission distribution. Iterations (called inactive cycles) are performed without tallies until  $k_{eff}$  and the fission distribution have converged to their stationary state. After convergence, tallies are turned on, and iterations (called active cycles) are continued until sufficiently small uncertainties are obtained for desired results.



Significant burdens are placed on nuclear criticality safety (NCS) analysts to properly run the MC calculations:

- The initial guess for the fission source locations must be defined by user input. Ideally, the initial guess should be similar to the steady-state stationary distribution (i.e., the final solution). Care must be taken to select the initial starting sites such that there are some fission sites located in each fissionable region of the problem.
- Users must ensure that a sufficient number of neutrons per cycle is used to prevent bias in  $k_{eff}$  and errors in the fission source shape. For small to moderate sized problems, it has been shown that using  $N > 10,000$  neutrons per generation effectively removes the bias in  $k_{eff}$  [7]. For larger physical systems, 100,000 or 1 M or more neutrons per generation may be needed.
- Users must also ensure that a sufficient number of initial cycles is discarded so that  $k_{eff}$  and the fission source have converged to the steady-state, stationary distribution. In a typical traditional calculation, a short trial run is made to determine the number of inactive cycles based on plots of  $k_{eff}$  and the Shannon entropy of the fission source distribution,  $H$ . Then the number of inactive cycles to achieve convergence is manually set in the *mcnp6* input file. Then a final run is made to determine results, with additional cycles run until the uncertainties on results are small enough.

While the procedures for running MC criticality problems properly are straightforward, they can be burdensome when many different problems need to be analyzed. It is common in NCS work to perform parameter studies, varying one or more parameters such as spacing or density and then running a MC criticality problem. Often 100s or 1,000s of runs must be made to span the range of expected conditions. For these parameter studies, it is not practical to follow all of the recommended procedures discussed above, and typically very conservative over-estimates are used for the number of inactive cycles, resulting in excessive computer time. Additionally, determining proper convergence of the iteration process using plots of  $k_{eff}$  and the  $H$  is highly subjective due to the statistical variations in cycle-wise variations in  $k_{eff}$  and the  $H$ , and most users will conservatively choose the number of inactive cycles much larger than necessary. Finally, there have been no tests available for determining whether a sufficient number of neutrons per cycle was used to reduce the renormalization bias to negligible levels (other than repeated calculations with different numbers of neutrons per cycle, an extremely burdensome task).

Recent work has addressed these burdens, providing automated acceleration of the convergence process, statistical tests for automatically determining convergence, and additional tests to assess whether a sufficient number of neutrons per cycle was used. These automated methods determine mesh spacing (with no user input) from cycle 1 estimates of physics results, provide adaptive meshing and a fission matrix reference solution, and provide quantitative evidence of convergence. Testing on a wide range of problems has demonstrated that the methods are robust and reliable.

This paper provides details on the tallying and solution of the fission matrix to provide a reference solution, the choice of mesh size for computing Shannon entropy and the fission matrix, and acceleration of fission source convergence. The statistical tests for determining convergence, the tests for assessing adequate population size, and the wide variety of test problems are discussed briefly, with more detail provided in a companion paper [8].

## 2. METHODS

A local version of *mcnp6.2.1* [9] was modified to:

- use physics-based problem data to create a mesh for convergence analysis,
- perform tallies of the fission matrix for all cycles after cycle 1 using sparse storage,
- monitor the stability of the fission matrix and solve for the fundamental mode eigenfunction after each block of  $M$  cycles,
- accelerate convergence of the fission source iterations,
- perform statistical tests for convergence and automatically activate the tallies, and
- assess whether a sufficient number of neutrons per generation was used.

Each of these procedures is described below.

### 2.1. Establishing the Mesh for Convergence Analysis and Adapting the Mesh

Previous work involving the fission matrix method reported in [10,11] focused primarily on determining higher-mode eigenfunctions and the eigenvalue spectrum. The mesh for tallying the fission matrix was specified by the user, and successive runs refining the mesh were performed until the eigenvalue spectrum converged. The mesh resolution required for an accurate eigenvalue spectrum was highly problem-dependent, and very fine meshing was needed to obtain detailed spatial resolution for the fundamental and higher-mode eigenfunctions.

For the current work in convergence testing and acceleration, a reference solution for the global shape is needed, and higher-mode eigenfunctions are not required. Only the fundamental mode eigenfunction is necessary for the methods described in the following sections.

For automatically determining adequate mesh resolution for the fundamental mode global solution, a physics-based parameter is desired. Since the fission matrix is essentially a collection of point-to-point Green's functions from neutron birth in fission to next-generation birth in fission, a reasonable choice for a metric for determining the mesh spacing is the RMS distance from birth to fission in a cycle,  $L_{fiss}$ .

$$L_{fiss} = \left( \sum_{n=1}^{N'} w_n |\vec{r}_n - \vec{r}_{0,n}|^2 / \sum_{n=1}^{N'} w_n \right)^{1/2}, \quad (1)$$

where  $\vec{r}_n$  is the location of a next-generation neutron born in fission,  $w_n$  is the weight of that neutron,  $\vec{r}_{0,n}$  is the location where the parent neutron started, and  $N'$  is the number of next-generation neutrons created in the cycle.  $L_{fiss}$  is trivial to calculate, using just the starting birth site and the resultant next-generation sites within a cycle, with tallies made only at the end of a cycle (i.e., there is no need to tally during the neutron random walks).

Examining the mesh used in the previous studies of higher-mode eigenfunctions and eigenvalue spectra in [10,11], the mesh resolutions required for accuracy in detailed eigenmodes and eigenvalue spectra were typically  $0.1 * L_{fiss}$  to  $0.2 * L_{fiss}$ , and sometimes even smaller. For the present work where only the global shape of the fundamental mode eigenfunction is needed, numerical experiments were performed on the test problems cited below and on 100s of ICSBEP benchmark problems to determine a recommended mesh resolution. It was found that using  $1.0 * L_{fiss}$  was suitable for all problems tested. That is, using a coarser

mesh did not adequately capture the global shape of the fission distribution; using a finer mesh required many additional iterations for the fission matrix to stabilize and produce a reliable solution. (If the mesh is too fine, then very many neutrons may be needed to produce fission matrix tallies that are not too noisy.) While there are user input options to override the factor applied to  $L_{fiss}$  for determining the mesh spacing, all of the testing discussed below used the default spacing of  $1.0 * L_{fiss}$ .

An adaptive Cartesian mesh is used for determining Shannon entropy,  $H_{neut}$ , of the fission source distribution,  $S_{neut}$ , and the starting/ending fission point bins for the fission matrix method. During the first (inactive) cycle, tallies are made to estimate the RMS distance to fission,  $L_{fiss}$ . While the value of  $L_{fiss}$  is not precise, since it is based on the initial source distribution guess and only 1 cycle of neutrons, it provides an adequate physics-based distance for setting a mesh size for convergence analysis. By default, a mesh spacing of  $L_{fiss}$  is used. Then a Cartesian mesh is defined for the axis-aligned bounding box of fission points at the end of the first cycle, with  $N = n_x n_y n_z$  mesh cells. The mesh storage is adaptive: If during subsequent cycles a source tally must be made in a location outside the mesh, then the appropriate mesh dimensions  $n_x$ ,  $n_y$ ,  $n_z$  are extended (keeping the same spacing) to include that region.

## 2.2. Fission Matrix Tallies

The fission matrix,  $F$ , is an  $N \times N$  matrix (where  $N$  is the number of mesh cells), potentially requiring  $N^2$  tally bins. However, in this implementation tallies for  $F$  are stored using a compressed-row-storage (CRS) scheme, such that only nonzero tallies are stored. If tallies are needed in an empty slot (i.e., a previously-zero location that is not stored), then the CRS tallies are automatically extended to include the new entries. The CRS tallies for  $F$  are also automatically re-indexed and extended if the underlying mesh is extended.

After cycle 1, inactive cycles proceed in the normal manner, with tallies of  $S_{neut}$  and  $F$  made at the end of each cycle. The tallies for  $F$  are cumulative, including both inactive and active cycles. During these cycles, the  $S_{neut}$  and  $F$  tallies may be extended if any neutron fission sites are found outside the previous mesh.

## 2.3. Fission Matrix Stabilization and Solution

The fission matrix equations are solved after each block of  $M$  cycles, where currently  $M$  defaults to 10 but can be overridden by user input.  $M$  is effectively a “window” for solving the fission matrix equations and checking on convergence of the fission neutron distribution.

In the initial iteration cycles, the  $S_{neut}$  and  $F$  tallies may have significant statistical noise from the MC random walks, and solution of the fission matrix equations may be unreliable. During the initial iteration cycles, the numbers of nonzero entries in the  $S_{neut}$  and  $F$  tallies are monitored for changes from one cycle to the next. Currently, if the number of nonzero entries in the  $S_{neut}$  tallies changes by more than 2% in successive cycles, or if the number of nonzero entries in the  $F$  tallies changes by more than 5% in successive cycles, then the convergence window is shifted. That is, the block of  $M$  cycles is reset and shifted by 1 cycle. The  $S_{neut}$  and  $F$  tallies are declared stable enough for solution only after the stability tests are met for a consecutive block of  $M$  cycles.

The fission matrix equations are solved using a standard, brute-force power iteration method. The method is robust, accommodates the non-symmetric  $F$  matrix, and requires only matrix-vector products, thus preserving the sparsity of  $F$ . Solution of the fission matrix equations yields  $k_{FM}$ ,  $S_{FM}$ , and  $\rho_{FM}$  (the dominance ratio).

## 2.4. Accelerate Convergence of the Fission Source Iterations

If acceleration of the fission source is to be performed, importance sampling weights of  $S_{FM}^{(m)}/S_{neut}^{(m)}$  are determined for each fission site, where  $(m)$  represents the mesh bin containing the fission site. This approach is essentially the same as that used in [12]. The importance sampling is a nonlinear acceleration method that essentially pushes the neutron distribution toward the reference distribution from the fission matrix eigenfunction. For this to be effective, it is essential that the statistical noise in fission matrix elements not be too large. In testing to date, the stability criteria for the  $S_{neut}$  and  $F$  tallies described in

Section 2.3 have been sufficient, and the acceleration method has been stable and effective. While  $\mathbf{S}_{FM}$  is determined only at the end of each block of cycles, the importance sampling weights are updated for each cycle, based on the current  $\mathbf{S}_{neut}$ . Acceleration is performed only during inactive cycles. The acceleration method typically reduces the number of inactive cycles required for convergence by factors of 2-20x.

## 2.5. Perform Statistical Tests for Convergence and Automatically Activate the Tallies

After determining  $\mathbf{S}_{FM}$  at the end of a block of cycles, statistical tests for convergence are performed. Eight tests are performed using metrics computed for cycles in the block, and 3 tests are performed using the distributions  $\mathbf{S}_{neut}$  and  $\mathbf{S}_{FM}$  [13]:

1. The slope of the single-cycle track-length estimates of  $k_{eff}$  for cycles in the block is computed, along with its standard deviation, using a least-squares method. The slope has a Student-T distribution, and if the magnitude of the slope is less than  $t_{0.025} \sigma_{slope}$ , then the slope is not statistically different from zero at the 95% confidence level. In addition, if the magnitude of the slope is less than 0.0001, then it is judged negligible. Thus a slope with magnitude less than  $t_{0.025} \sigma_{slope}$  or 0.0001 is assessed to be consistent with a steady-state converged distribution.
2. The same slope-test procedure is performed for the collision estimates of  $k_{eff}$ .
3. The same slope-test procedure is performed for the absorption estimates of  $k_{eff}$ .
4. The same slope-test procedure is performed for the Shannon entropy of  $\mathbf{S}_{neut}$ . That is,  $H_{neut}$  is computed from  $\mathbf{S}_{neut}$  separately for each cycle in the block, then the slope test procedure is applied to the  $H_{neut}$  values within the block.
5. The same slope-test for  $H_{neut}$  is applied to the marginal distribution in  $x$  of  $\mathbf{S}_{neut}$ .
6. The same slope-test for  $H_{neut}$  is applied to the marginal distribution in  $y$  of  $\mathbf{S}_{neut}$ .
7. The same slope-test for  $H_{neut}$  is applied to the marginal distribution in  $z$  of  $\mathbf{S}_{neut}$ .
8. The  $\mathbf{S}_{neut}$  tallies for each cycle in the block are accumulated and  $H_{block}$  is computed for the cumulative sources in the block.  $H_{FM}$  is then computed for the fission matrix eigenfunction determined at the end of the block. If  $H_{FM}$  and  $H_{block}$  agree within 1%, this test provides strong evidence that the reference fission matrix eigenfunction and the cumulative fission neutron source distribution for the block agree. However,  $H_{FM}$  and  $H_{block}$  may differ if there is significant source renormalization bias in the neutron distribution. Source renormalization bias is independent of the convergence. That is, if too few neutrons per cycle are used in the calculation, the fission neutron source would still converge, but to the wrong, biased solution. ( $\mathbf{S}_{FM}$  from the fission matrix method is not subject to source renormalization bias.) Thus, passing this test provides evidence of both convergence and an adequate neutron population. If the test is not passed, convergence is not precluded, however, since the cause may be an inadequate population size.
9. The Kolmogorov-Smirnov goodness-of-fit test is applied at the 95% confidence level to compare the distributions given by  $\mathbf{S}_{neut}$  and  $\mathbf{S}_{FM}$ . Since these are multidimensional distributions, the test is repeated for many random permutations of the ordering, with the worst-case statistic used to determine the test outcome.
10. The Chi-squared 2-point distribution goodness-of-fit test is performed on  $\mathbf{S}_{neut}$  and  $\mathbf{S}_{FM}$  at the 95% confidence level.
11. The relative entropy (Kullback-Liebler discrepancy) [14] between the distributions for  $\mathbf{S}_{neut}$  and  $\mathbf{S}_{FM}$  is computed. The Kullback-Liebler discrepancy is related to the G-test for goodness-of-fit [15], and is an alternative to the chi-squared test. A 95% confidence level may be determined from the chi-squared distribution. If this test is passed, it provides strong evidence of both fission source convergence and adequate population size. However, as for Test 8, this test may fail if the

population size is too small (leading to source renormalization bias). Thus, passing this test provides evidence of both convergence and an adequate neutron population. If the test is not passed, convergence is not precluded, however, since the cause may be an inadequate population size.

It should be noted that Tests 8 and 11 that involve entropy or relative entropy are sensitive to the neutron population size (number of neutrons per cycle), whereas Test 9 and 10 are not. The underlying causes of the different sensitivities are not known, and are certainly in need of further investigation.

If all of these statistical tests pass (more precisely, if none fail), then convergence is achieved and locked-in for the remainder of the calculation, and active cycles with tallies will begin with the next cycle. To declare convergence, Tests 1-7 and 9-10 are required to pass. Tests 8 and 11 are not required to pass (since they may be affected by source renormalization bias), but provide additional strong evidence of convergence if they are passed.

Due to the statistical nature of the testing, it is likely that some of the convergence tests may not pass in later cycles. Convergence is not rescinded, however. Typically, some tests that occasionally fail after convergence are passed on most subsequent cycles.

On a historical note, it is important to consider the difference between the statistical testing described above and 2 previous attempts at automated convergence testing [16,17]. The previous attempts at automated convergence testing performed statistical slope tests for only 2 quantities,  $k_{eff}$  and  $H$ , for blocks of cycles. While that is essentially what users do in examining plots of  $k_{eff}$  and  $H$  to assess convergence, it was found to be not sufficiently reliable or conclusive for NCS purposes. In the current work, 11 different statistical tests are performed, including additional tests with sensitivity to spatial effects (e.g., the entropy for x-, y-, and z-marginal distributions) and tests for goodness-of-fit for distributions, with the fission matrix eigenfunction used as a reference solution. While the previous work suffered from occasional false positives or false negatives, the current approach using 11 statistical tests has been reliable and robust for all problems tested. In addition, reporting of the results for the statistical tests provides documented evidence of the convergence assessment.

## **2.6. Assess Whether Sufficient Neutrons Per Generation Were Used.**

After convergence, statistical testing continues to be performed at the end of each block of cycles. In addition, 2 novel statistical tests are made to assess population size [18]. These tests are based on comparing the Shannon entropy and relative entropy (Kullback-Leibler discrepancy) of  $S_{neut}$  and  $S_{FM}$ . If these tests indicate that an insufficient number of neutrons/cycle was used, warning messages are issued.

## **3. TESTING**

The automated acceleration and convergence testing methods have been applied to an assortment of criticality problems, including:

- the MCNP *validation\_criticality* suite, containing 31 ICSBEP benchmark problems,
- the MCNP *validation\_crit\_extended* suite, containing 119 ICSBEP benchmark problems,
- a 2D model of a commercial PWR,
- the ATR (advanced test reactor),
- the ACRR burst reactor at Sandia,
- the OECD-NEA Hoogenboom-Martin 3D reactor computer-performance benchmark,
- the 3D C5G7 U-Mox OECD-NEA benchmark problem,
- the ICSBEP benchmark case LEU-COMP-THERM-078 (a Sandia experiment),
- a large 3D storage pool with checkerboard arrangement (OECD-NEA EG on source convergence benchmark #1),



- a 400 cm tall single reactor fuel-pin unit cell with reflecting boundary conditions,
- the Whitesides problem ( $k_{eff}$  of the world),
- a 3D Triga reactor model,
- the OECD-NEA source convergence benchmark #4, *test4s*, and
- the Godiva HEU sphere

For all of these cases, standard *mcnp6* input files were used with ENDF/B-VII.1 nuclear data. The only additional input supplied consisted of commands to activate the fission matrix treatment, automated convergence testing, and fission source acceleration:

```

kopts      fmat=      yes
           fmatconvrg= yes
           fmataccel=  yes

```

Figure 2 shows some results from testing on the *test4s* problem (OECD-NEA Source Convergence benchmark #4). The upper plots of  $k_{eff}$  (for neutrons, fission matrix, and cumulative) and  $H$  (for neutrons and fission matrix) are from a typical traditional calculation, where a short trial run is made to determine the number of inactive cycles based on plots of  $k_{eff}$  and  $H_{neut}$ , and then a final run is made after manually setting the number of inactive cycles in the *mcnp6* input file. The number of inactive cycles is somewhat arbitrary, and typical conservative-minded users would choose 150-200 inactive cycles. The lower plots of  $k_{eff}$  and  $H$  show results from the automated acceleration and convergence methods, with convergence achieved after 31 cycles with no additional user input or trial runs. Figure 3 shows the quantitative evidence reported when the statistical tests for convergence are all passed. Figures 4 and 5 show details of the initial cycles, including the stabilization of the  $S_{neut}$  and  $F$  tallies, blocks of cycles for acceleration and convergence testing, and the automatic start of active cycles.

Regarding robustness and reliability of the automated methods, all of the 162 test problems performed as expected, with no false positives for convergence or false negatives (which would not affect results, but would increase computer time). When run using conventional methods, the 31 problems in the *validation criticality* suite required 108 minutes of run time with *mcnp6* using 12 threads. Using the automated acceleration and convergence, the same suite of 31 problems required only 70 minutes to complete correctly. These and other results demonstrate that the combination of automated methods provides effective acceleration and reduces unnecessary conservatism in the number of inactive cycles. The most difficult problem for convergence is the OECD source convergence benchmark #1, a large fuel storage pool where over 2,000 inactive cycles are required for convergence using standard methods. With the automated methods, that problem converged properly in only 108 cycles, a factor of 20x improvement.

## 4. CONCLUSIONS

The convergence acceleration typically reduces the number of initial inactive cycles by factors of 2-10 times, and the use of 11 statistical checks for convergence has proven to be extremely robust and reliable. In the future, *mcnp6* users will not have to make trial runs, plot  $k_{eff}$  and  $H$  vs cycle, manually edit the input file to adjust the *kcode* card parameters and then rerun the problem. Common user pitfalls and annoyances have been removed by the automated convergence and statistical testing methods, and quantitative evidence of convergence is provided. That quantitative evidence is extremely important in today's regulatory environment – “eye-balling the plots” is difficult to defend, whereas documented evidence of passing 11 statistical tests is clear cut. Further work in progress includes the automated generation of the initial guess for the fission neutron distribution.

## ACKNOWLEDGMENTS

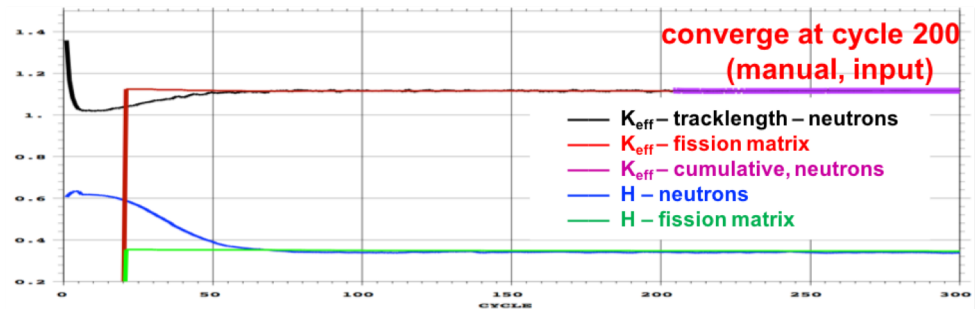
This work was supported by the US DOE-NNSA Nuclear Criticality Safety Program and the US DOE-NNSA Advanced Simulation & Computing Program.

## REFERENCES

*Note: LANL reports cited below are available in the MCNP Reference Collection on the MCNP web site, [mcnp.lanl.gov](http://mcnp.lanl.gov).*

1. W. Goad and R. Johnston, "A Monte Carlo Method for Criticality Problems," *Nucl. Sci. Eng.* **5**, 371-375 (1959).
2. F.B. Brown, "Monte Carlo Techniques for Nuclear Systems", LANL report LA-UR-16-29043 (2016).
3. E. Dumonteil, et al., "Particle Clustering in Monte Carlo Criticality Simulations", *Ann. Nucl. Energy*, **Vol 63**, 612-618 (2014)
4. F.B. Brown, "Investigation of Clustering in MCNP6 Monte Carlo Criticality Calculations", *Int. Conf. on Transport Theory*, Monterey CA, Oct 2017, also LANL report LA-UR-17-29261 (2017).
5. E.M. Gelbard and R.E. Prael, "Monte Carlo Work at Argonne National Laboratory", in *Proc. NEACRP Meeting of a Monte Carlo Study Group*, ANL-75-2, Argonne National Laboratory, Argonne, IL (1974).
6. R.J. Brissenden and A.R. Garlick, "Biases in the Estimation of Keff and Its Error by Monte Carlo Methods," *Ann. Nucl. Energy*, **Vol 13**, No. 2, 63-83 (1986).
7. F.B. Brown, "A Review of Best Practices for Monte Carlo Criticality Calculations", *ANS NCSD 2009*, Hanford WA, also LANL report LA-UR-09-03136 (2009).
8. F.B. Brown, C.J. Josey, S. Henderson, W.R. Martin, "Automated Acceleration and Convergence Testing for Monte Carlo Nuclear Criticality Safety Calculations", *Int. Conf. Nuclear Criticality*, ICNC 2019, Paris FR, Sept. 15-20, 2019 (2019).
9. C.J. Werner, et al., "MCNP6.2 Release Notes", LANL report LA-UR-18-20808 (2018).
10. F.B. Brown, S.E. Carney, B.C. Kiedrowski, W.R. Martin, "Fission Matrix Capability for MCNP, Part I - Theory", *Mathematics & Computation 2013*, Sun Valley, ID, also LANL report LA-UR-13-20429 (2013).
11. S.E. Carney, F.B. Brown, B.C. Kiedrowski, W.R. Martin, "Fission Matrix Capability for MCNP, Part II - Applications", *Mathematics & Computation 2013*, Sun Valley, ID, also LANL report LA-UR-13-20454 (2013).
12. L.L. Carter, N.J. McCormick, "Source Convergence in Monte Carlo Calculations", *Nucl. Sci. Eng.* **36**, 438-441 (1969).
13. F.B. Brown, W.R. Martin, "Statistical Tests for Convergence in Monte Carlo Criticality Calculations", LANL report LA-UR-18-28764 (2018).
14. T.M. Cover, J.A. Thomas, *Elements of Information Theory (2<sup>nd</sup> ed)*, Wiley-Interscience (2006).
15. J.H. McDonald, "Handbook of Biological Statistics (3<sup>rd</sup> ed.)", URL: <http://www.biostathandbook.com>, (2015).
16. B.R. Nease, F.B. Brown, "Convergence Tests for the Fission Source Distribution in MCNP5", LANL report LA-UR-05-6226 (2005).
17. F.B. Brown, B.R. Nease, J. Cheatham, "Convergence Testing for MCNP5 Monte Carlo Eigenvalue Calculations", *Mathematics & Computation 2007*, Monterey CA, also LANL report LA-UR-07-1123 (2013).
18. F.B. Brown, C.J. Josey, "Diagnostics for Undersampling and Clustering in Monte Carlo Criticality Calculations", LANL report LA-UR-18-27656 (2018).

50,000 neut/cycle  
no acceleration  
no auto-converge  
k = 1.1164 (2)



50,000 neut/cycle  
acceleration  
auto-converge  
k = 1.1165 (2)

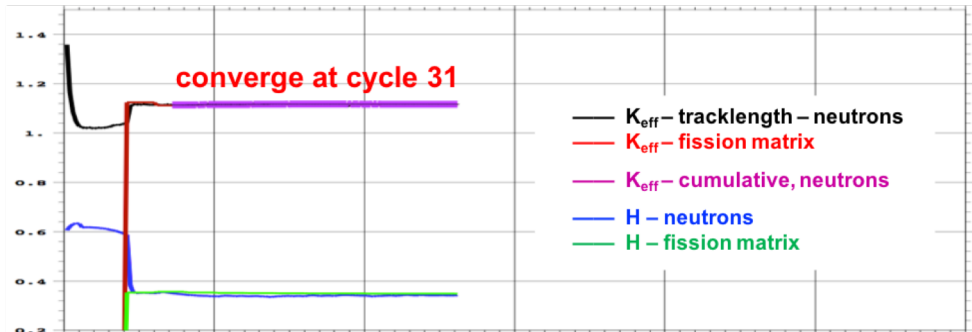


Figure 2.  $K_{eff}$  and  $H$  by cycle for problem *test4s*, without/with automated acceleration & convergence

31 1.11257 1.12 0.35069 680

fmatrix keff= 1.11187, DR= 0.91653, iters= 138

CONVERGENCE INFO & CHECKS: (based on last 10 cycles)

entropy for fmatrix eigenvector = 0.35656  
entropy for neutron last cycle = 0.35069 dif= -1.65%  
relative entropy for last cycle = 0.00972

slope of keff (tracklen)	= 4.2E-03,	target: < 5.1E-03	PASS
slope of keff (collide)	= 4.6E-03,	target: < 4.9E-03	PASS
slope of keff (absorb)	= 4.6E-03,	target: < 4.9E-03	PASS
slope of entropy	= -1.4E-02,	target: < 1.6E-02	PASS
slope of entropy X marginal	= -1.8E-02,	target: < 1.9E-02	PASS
slope of entropy Y marginal	= -1.8E-02,	target: < 1.9E-02	PASS
slope of entropy Z marginal	= 1.3E-03,	target: < 1.6E-03	PASS
entropy dif, neut vs fmat	= -9.1E-04,	target: < 1.0E-02	PASS
Kolmo-Smirnov, distrib, stat	= 2.5E-03,	target: < 9.1E-02	PASS
Chi-square, distrib, stat	= 9.0E+01,	target: < 5.1E+02	PASS
rel-h-block, distrib, stat	= 2.8E-03,	target: < 5.1E-03	PASS

Quantitative  
Evidence  
For  
Convergence

\*\*\*\*\*  
\*\* FISSION SOURCE HAS CONVERGED, based on last 10 cycles \*\*  
\*\* Metrics: \*\*  
\*\* slope of keff (tracklen) is 0 (within uncert) \*\*  
\*\* slope of keff (collide) is 0 (within uncert) \*\*  
\*\* slope of keff (absorb) is 0 (within uncert) \*\*  
\*\* slope of entropy is 0 (within uncert) \*\*  
\*\* slope of entropy X marginal is 0 (within uncert) \*\*  
\*\* slope of entropy Y marginal is 0 (within uncert) \*\*  
\*\* slope of entropy Z marginal is 0 (within uncert) \*\*  
\*\* entropy dif, neut vs fmat is 0 (within uncert) \*\*  
\*\* Distribution checks: \*\*  
\*\* Kolmo-Smirnov, distrib, stat, neut vs fmat (within conf) \*\*  
\*\* Chi-square, distrib, stat, neut vs fmat (within conf) \*\*  
\*\* rel-h-block, distrib, stat, neut vs fmat (within conf) \*\*  
\*\*\*\*\*

Quantitative  
Evidence  
For  
Convergence

Convergence is locked-in, even if some tests fail in future cycles

Active cycles will begin with cycle = 32  
Active cycles will end with cycle = 131  
Total active cycles to be run = 100

Figure 3. Automated convergence reporting for problem *test4s*

cycle	k(col)	ctm	entropy	active	k(col)	std dev	chains	
1	1.35733	0.04	0.60521				35416	
2	1.16857	0.10	0.62080	extend H-mesh to:	36 x 35 x 4		22433	
3	1.08223	0.13	0.63109	extend H-mesh to:	37 x 35 x 4		17100	
4	1.05100	0.17	0.63410	dS= 3%, dF= 34%, shift window extend H-mesh to:	37 x 36 x 4		13800	
5	1.02827	0.21	0.63348	dS= 2%, dF= 19%, shift window extend H-mesh to:	37 x 37 x 4		11529	
6	1.02118	0.25	0.61732	dS= 1%, dF= 14%, shift window extend H-mesh to:	37 x 37 x 5		9997	
7	1.02018	0.29	0.61762	dS= 0%, dF= 10%, shift window			8746	
8	1.02413	0.32	0.61845	dS= 1%, dF= 9%, shift window			7790	
9	1.01974	0.37	0.61766	dS= 0%, dF= 7%, shift window			6974	
10	1.01709	0.43	0.61656	dS= 0%, dF= 7%, shift window			6313	
11	1.02129	0.48	0.61606	dS= 1%, dF= 5%, shift window dS= 1%, dF= 5%, shift window			5815	
12	1.01705	0.53	0.61452				5351	
13	1.02459	0.58	0.61263				4975	
14	1.02193	0.65	0.61214				4640	
15	1.02741	0.70	0.60894				4372	
16	1.03005	0.73	0.60600				4091	
17	1.03266	0.78	0.60435				3852	
18	1.03369	0.83	0.60065				3628	
19	1.03485	0.87	0.59622				3426	
20	1.03631	0.91	0.59177				3245	
21	1.04159	0.96	0.58774				3074	

Source,  
fission matrix,  
& mesh  
stabilization

Block  
of  
cycles

fmatrix keff= 1.12401, DR= 0.91098, iters= 199

Figure 4. Details of initial cycles for problem *test4s*

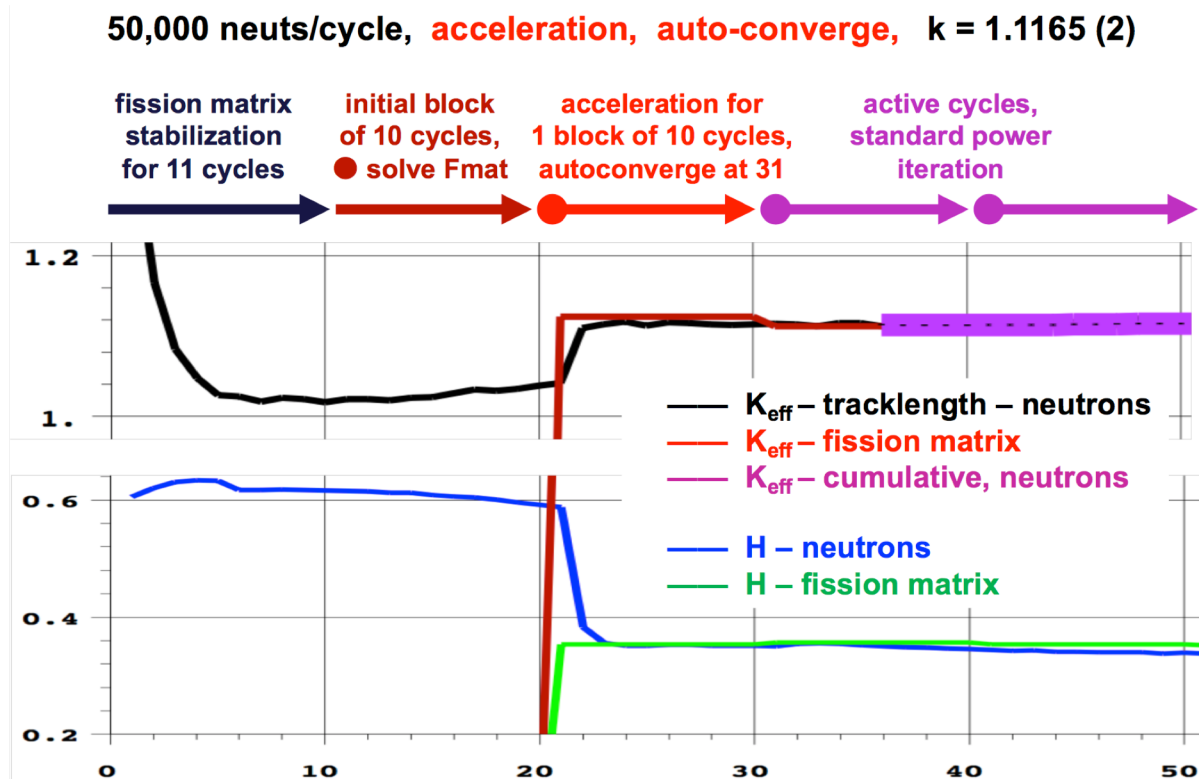


Figure 5. Details of initial cycles for problem *test4s*