ENTROPY-BASED DIAGNOSTICS OF CRITICALITY MONTE CARLO SIMULATION AND HIGHER EIGENMODE ACCELERATION METHODOLOGY

A Thesis
Presented to
The Academic Faculty

by

Shi, Bo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Nuclear and Radiological Engineering in the
School of Mechanical Engineering

Georgia Institute of Technology
August 2010

ENTROPY-BASED DIAGNOSTIC OF CRITICALITY MONTE CARLO SIMULATION AND HIGHER EIGENMODE ACCELERATION METHODOLOGY

Approved by:

Dr. Bojan Petrovic, Advisor
School of Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Farzad Rahnema
School of Nuclear and Radiological Engineering
Georgia Institute of Technology

Dr. Dingkang Zhang
School of Nuclear and Radiological Engineering
Georgia Institute of Technology

Date Approved: May 04, 2010

[This thesis is dedicated to my parents, who are supporting me back in China. It is also dedicated to my wife, Anjie Xie, because of her encouragement.]

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Dr. Bojan Petrovic, for his valuable advice and unparalleled support. I am also grateful to Dr. Farzad Rahnema and Dr. Dingkang Zhang for their kindness to be my master degree committee members and their invaluable suggestions. I gratefully acknowledge the help of Dr. Thomas Booth for the instructive discussions. I would like to recognize the effort that I received from my colleagues, Jordan McKillop, Jeff Ryckman, and David Hartmangruber, for their helpful advice on English or the U.S. life style. Special thanks to my senior fellow apprentices, Bo Lu, Zhan Zhang, and Yidong Yang, for their relentless support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$\psi\left(\vec{r},\hat{\Omega},E,t\right)$	Neutron flux
A	Matrix or operator
ψ	Well-behaved function
ψ_i	The ith eigenfunction
ψ_1	Fundamental eigenfunction
ψ_2	The second eigenfunction
k_i	The ith eigenvalue
k_1	Fundamental eigenvalue
k_2	The second eigenvalue
ψ_i	The ith eigenfunction
ψ_i	The ith eigenfunction
k_{eff}	Effective multiplication factor, same as k_1
s^2	Sample variance
σ^2	Variance
H	Shannon entropy value
μ	Mean value
P	Matrix
DR	Dominance ratio
PDF	Probability distribution function

SUMMARY

Because of the accuracy and ease of implementation, Monte Carlo methodology is widely used in analysis of nuclear systems. The obtained estimate of the multiplication factor (k_{eff}) or flux distribution is statistical by its nature. In criticality simulation of a nuclear critical system, whose basis is the power iteration method, the guessed source distribution initially is generally away from the converged fundamental one. Therefore, it is necessary to ensure that the convergence is achieved before data are accumulated. Discarding a larger amount of initial histories could reduce the risk of contaminating the results by non-converged data but increases the computational expense. This issue is amplified for large loosely coupled nuclear systems with low convergence rate. Since k_{eff} is a generation-based global value, frequently no explicit criterion is applied to the diagnostic of k_{eff} directly. As an alternative, a flux-based entropy check available in MCNP5 works well in many cases. However, when applied to a difficult storage fuel pool benchmark problem, it could not always detect the non-convergence of flux distribution. Preliminary evaluation indicates that it is due to collapsing local information into a single number. This thesis addresses this problem by two new developments. First, it aims to find a more reliable way to assess convergence by analyzing the local flux change. Second, it introduces an approach to simultaneously compute both the first and second eigenmodes. At the same time, by computing these eigenmodes, this approach could increase the convergence rate. Improvement in these two areas could have a significant impact on practicality of Monte Carlo criticality simulations.

CHAPTER 1

INTRODUCTION

1.1 Neutron Transport Equation

The transport of neutrons is described by the neutron transport equation [1]

$$\frac{1}{v}\frac{\partial}{\partial t}\psi\left(\vec{r},\hat{\Omega},E,t\right)+\hat{\Omega}\bullet\vec{\nabla}\psi\left(\vec{r},\hat{\Omega},E,t\right)+\sigma\left(\vec{r},E\right)\psi\left(\vec{r},\hat{\Omega},E,t\right)=q\left(\vec{r},\hat{\Omega},E,t\right), \quad (1.1)$$

where $\psi\left(\vec{r},\hat{\Omega},E,t\right)$ is the neutron flux depending on seven variables: position, direction, energy, and time. The notation $\sigma\left(\vec{r},E\right)$ is the total cross-section of the material and $q\left(\vec{r},\hat{\Omega},E,t\right)$ is the source term including external source, scattering source term, and fission source term

$$q\left(\vec{r},\hat{\Omega},E,t\right)=s(\vec{r},\hat{\Omega},E,t)+\int_0^{+\infty}dE'\int d\Omega\left[\sigma_s\left(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega}\right)+\frac{1}{k}\nu\sigma_f\left(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega}\right)\right]\psi\left(\vec{r},\hat{\Omega}',E',t\right). \quad (1.2a)$$

The fission cross-section in the fission source term is sometimes written as a multiplication of two terms: one represents the cross-section independent on the final energy and the other represents the final energy spectrum $\chi(E)$.

$$q\left(\vec{r},\hat{\Omega},E,t\right)=s(\vec{r},\hat{\Omega},E,t)+\int_0^{+\infty}dE'\int d\Omega\left[\sigma_s\left(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega}\right)+\frac{1}{k}\nu\sigma_f\left(\vec{r},E',\hat{\Omega}'\to\hat{\Omega}\right)\chi(E)\right]\psi\left(\vec{r},\hat{\Omega}',E',t\right) \quad (1.2b)$$

The factor k introduced to balance the equation is called the multiplication factor and usually referred to k_{eff}. Proper boundary conditions and initial conditions should also come along with the transport equation for a specific problem.

1.2 Solutions

The neutron transport equation is essentially an integral-differential equation. Even though the transport equation is difficult to solve, two categories of solving strategies are well developed.

1.2.1 Deterministic Method

The purpose of solving the transport equation deterministically is to reach a global solution using appropriate approximations and discretizations. The approach in the deterministic method is to discretize variables—such as time t, energy E, or position r—and solve the discretized equations numerically. Two well known deterministic methods are spherical harmonics method (known as P_N method) and discrete ordinates method (known as S_N method). The basic details of these methods are available in Ref. 1.

1.2.2 Monte Carlo Method

Different from deterministic methods, which are complex in mathematical derivation, the Monte Carlo method is a straight forward technique that simulates many neutrons over their life time. [2] Two types of problems are widely investigated based on the purpose of the simulation.

1.2.2.1 Fixed source or Shielding problem

In this type of problem, one is interested in a local flux distribution as a consequence of fixed sources. Neutrons are generated according to the source distribution and travel through the system to contribute to tallies at the places of interest. After simulating sufficient number of neutron histories, tally results are obtained with a confidence interval.

1.2.2.2 Criticality or Eigenvalue problem

The other kind of problem is to obtain the multiplication factor (k_{eff}) of a nuclear system to analyze its criticality status. The foundation of solving these problems is an iterative algorithm so called the power iteration method that eliminates the higher mode components in the solution. Generally speaking, an operator A with eigenvalue k_i and corresponding eigenfunction ψ_i satisfies the relation

$$A\psi_i = k_i\psi_i \text{, in which } k_1 > |k_2| > |k_3| > \cdots\cdots. \tag{1.3}$$

Any "well-behaved" function ψ can be expanded with the eigenfunctions

$$\psi = \sum_i a_i\psi_i \,. \tag{1.4}$$

With renormalization and iteration, the well-behaved function will converge to the fundamental eigenfunction by the power iteration method. Estimates of the fundamental eigenfunction ψ_1 and the fundamental eigenvalue k_1 (also known as k_{eff}) are given by

$$\lim_{n\to\infty} \frac{1}{k_1^n} A^n\psi = \psi_1, \text{ and } k_1 = \lim_{n\to\infty} \frac{A^n\psi}{A^{n-1}\psi}. \tag{1.5}$$

The ratio $|k_2|/k_1$ is known as the dominance ratio (DR) that determines the convergence rate for a computation. [3] When DR is close to one, many iterations are needed before the source distribution converges to the fundamental eigenfunction.

As a result of the power iteration method, only the fundamental mode remains, which the source distribution converges to. The source stored for the next iteration, which is also called generation or cycle, is updated by the latest source points produced. After reaching the convergence, k_{eff} results will be gathered to yield an expected value with a confidence interval. Either way, Monte Carlo simulation is a generally-understandable approach to solve the neutron transport equation. However, the simulation does not automatically provide the flux or source distribution globally. In other words, one has to specify the places he is interested in beforehand resulting in additional computational cost.

CHAPTER 2

OBSTACLES IN MONTE CARLO CRITICALITY SIMULATIONS

Monte Carlo criticality simulations have been used in investigation of nuclear systems, to generate reference results and evaluate safety issues. Despite the wide acceptance of the general accuracy of Monte Carlo criticality simulations, this method still faces several obstacles in different aspects. This chapter will review some common known drawbacks and challenges of the Monte Carlo criticality simulations.

2.1 Systematic Challenges

2.1.1 Undersampling

The foundation of the Monte Carlo methodology is statistical or stochastic principle. According to the Law of Large Number and Central Limit Theorem, [4] the more particles a simulation uses, the more accurate the results will be. Unfortunately, because of the costly computational consumption by just increasing the particle number for a simulation in real life, a compromise is needed between the computational expense and accuracy. Either way, the condition requirement is that the reported results remain unbiased. However, if the number of particles is insufficient, the simulation may not adequately represent the property of the entire system, resulting in biased results. This lack of sufficient particles is called undersampling. Some of the previous works discussing undersampling problems include n Refs. 5, 6, and 7.

The criterion of how many particles pre generation are sufficient to sample for a particular problem is not quite clear, but it does depend on the geometry scale and material cross-section properties of the problem. T. Ueki presented a posteriori undersampling checking criterion in Ref. 8 and an on-the-fly check criterion in Ref.9. Both of them are based on the information theory and using entropy correlated value for

the criteria. In addition, E. Larsen and J. Yang proposed a hybrid method, combing deterministic and Monte Carlo methods together, to solve the undersampling problem. [10] They named their method functional Monte Carlo (FMC) method and claimed that FMC could minimize undersampling difficulty for many problems and yield a more accurate estimate of the k_{eff} results.

Undersampling problem has a significant impact for the huge or real life scale system simulation. When one is dealing with such problems, he needs to pay attention to the potential undersampling problem.

2.1.2 Loosely Coupled Systems

Another problem may be identified along with a challenging question: how does a nuclear plant in New York City would affect another one in Paris? A famous statement about this thought was the "k-effective of the world," [11] which states that the k_{eff} of the entire world would tend to be estimated as zero since none of the simulation particles had a sufficient probability to be found in a fuel assembly.

A mathematical expression of the problem recalls the principles of the power iteration method, whose convergence rate depends on the DR in the above section. One possible cause of the high DR is loosely coupled property of the problem. The loosely coupled property refers to that two fission regions with high neutron flux are separated by a material with high absorption or scattering cross-section, such as water or concrete. As a result, one neutron in one fission region has low probability of travelling through the separation gap and reaching the other fission region. Therefore, the coupling of these two regions is loose. This kind of problems is significant especially for criticality safety applications and large reactors.

The OECD (Organization for Economic Co-Operation and Development) Nuclear Energy Agency (NEA) established an Expert Group on Source Convergence in Criticality Safety Analysis in 2000. [12] They generated four test problems:

1. The Checkerboard Fuel Storage Array
2. Pin-cell Array with Irradiated Fuel
3. Loosely Coupled Uranyl Nitrate Solution Slabs
4. Array of Interacting Spheres.

Researchers have used these problems to systematically examine the loosely coupled issue as well as the undersampling problem much better.

2.1.2 Underestimate of the Variance

In addition to the flux, current, or k_{eff} results, the variance of the estimated results should also be determined during a simulation. The contribution from each particle to the tally is just like a random sample in a statistical experiment, [4] which is generated from a specific probability distribution function (PDF). In a fixed source problem, every particle generated independently from the source distribution may contribute to the tally. If N particles, each one of which contributes x_i to the tally, are used in a simulation, the estimated tally will be the average number of x_i [4]

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}. \tag{2.1}$$

The sample variance of samples x_i, denoted as s^2, can be obtained by the following relation

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2 = \frac{1}{N-1}\left[\left(\sum_{i=1}^{N} x_i^2\right) - N * \bar{x}^2\right]. \tag{2.2}$$

For very big N

$$s^2 = \frac{1}{N} * \sum_{i=1}^{N} x_i^2 - \bar{x}^2. \tag{2.3}$$

The sample variance of the average value $\bar{x}$ could be obtained as

$$s_{\bar{x}}^2 = \frac{s^2}{N}. \tag{2.4}$$

The final variance of the average value $s_{\bar{x}}^2$ is proportional to $1/N$. In other works, the standard deviation $s_{\bar{x}}$ is proportional to $1/\sqrt{N}$, which is well known in Monte Carlo simulations and used in Monte Carlo analysis.

The condition for the validity of the variance estimate is that the random samples are from the same PDF, which has been met, and independent from each other. This condition is exactly correct for a fixed source problem because there is no correlation between any two particles. However, when it turns to criticality problem, it is not always true. The most interesting tally in criticality problem is the cycle-by-cycle-based k_{eff}, which is a global value instead of a local value like flux. For each cycle, the starting sources are from the source bank established from the simulation of the last cycle. There is, as a consequence of using the source bank, correlation between the successive cycles in estimating the k_{eff} or flux distribution. This correlation always makes the random difference between successive cycles smaller than it should be if they were truly independent. This correlation will cause the underestimate of the variance.

The most famous methodology to deal with underestimate of the variance was MacMillan's Formula, [13] illustrating a bounding of the true variance instead of estimating the true variance directly. However, the formula involved DR value and lag one auto-correlation coefficient which are not easily acquired. Later, Gelbard and Prael proposed a batching method [14] to overcome the underestimate of variance. In this method, groups of estimates of k_{eff} from several successive cycles were taken as batches to estimate a new variance. The normal way of computing the variance was just the special case of the batch method with only one cycle for each batch. The drawback of this method was the choosing of the optimal batch size: no explicit estimate of the optimal size for a batch to make the difference between the new variance and the true one as small as possible is available.

T. Ueki, T. Mori and M. Nakagawa presented two iterative approaches to express the bias [15] between the true variance and apparent variance (expected value of the reported variance after a simulation). These corrections required computing so called correlation coefficients as intermediate values. These two methods were good estimates of the variance but did need extra computing expense. In the past a few years, some other attempts also appeared. H. J. Shim and C. H. Kim specified so called inter-cycle fission source correlation to estimate the true variance. [16] T. Ueki and B. R. Nease proposed to introduce time series analysis theory into this field and achieved good results. [17] All these endeavors made the computation of the true variance possible and the next desired attempt is the on-the-fly estimate without extra computing expense.

2.2 Convergence Diagnostics

The primary objective of the criticality calculation is to estimate the multiplication factor—k_{eff}. To make sure the reported k_{eff} is statistical accurate, one needs to clear out the contamination of the non-converged k_{eff} results. In other words, one should only gather the cycle-based k_{eff} data after the convergence of the simulation. Although this convergence diagnostic could be referred as a statistical check, it is different from the normal statistical check for the flux or current tally. MCNP5 includes ten statistical checks [18] to assess the statistical quality of the flux or current tally. However, for k_{eff} value, the sample number is not the only concern and the ten checks could not always help to identify the non-convergence of k_{eff}. Even if one could generate a convincing way to identify the convergence of k_{eff}, the results may still be wrong due to the slower convergence of source distribution. [19]. Therefore, convergence diagnostic over source distribution is more reliable, despite its difficulty. Numbers of methods have been proposed so as to overcome this difficulty.

2.2.1 Entropy Check in MCNP5

Unlike the entropy concept in information theory, which represents the capability to carry information, entropy value in criticality simulation uses a single value to represent a source distribution status. The entropy value (which is also known as Shannon entropy) is defined as [20]

$$H(S^B) = -\sum_{i=1}^{B} S^B(i)\log_2(S^B(i)), \tag{2.5}$$

where, H represents the entropy value; B is the number of meshes over the entire system; S^B represents the percentage of source initialized in each mesh; and i is the index of the mesh increasing from 1 to B. After each iteration, the percentage of emerging source in each mesh S^B will be computed and the entropy value will be generated for that iteration. This entropy value is used for source convergence diagnostic in MCNP5, [21] a well known Monte Carlo simulation program produced by Las Alamos National Laboratory. According to the MCNP5 user manual, the following procedure applies to the entropy value:

> *Upon completion of the problem, MCNP will compute the average value of H_{src} for the last half of the active cycles, as well as its (population) standard deviation. MCNP will then report the first cycle found (active or inactive) where H_{src} falls within one standard deviation of its average for the last half of the cycles, along with a recommendation that at least that many cycles should be inactive. Plots of H_{src} vs. cycle should be examined to further verify that the number of inactive cycles is adequate for fission source convergence. (MCNP — A General Monte CarloN-Particle Transport Code, Version 5Volume I: Overview and Theory)*

This entropy check is the only available explicit convergence diagnostic in MCNP5. Many research results have shown the validity of the entropy check, but some drawbacks and limitations still exist:

- It is a posteriori check. One can only run the entropy check after the simulation of the problem. If the simulation does not pass the entropy check, another re-simulation is needed with adjustment of the inactive and active cycle number, which is an extra expense of computing resources. Although some recommendation of the required number of inactive cycles is provided after the entropy check, this estimate may not be very robust. Thus, if one then changes the number of inactive cycles to satisfy the recommendation, the new results is not guaranteed to pass the entropy check. In that case, one has to modify the numbers of inactive and active cycles potentially several times to make sure the simulation could pass the entropy check.
- At first, entropy check gathers the local source distribution information by the percentage of source emerging in each mesh. Then for the summation, it discards the local information and represents the whole system status by only one number. This results in the incapability to distinguish two different source distributions with the same entropy value.

2.2.2 Entropy-related Diagnostics Methods

To overcome the posterior property and turn it to an on-the-fly convergence diagnostic, T. Ueki and F. Brown introduced the concepts of “relative entropy” and “progressive relative entropy”. [20] The “relative” there indicated the entropy computed is depending not only on the current cycle but also on some previous reference cycle. By measuring the distance of the reference cycle and current cycle, some criteria were proposed for the corresponding diagnostic. After that, T. Ueki proposed another method called Wilcoxon Signed Rank method, [22] which was similar to one standard statistical check, to handle the average permuted relative entropy (APR), maximum permuted

relative entropy (MPR), and progressive relative entropy (PRR). However, none of these entropy-related diagnostics are available in the current practically released version of MCNP5.

P. Romano proposed another means to utilize the entropy value. [23] He introduced the stochastic oscillator, which was commonly used in the analysis of financial markets, as the method to handle entropy value. The current oscillator value would be greater or smaller than the previous one; if the entropy value converged, this oscillation would be totally random. Some criteria to test its random behavior were applied after finishing a simulation. Therefore, this method was also a posterior diagnostics method. E. Dumonteil etc. proposed to use another entropy value called Boltzmann entropy, [24] which was a one-order higher approximation of the entropy for the continuous case than the Shannon entropy, to perform convergence diagnostic. Although this Boltzmann entropy was more accurate, it resulted in a larger computational expense than Shannon entropy. In addition, as the particle number was increasing, the difference between these two entropies was decreasing.

2.2.3 Other Diagnostics Methods

M. Wenner and A. Haghighat proposed to use the entropy test and the generalized KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test together to reduce the occurrence of false diagnoses. [25] This KPSS test was a statistical test to compare the residuals of a linear regression of a data series. It was powerful because it could detect both undersampling and stationarity problems. H. J. Shim and C. H. Kim proposed two types of convergence criteria, [26] which were based on the statistics analysis of the local fission source distribution. These criteria included computing a covariance matrix of the noise which was not ease to obtain. However, in general, this method could provide a trustable diagnostic result.

2.3 Convergence Acceleration

Indeed, the simplest way to deal with the convergence diagnostic is to simulate the problem for infinite number of generations, which is clearly impractical, so that the convergence is guaranteed. In practice, if the convergence rate could be improved, the convergence diagnostic is also easier to conduct.

2.3.1 Wielandt's Method

An acceleration method used in deterministic source iteration method, Wielandt's method, was recently introduced to Monte Carlo criticality simulation several years ago. [27] [28] In Wielandt's method, each emitted neutron after a fission reaction was tracked until it "died" or induced another fission reaction. All of the tracking of the neutrons was in one iteration, so the source distribution would spread more in space than in traditional Monte Carlo method. As a result, Wielandt's method would reduce the number of iteration needed before the convergence of the source distribution. At the same time, this method would increase computational expense for each iteration. Thus, the total computational expense for Wielandt's method and traditional Monte Carlo method are approximately on the same order. Most of times, this Wielandt's method even increased the total computational cost. However, the advantages of Wielandt's method, such as reducing the possibility of false convergence, and increasing the figure-of-merit (FOM), made it attractive for further implementations.

2.3.2 Hybrid Methods

Since it is easier to analyze deterministic method, some hybrid methods appeared to combine the strengths of Monte Carlo method and deterministic method. Fission source acceleration method was one of these hybrid methods. [29] [30] This method divided the analyzed system into meshes and used Monte Carlo method to compute the fission matrix to represent the probability of a neutron in one mesh inducing a fission reaction in

another mesh. By multiplying this fission matrix many times, a quasi-converged source distribution was obtained for further simulations. Another hybrid method was called anchoring method. [31] This method decomposed fission source into two parts: traditional Monte Carlo source and "anchoring" source which was updated by a deterministic method. As a result, this method would stabilize the source distribution and reduce iterations needed for convergence.

2.3.3 Smoothed Residual Acceleration Method

Lately, a novel technique called smoothed residual acceleration (SRA) [32] was introduced into Monte Carlo criticality simulation. This method was similar to the fixed-parameter extrapolation method widely used in mathematics. Theoretically, SRA operated as a low-pass filter to reduce higher order eigenmodes and increased convergence rate. Practically, it was easy to implement in Monte Carlo simulation by adjusting sources used for new iteration involving sources from previous iteration. This SRA mechanism must be shut down after reaching the convergence to prevent further contamination of the source distribution by artificial adjustment.

2.4 Summary of the Obstacles

Since overcoming the convergence obstacles summarized above is crucial to Monte Carlo criticality simulation, this thesis will focus on some possible solutions. Chapter three will focus on the comprehension and analysis of the entropy check for a huge subcritical system—fuel storage pool. This system is loosely coupled as well as exhibiting the undersampling problem, causing the difficulty in convergence and flux distribution while simulating it. Investigation of the behavior of the entropy check will try to bring up an alternative way for the convergence diagnostics based on the local flux distribution. In addition, chapter four will introduce a modified power iteration method, which generates first two eigenvalues and eigenfunction at the same time, in order to

increase the convergence rate. Refinements to stabilize the iterations will be explained and analysis of the behavior of the variance will also follow in chapter four.

CHAPTER 3

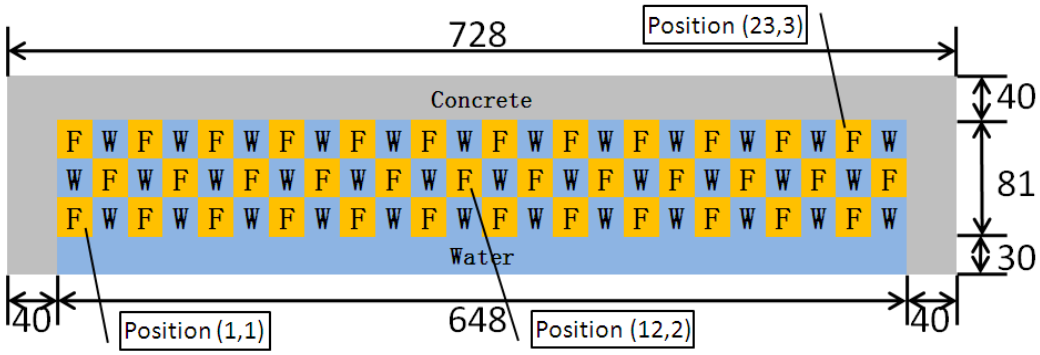
CONVERGENCE OF THE OECD/NEA BENCHMARK PROBLEM NUMBER ONE

To facilitate systematic analysis of the difficulties in the Monte Carlo criticality simulation, this chapter uses the OECD/NEA benchmark problem number one [33] [34] as an illustration. This problem is widely employed in the convergence analysis and the inspection of new methodologies of the convergence diagnostic because of its properties. The main feature of the problem is the large scale (hundreds of the neutron mean free path) which will preclude effective communication between the neutrons on opposite sides. In addition, the water holes between the fuel assemblies reduce the communication between the checkboard placed assemblies. These two reasons cause that the system is loosely coupled. In the meantime, the large scale of the system also yields the undersampling problem since simulating sufficient number of particles to cover all the space is extremely expensive for practical analysis. Therefore, the convergence analysis of this benchmark problem is indeed challenging.

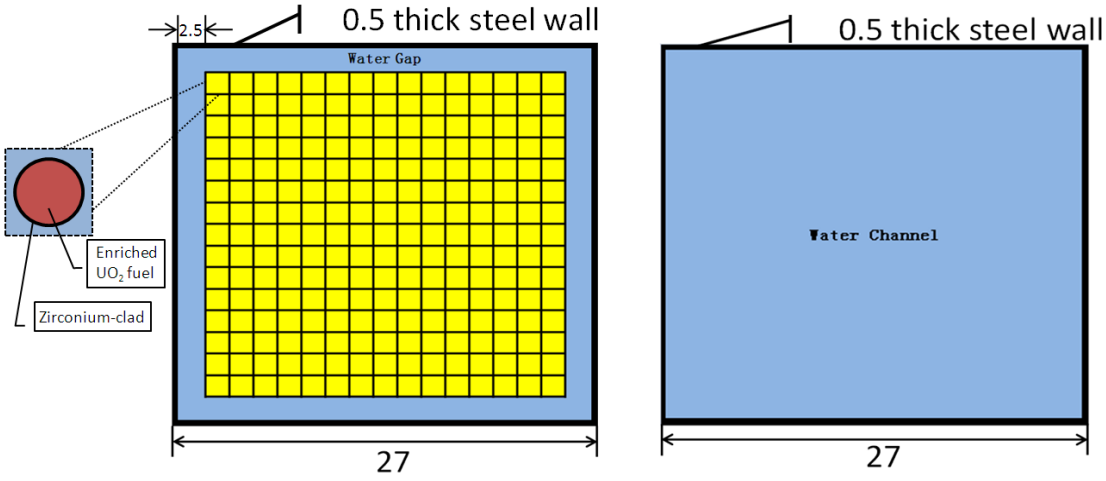
3.1 Benchmark Description

This benchmark problem is also known as a checkerboard fuel storage array or a fuel storage pool with a 24-by-3 array. Figure 3.1 shows the geometric structure of the benchmark problem. The system contains 36 fuel elements with uranium enriched to around 5.0% by weight. The lower left corner fuel assembly is assigned as position (1,1), with two indexes representing right and up directions. Therefore, all the other assemblies are represented by the same position format. For example, the right up corner assembly is represented as position (24,3). Besides these assemblies, three walls outside the array are concrete and the fourth wall is water. The same water also fills top, bottom, and all the

gaps between fuel assemblies. The fuel assembly is composed of 1.44cm-pitch, 15-by-15 lattice. In the center of the fuel rod is a 0.44 cm-radius UO_2 fuel pin with 0.05cm-thick Zirconium-clad. Both the fuel and water assemblies have 0.5cm steel walls outside.

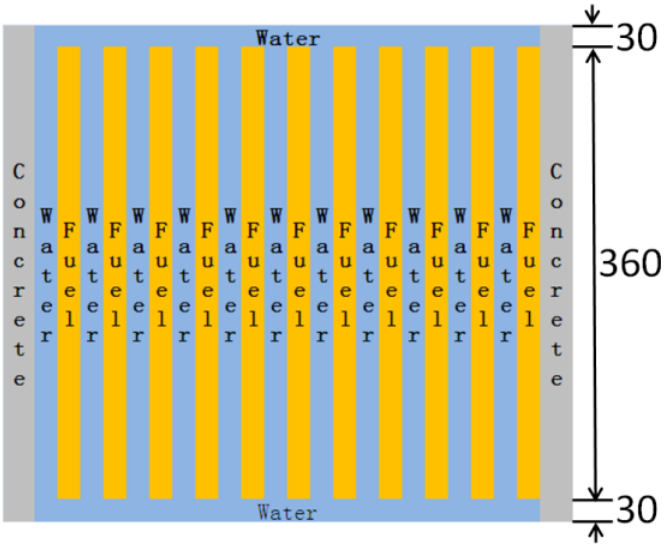


(a) Vertical view of the array.



(b) Fuel and water element and fuel pin.

Figure 3.1. Geometry specification of the benchmark problem



(c) Front view from the center of the middle row with water in the top and bottom.

Figure 3.1 continued. Geometry specification of the benchmark problem

The definition of the benchmark problem specifies three parameters to be used in simulations:

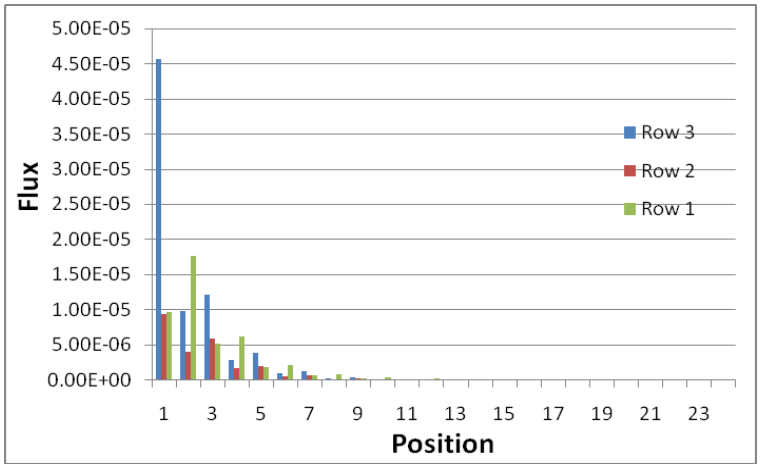
- Number of skipped generations: 20, 40, and 100.
- Number of source point used per generation: 1,000, 2,000, and 5,000
- Distribution of the initial source: all sources in position (1,1), all sources in position (12,2), all sources in position (23,3), and uniform sources over all 36 fuel assemblies.

All possible combinations of these parameters could produce 36 different cases. The results of these 36 cases reported by different research groups with different Monte Carlo program are available in Refs. 33 and 34. This chapter will not review or achieve all these results, but the following simulations used some of the combinations of the parameters.

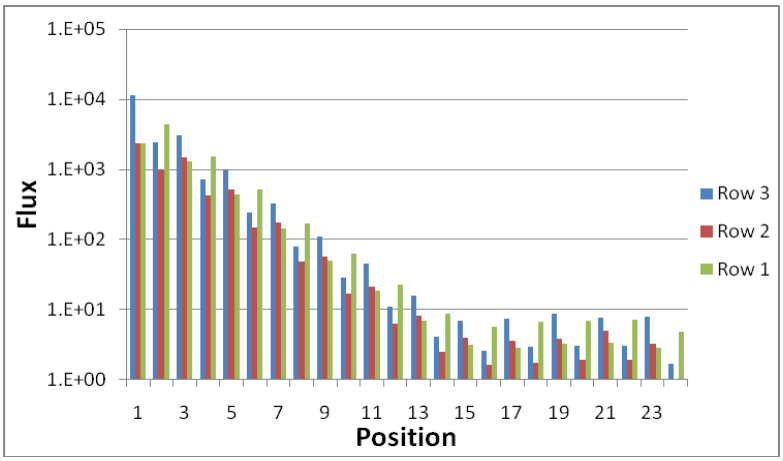
3.2 Basic Results

Before simulating different cases with different parameters, a reference k_{eff} results and flux distribution would be helpful. The problem was simulated using 1,000,000 (10^6)

particles per generation, 1,000 inactive generations, and 1,000 active generations. The initial source distribution was different from the four possible choices above. Since concrete had stronger ability of reflecting neutrons than water, the assembly in position (1,3), which was surrounded by concrete on two sides, having a flux peak is predictable. Thus, a biased initial source distribution was used here: the source points were divided into 55 sets evenly; 20 of 55 sets were generated in the position (1,3) assembly; one set was generated in each other assembly uniformly. By doing this, the initial source distribution had a peak in position (1,3), and this distribution was more closed to the converged distribution than the other four possible choices. To obtain the flux distribution, the mesh tally was used with each assembly as a mesh. Figure 3.2 illustrates the reference flux distribution for all the assemblies and water channels in both the linear and the log scale. This distribution confirmed the prediction that a flux peak existed in position (1,3). The biggest difference of the flux was about 4 orders of magnitude in logarithm scale, which results in the unnoticeable values in arbitrary unit scale. The final k_{eff} reported was 0.88580 ＋/－ 0.00002 for this reference simulation. This reference keff is just in the middle of the reported value in Refs. 33 and 34 from 0.8538~0.8960 with ~0.0007 as the confidence interval.



(a) Flux in arbitary unit



(b) Relative flux compared to the smallest one in position (24,2)

Figure 3.2. Reference flux distributions of the benchmark problem

After this reference simulation, simulations using the parameters specified in the definition were performed. To make the results more reasonable, 100 skipped generations, 500 active generations, and 5,000 source neutrons per generation were used for the four different initial source distributions. Table 3.1 summaries all the k_{eff} results and computing times.

Table 3.1. Input and output summaries for cases 1-4

Case	Source	Computing time	keff±σ
1	Uniform in all fuel regions	43.81 min	0.88386 ±0.00048
2	Uniform in (1,1)	43.51 min	0.88537 ±0.00051
3	Uniform in (12,2)	44.01 min	0.88310 ±0.00047
4	Uniform in (23,3)	43.53 min	0.88419 ±0.00050

By examining the k_{eff} results in Table 3.1, a user cannot tell which result was better converged, i.e., more accurate. In addition to the final reported results, the evolution of the combined average k_{eff} versus the generation number (after skipping the initial one hundred inactive cycles) is depicted in Figure 3.3.

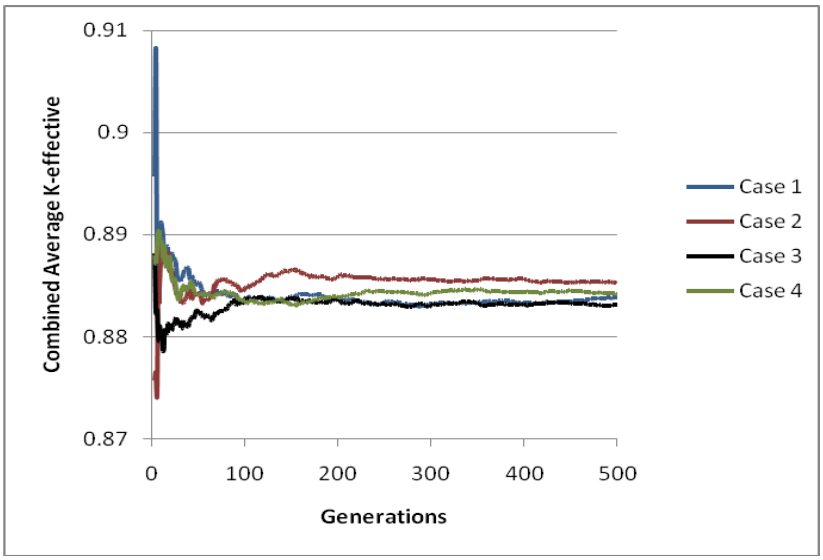


Figure 3.3. K_{eff} versus generation for cases 1-4

The difference between each two runs, expressed by subtraction and combined standard deviation, [3] is given in Table 3.2.

Table 3.2. Standard deviation of the difference between each case for cases 1-4

	Case 1	Case 2	Case 3	Case 4
Case 2	0.00151 ±0.00070 (2.2σ)	X	X	X
Case 3	0.00076 ±0.00067 (1.1σ)	0.00227 ±0.00069 (3.3σ)	X	X
Case 4	0.00033 ±0.00069 (0.5σ)	0.00118 ±0.00071 (1.7σ)	0.00109 ±0.00069 (1.6σ)	X

In most cases, the difference of k_{eff} exceeded 1σ confidence interval. This difference indicated that the k_{eff} results from different cases had not converged yet after the inactive generations. This statistical inconsistency of k_{eff} could be explained by non-convergence of the fission source distribution. The difference between the biggest and smallest k_{eff} in Table 3.2 was ~0.00200. This difference was much larger than the estimated individual σ of ~0.00050 in Table 3.2 or the combined one of ~0.00070 in Table 3.2. Thus, the results from different cases were statistically inconsistent. However, due to the fact that all assemblies were identical, the only problem characteristic causing the diversity of k_{eff} is the boundary condition outside each assembly. As a result, the k_{eff} results were not too far away from each other even if the source distribution has not converged. This small difference in k_{eff} indeed could mislead users to believe that all the results were statistical accurate.

3.3 Entropy Check and its Limitations

Entropy check in MCNP5 may be a powerful tool for convergence diagnostics, [21] [35] when used cautiously and understanding its limitations. In the above simulations, every assembly was defined as one entropy mesh, and half of the meshes (water channels) contributed zero to the entropy value because they didn't contain any fission sources. Table 3.3 summarizes the output results of the entropy check after completing MCNP5 runs. None of the four cases passed the entropy check.

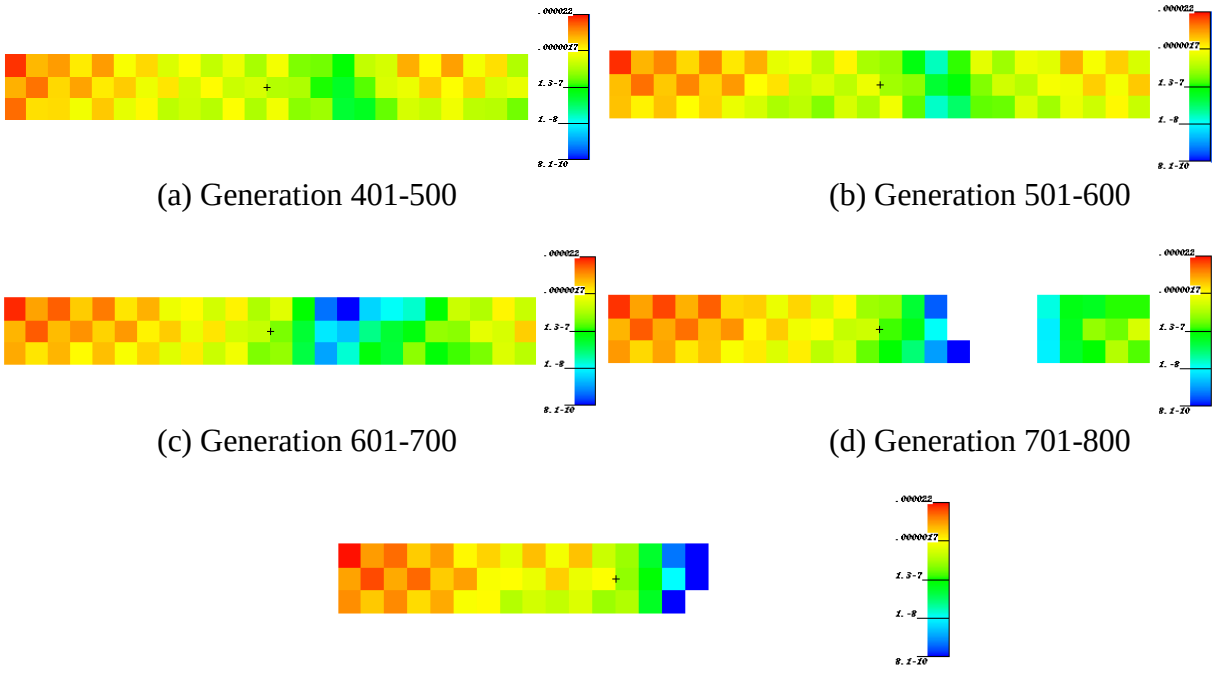
Table 3.3. Results of the entropy check for cases 1-4

Case	First cycle within 1σ with rerun suggestion
1	370, rerun
2	102, rerun
3	141, rerun
4	308, rerun

To understand how this entropy check and rerun suggestion affects users, consider what a typical user, assigned to analyze the k_{eff} and flux distribution of this problem, would do. He would likely place the initial sources uniformly over all the fuel assemblies, and keep all the other parameters the same as in case 1. After running MCNP5, the entropy check suggested him to simulate the problem again with at least 370 generations discarded. He followed the instruction and discarded 400 generations to ensure the convergence before accumulating data from the next 500 active generations. This time, the message (this case was not included in Tables 3.1 or 3.3) showed that the 393rd cycle was the first one having the entropy value within 1σ of the reference value. He would therefore think that the result from this simulation was accurate for further investigation.

However, was this result really accurate or were the 400 inactive generations sufficient for the convergence? To answer this, we used the mesh tally with each

assembly as a mesh to obtain the radial flux distribution for each successive 100 generations. Figure 3.4 depicts the mesh tally results; note that the zero flux regions are shown blank. One could easily conclude that most of the meshes had not converged based on the changing of the colors. Two reasons would explain this non-convergence of flux distribution. The first reason was the undersampling effect in this problem. [8] Using 5,000 particles per generation was certainly insufficient for this giant system. The other reason was the loose coupling property of this problem. The water channels isolated the fuel assembly from each other. Either way, 400 inactive generations were clearly insufficient. This conclusion was further supported by focusing on the flux density change over each 100 successive generations of the high flux element in position (1,1), shown in Table 3.4. The flux value changed dramatically compared with the relatively small estimated σ. This change thus demonstrated that the result is still not converged.



(a) Generation 401-500

(b) Generation 501-600

(c) Generation 601-700

(d) Generation 701-800

(e) Generation 801-900

Figure 3.4. Radial flux distribution evolution for each 100 successive generations

Table 3.4. Flux of position (1,1) for each successive 100 generations

Generations	401-500	501-600	601-700	701-800
Flux (arbitrary units)	9.8960	5.1271	3.9084	6.1409
Estimated σ	0.0489	0.0359	0.0315	0.0390

The observation above exemplified the failure of the entropy check if one was interested in the flux distribution. Passing the entropy check cannot guarantee the convergence of the flux distribution. Moreover, satisfying the entropy check was indeed a necessary but not sufficient condition. In other words, if a simulation did not pass the entropy check, the results were certainly questionable regarding the flux distribution. As a result, the k_{eff} results from a non-converged flux distribution were likely not accurate.

To further examine the utility of the entropy, Figure 3.5 depicts the entropy versus the generation number for the four cases in Table 3.1. While each case by itself seemed to flatten out in the right half, examining them all together against each other clearly showed that the convergence was still far away.

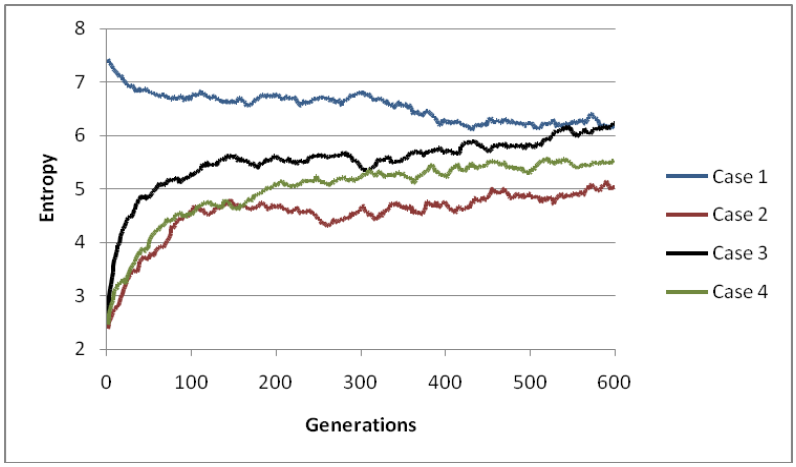


Figure 3.5. Entropy versus generation for cases 1-4

3.4 Bounding Approach

One simple means to estimate the k_{eff} convergence is to use the bounding approach suggested in Refs. 19 and 36. In this approach, one simulation with the initial sources in the highest reactivity region is performed first. Then, another set of the initial sources in the lowest reactivity region initializes the simulation again. Plot of these two the combined average k_{eff} values could generate a bounding curve for the actual k_{eff}. If these upper and lower bounds of k_{eff} merged together after certain number of several generations, it was likely that a simulation with uniform sources would converge as well after that many generations.

Likewise, this bounding approach could also be applied to the entropy value to improve the false convergence detection based on the property of the entropy: the entropy value will reach the minimum if all the initial sources are in one entropy mesh; and reach the maximum for a uniform source distribution over all entropy meshes. [35], [7] To illustrate this property, a uniform source distribution over all the fuel assemblies was used to form the upper bound (case 5); and a point source in the center of position (1,1) was used to form the lower bound (case 7). The third case (case 6) used a biased initial source distribution: 90% sources were generated in position (1,3), and the other 10% sources were generated in the remaining fuel assemblies evenly. This biased source distribution was closer to the converged flux distribution, which was explained in the reference case. Therefore, the entropy value from the biased source distribution should be between the bounding curves. All of the three simulations used 5,000 particles per cycle, 0 inactive cycle, and 500 total cycles. The results in Figure 3.6 confirmed the expected behavior of the bounding approach.

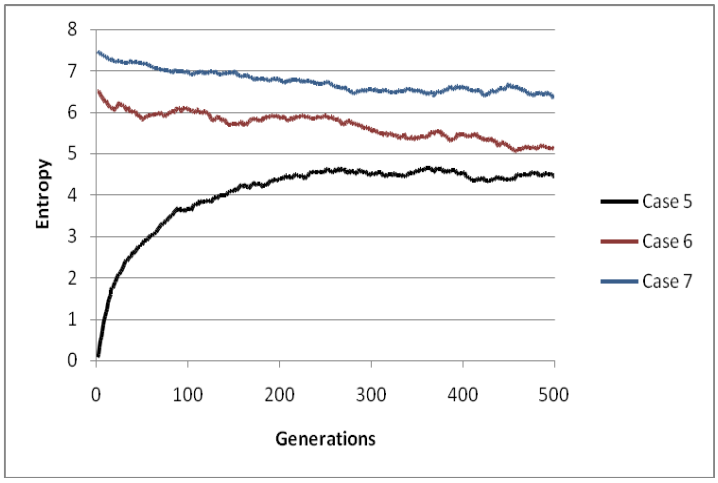


Figure 3.6. Bounding approach for the entropies for cases 5-7

If the flux distribution rather than the global k_{eff} was of interest, the earlier parameters used were insufficient (in terms of generation number and history number) to achieve an accurate result. From Figure 3.6, one could tell that there was a significant difference between each case at the end of the simulation, which indicated the non-convergence of the source distribution. Therefore, more generations were necessary to the convergence of an accurate flux distribution. Reexamining Figure 3.5 with the view of the bounding approach showed that the curves from cases 1 and 3 were merging together at the end of the simulations. If these two cases were chosen to form the bounding of the entropy values, one may reach the conclusion that the source distributions are converged already for cases 1 and 3. However, the analysis in the previous section indicated the non-convergence of case 1. Therefore, bounding approach is not always reliable. It will be valid only if a proper bounding—in other words, bounding initial source distributions—may be established. For simple test problems, this may be feasible, but for complex real-life problem, it may be difficult to guarantee the initial bounding source distributions.

In addition to this limitation, it was still not practical because of its inacceptable computational expense. In MCNP5, the only way to generate a reference entropy value is

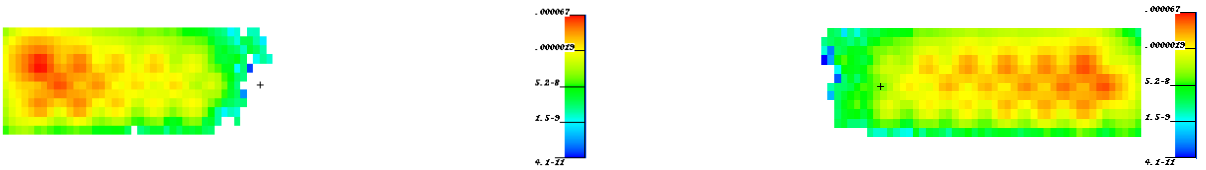
to compute the average value with standard derivation of the last half of the entropy values. Table 3.5 lists the reference values for the above three cases and the difference between each pair of cases.

Table 3.5. Average values of the entropy (2nd half) and the difference between each case for cases 5-7

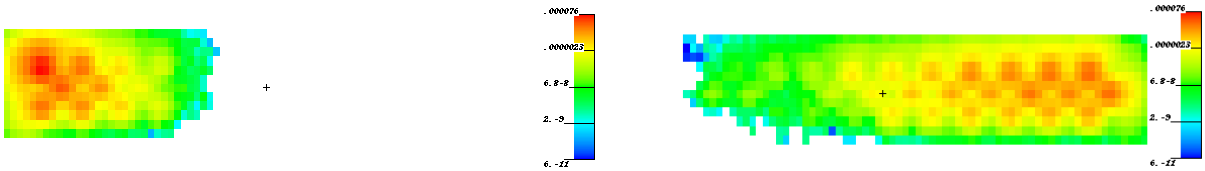
Case	Average values of entropy (2nd half) with estimated σ	Difference	
		Case 6	Case 7
5	4.50961 ±0.00516	0.91114 ±0.01417 64.3σ	2.01757 ±0.00668 302.0σ
6	5.42075 ±0.01320	X	1.10643 ±0.01386 79.8σ
7	6.52718 ±0.00424	X	X

One may observe that the slow convergence of the source distribution caused the bigger gap (~100σ) between entropy curves for different initial sources than that (~3σ) for the k_{eff} curves after 500 generations. To further investigate the evolution of the flux distribution, two new cases with 10,000 particles per generation were simulated with the other parameters remaining the same. The mesh tally was accumulated every 200 generations to compare the flux for the initial source in position (1,1) and in position (23,3). Figure 3.7 shows the flux distribution in the axially middle section (slice) for cycles 801-1,000, 1,801-2,000, and 2,801-3,000. Indeed, the flux distribution was a good proxy for the fission source distribution, so these two distributions would converge in essentially the same way. Analyzing Figure 3.7 clearly indicates that even after 3,000 generations, the flux distributions from the two selected different initial source

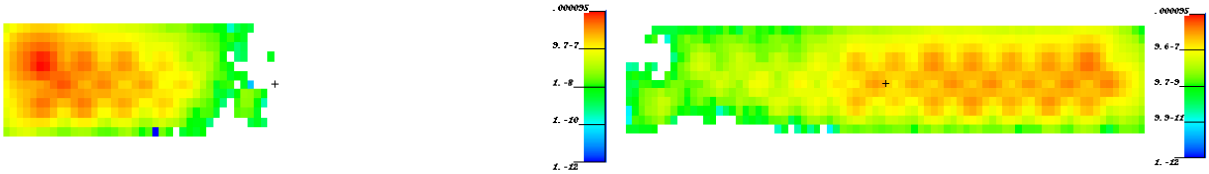
distributions were far away from each other. The simulation initialized with the sources in position (1,1) had the flux only in the left half of the system because of the high reactivity in the left corner. For the other case, the flux distribution from the initial sources in position (23,3) was gradually propagating left for the same reason. However, the rate of spreading was so slow that the flux distribution just reached the left boundary after 3,000 generations. To tell how many more generations were needed for the distribution to converge was difficult.



(a) Generation 800-1,000



(b) Generation 1,800-2,000



(c) Generation 2,800-3,000

Figure 3.7. Comparison of mesh tallies from position (1,1) source(left) and position (23,3) source(right), with blank indicating zero flux

Figure 3.8 depicts the entropy changing curves for the above two cases within the specific generations listed above. The 800-1,000 and 1,800-2,000 line for both cases had some increasing trend, but the 2,800-3,000 curves were almost oscillating flatly. If a user just paid attention to one case of the two, and when the simulation reached 3,000

generations, he would probably conclude that the flux distribution had converged based on the entropy curve. In contrast, if he examined the difference between line (a) and line (b) in Figure 3.8, which was essentially the bounding of the entropy, he would realize that (at least for one of the two) the flux and source distributions were still away from convergence.

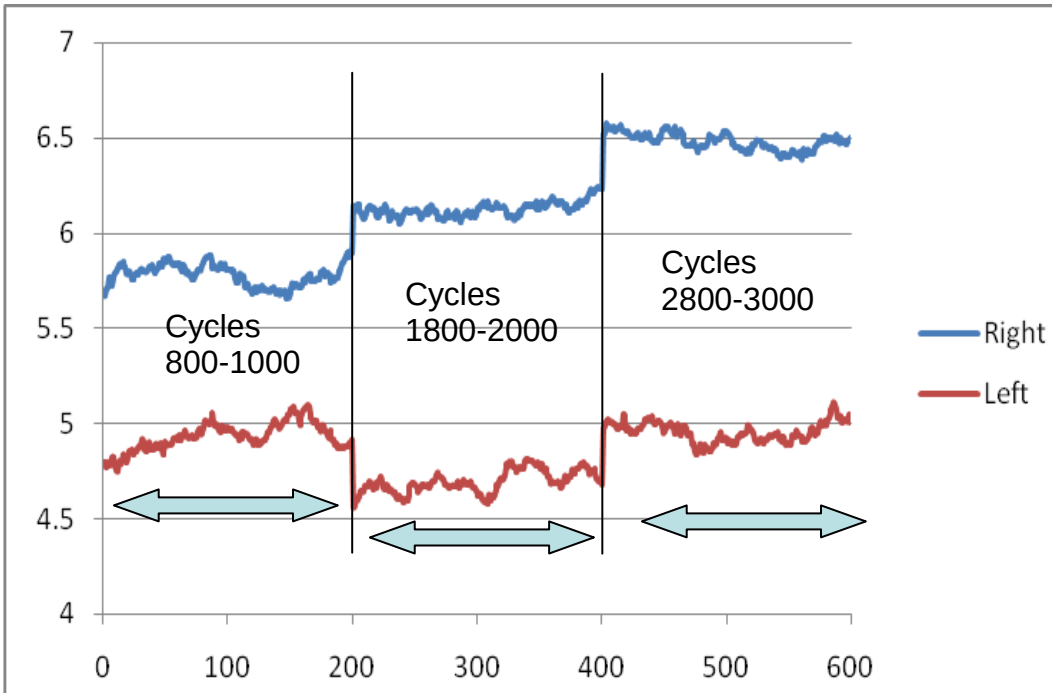


Figure 3.8. Entropy plot of generation 800-1000, 1800-2000 and 2800-3000 for two different initial source distributions:

(a) Red--initial sources at position (1,1) and (b) Blue--initial sources at position (23,3)

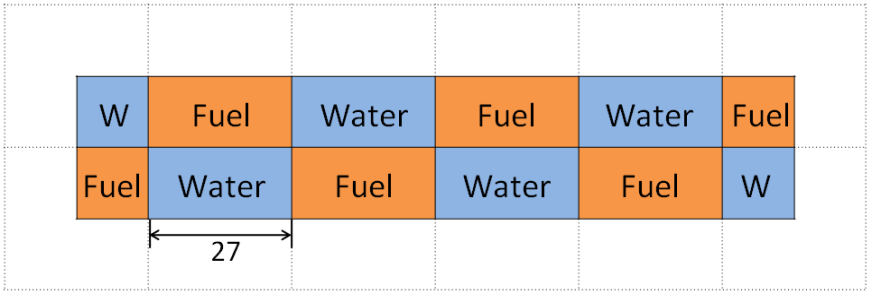
3.5 Statistical Check on the Flux Distribution

For this problem, examining the entropy curve or applying the entropy check is not reliable enough, while visual inspecting the plot of the flux distribution using the mesh tally indicates the non-convergence. However, visual inspection of the flux distribution is somewhat arbitrary and requires an experienced user and preliminary

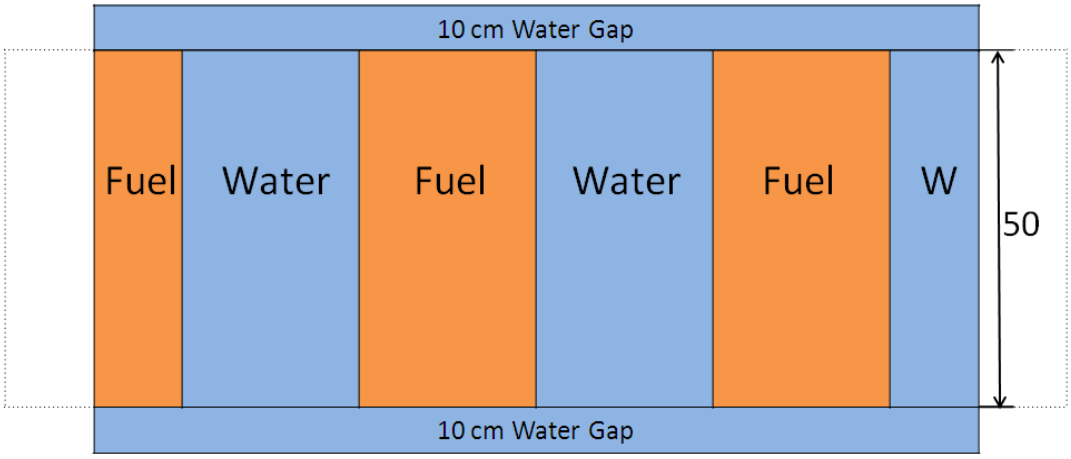
prediction of the results, which is impractical. Therefore, applying some quantitative automatic statistical check mechanisms to the analysis of the mesh tally is highly desirable. A statistics-based method was proposed in Ref. 26 by analyzing the number of source points in each mesh, which was not available in MCNP5, after each generation. Instead, the following investigation was based on the average flux value with standard deviation in each mesh. The first advantage of using the flux instead of the source number was that the flux value was available in the regions composed of water, clad, and concrete. These extra data could increase the accuracy of the statistical check. Another advantage was the availability of the standard deviation, which could result in a better estimate of the statistical behavior.

To accomplish the objective of proposing a robust, automatic statistical check, understanding the behavior of the flux spreading toward convergence using the mesh tally was the first step. To make the simulation easier for analysis, a simplified problem based on the storage fuel pool benchmark was used. Figure 3.9 shows the geometry structure of this simplified problem. This problem was an inner part of a five-by-two sub-array of fuel and water assemblies with four reflective boundaries in radial direction. All the assemblies had the same composition and cross-section properties as the original benchmark. The length of the assemblies was reduced to 50 cm with 10 cm water gap in the top and bottom. Vacuum boundary conditions were applied for the top and bottom boundaries.

While facilitating simulation of the problem, the simplified problem still preserved some characteristics of the original benchmark problem. The loosely coupled property due to the checkboard water gap between fuel assemblies remained, but the difficulty of the communicating over the entire system was reduced. Thus, the total impact of the loosely coupled property was lessened. Likewise, the simplified problem reduced the effects of the undersampling.



a) Vertical view

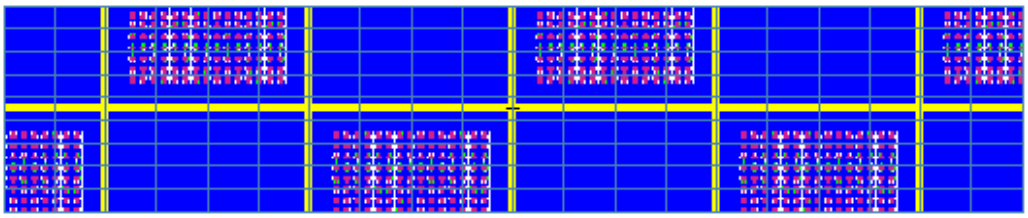


b) Front view

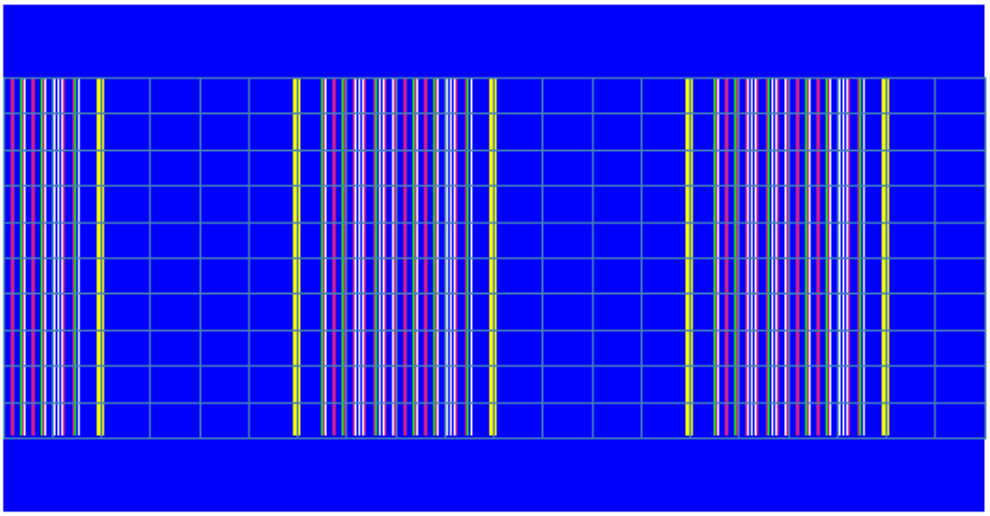
Figure 3.9. Geometry specification of the simplified problem

The simulation used 20,000 particles per generation and 100 generations for each run without inactive cycles. Evenly distributed initial sources over all the fuel and water assemblies were used for the simulation. Although putting the source points in the water channel was inconsistent with the physics of the problem, it initialized the flux distribution in water channel. This source distribution was close to the predicted converged flux distribution—the same flux for the relative same position in the same material. Therefore, the simulation did not need many generations to converge the flux distribution. In addition, to obtain the flux distribution by the mesh tally, 1,800 meshes (mesh size is 6.75cm*3cm*5cm) over the fuel and water assemblies were used as shown

in Figure 3.10. After simulating 100 generations, the random number seed was changed and the previous source file was used as initial sources for the next 100 generation simulation. This procedure enabled the continuous simulations and made the analysis of the successive flux much easier. The simulation of 100 generations was repeated ten times, increasing the total number of generations to 1,000. The first 500 generations were discarded in order to perform diagnostics of the convergence after 500 generations.



a) Vertical view



b) Front view

Figure 3.10. Mesh specification of the simplified problem

After acquiring the five sets of mesh tally data, the first step was to compare the reported flux value for each mesh. These reported flux values were essentially random samples of a random variable representing the converged steady-state flux. If X is a

random variable for a specific mesh, x_1 and x_2 were the sampled values of X along with estimated variances σ_1 and σ_2. Once the simulation converged, the difference between the two samples, x_1-x_2, would follow a specific zero-mean normal probability distribution function [3] with some variance σ

$$(x_1 - x_2) \sim N(0,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{x^2}{2\sigma^2}) . \quad (3.1)$$

Practically, the variance σ was unknown for a simulation, so the sample variance (represented by $\hat{\sigma}$) was always an good approximation of the variance σ. Therefore, normalization would yield a variable following standard (0,1) normal distribution

$$\frac{x_1 - x_2}{\hat{\sigma}} \sim N(0,1) . \quad (3.2)$$

Since σ_1 and σ_2, which were also approximations using sample variances, were the reported variances of x_1 and x_2, a possible variance estimate of the subtraction value x_1-x_2 following the variance combination rules is:

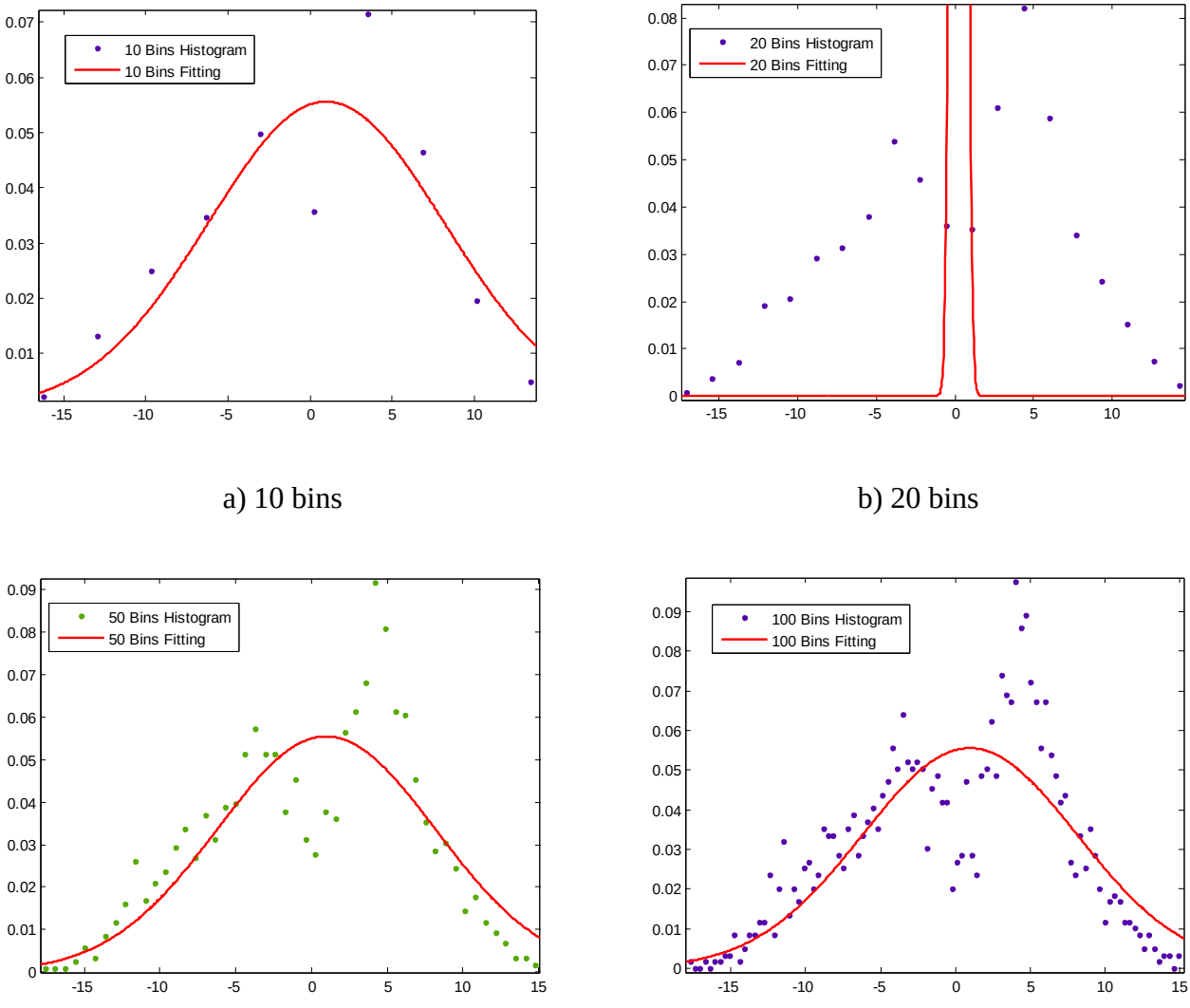
$$\hat{\sigma} = \sqrt{\sigma_1^2 + \sigma_2^2} . \quad (3.3)$$

Based on the above analysis, the normalized subtraction of the successive flux value for a specific mesh was a sample from the standard normal distribution. Therefore, a set of the normalized subtraction for all the meshes from two successive 100 generations would shape a histogram approximating a standard Gauss curve. As an illustration, the mesh tallies obtained from 501-600 generations and 601-700 generations were used to shape histograms. The histograms were characterized with different bin numbers as 10, 20, 50, and 100 for the 1,800 values from meshes. No matter what the histogram looked like,

$$\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2}) \quad (3.4)$$

was used to fit the histogram since the choice of the fitting equation was based on the assumption that the sample points followed the normal distribution. the curve fitting

toolbox in MATLAB was used to obtain the fitting coefficients for the mean value μ and the variance σ. Figure 3.11 shows the histograms and fitting curves for the four cases with different bin numbers. The y-axis represents the relative averaged probability for each bin.



a) 10 bins

b) 20 bins

c) 50 bins

d) 100 bins

Figure 3.11. Probability histograms and normal distribution fitting curves for the normalized subtraction between generation 501-600 and 601-700

Table 3.6. Fitting coefficients of one normal distribution with 95% confidence interval for the difference between generation 501-600 and 601-700

Coefficient	10 bins	50 bins	100 bins
μ	0.9916 (-1.382, 3.366)	0.9815 (-0.06596, 2.029)	0.9725 (0.1936, 1.751)
σ	7.174 (5.201, 9.147)	7.187 (6.317, 8.058)	7.174 (6.527, 7.822)

For some reason, the fitting of 20 bins did not produce meaningful results. Thus, Table 3.6 just shows the fitting coefficients with 95% confidence interval for the other three cases. Both of the two coefficients, μ and σ, converged to some value, and the 95% confidence intervals were shrinking because less information about the sample distribution was lost when using more bins for the histogram. Although it seemed that more bins reduced the confidence intervals for the estimates of the coefficients, the small counts in each bin would also reduce the confidence on the acquired data since they were statistical as well. Thus, an optimal bin number may be available after further exploration. For now, the number of bins was chosen as 50 as a standard parameter for the further investigation.

The estimated mean value μ was 0.9815, which at first glance is not too far away from the expected value zero. In contrast, the estimated variance σ was 7.187, which was tremendously far from the expected value one. This inconsistence could be understood for two reasons. One reason was underestimate of variance due to the auto-correlation effect. This effect caused the estimated variance $\hat{\sigma}$ smaller than the expected one in Eq. (3.2), so the normal distribution used as a fitting target had a larger variance than one. Despite this underestimate, the fitting variance σ should not be ~7 times bigger than expected. The other possible reason was the loosely coupled property of this problem.

This property would cause the isolation of the fuel assemblies for several generations. In this case, due to flux re-distribution, some meshes had a trend of increasing flux and some other meshes had a trend of decreasing flux. These two groups of meshes together induced the large fitting σ. Indeed, if one observed the histogram plot in Figure 3.11 carefully, two or more potential peaks, representing the two groups of meshes, were clear. Therefore, a summation of two normal distributions as

$$\frac{1}{\sqrt{2\pi}\sigma_1} * \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^{\ 2}}\right) + \frac{1}{\sqrt{2\pi}\sigma_2} * \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^{\ 2}}\right) \tag{3.5}$$

may be a better option for the curve fitting. Figure 3.12 shows the summation fitting curve, and Table 3.7 shows the corresponding estimated coefficients.

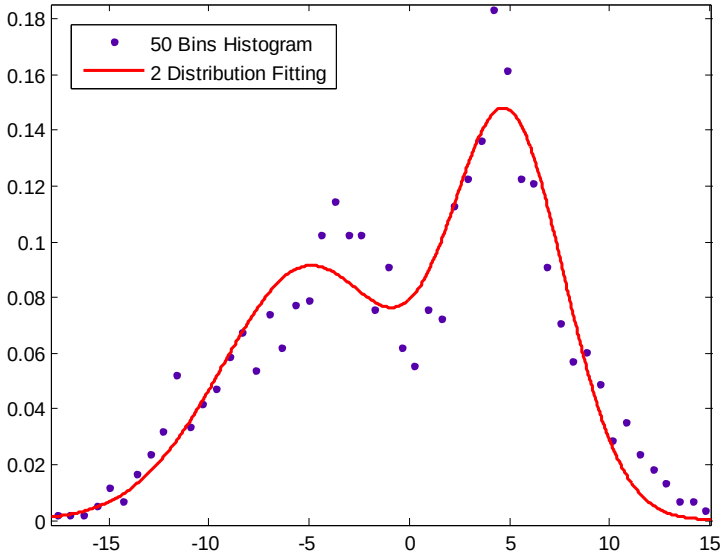


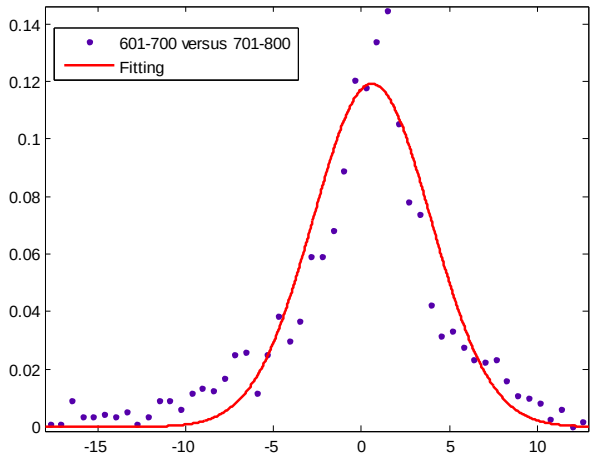
Figure 3.12. Two normal distribution fitting curve for the normalized subtraction between generation 501-600 and 601-700

Table 3.7. Fitting coefficients for two normal distributions with 95% confidence interval for the difference between generation 501-600 and 601-700

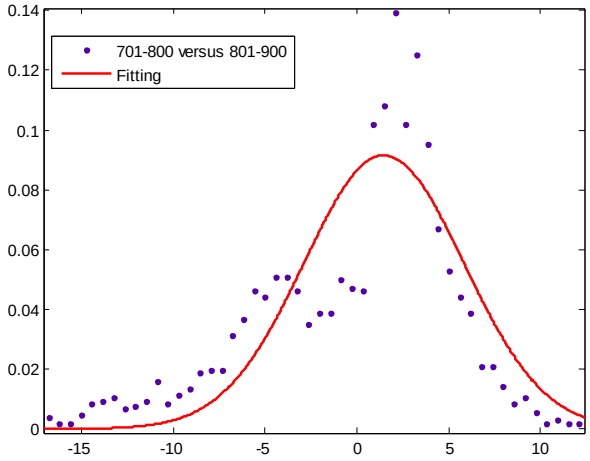
Coefficient	Fitting value with 95% confidence interval	Coefficient	Fitting value with 95% confidence interval
μ_1	4.918 (4.581, 5.255)	μ_2	-4.961 (-5.603, -4.318)
σ_1	2.839 (2.576, 3.101)	σ_2	4.377 (3.872, 4.881)

The two estimated mean values were almost a pair of opposite values. However, the two estimated standard deviations were away from each other for ~1.5. Further observation and analysis of the fitting behavior are a possible research topic.

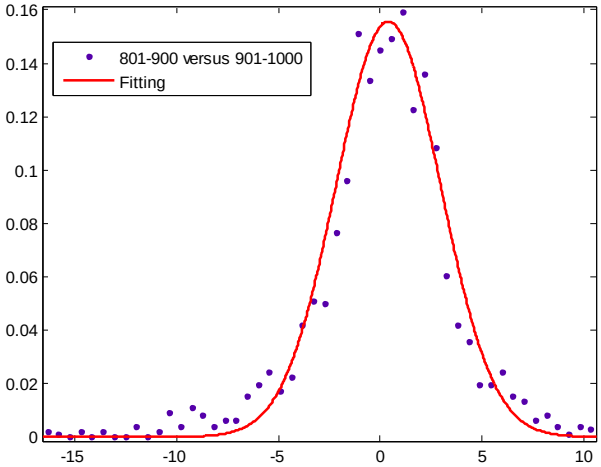
The same procedure was used to analyze the other data: 601-700 generations versus 701-800 generations, 701-800 generations versus 801-900 generations, 801-900 generations versus 901-1,000 generations, 501-600 generations versus 701-800 generations, 501-600 generations versus 801-900 generations, 501-600 generations versus 901-1,000 generations, 601-700 generations versus 801-900 generations, 601-700 generations versus 901-1,000 generations, and 701-800 generations versus 901-1,000 generations. For now, Figure 3.13 shows just the histograms with 50 bins and single distribution fitting curves for the three cases. Table 3.8 summaries the corresponding coefficients. The two normal distribution fitting method was not used for these cases, but it could be a potential way to analyze the behavior of fittings.



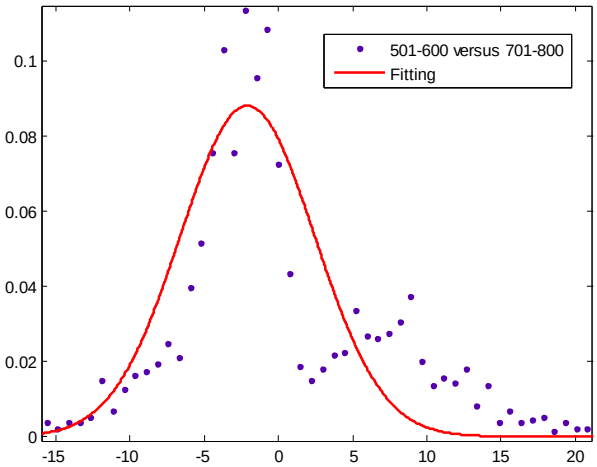
(a) 601-700 versus 701-800



(b) 701-800 versus 801-900

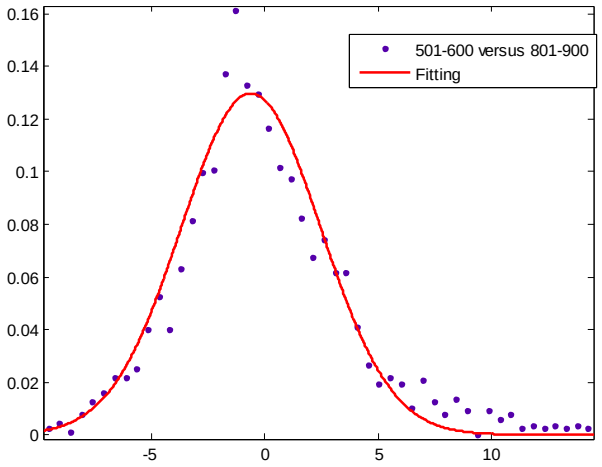


(c) 801-900 versus 901-1,000

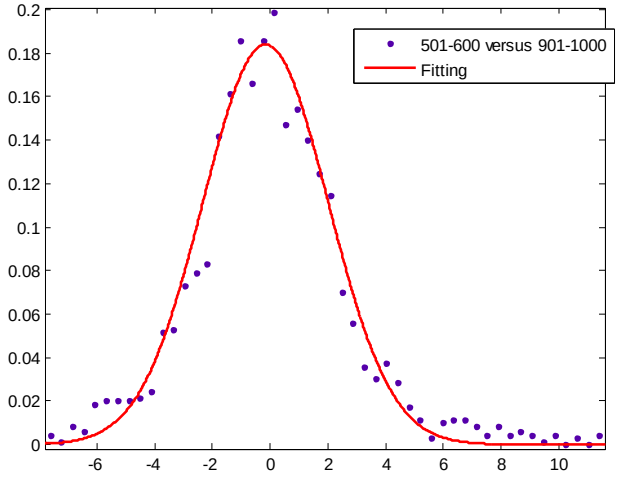


(d) 501-600 versus 701-800

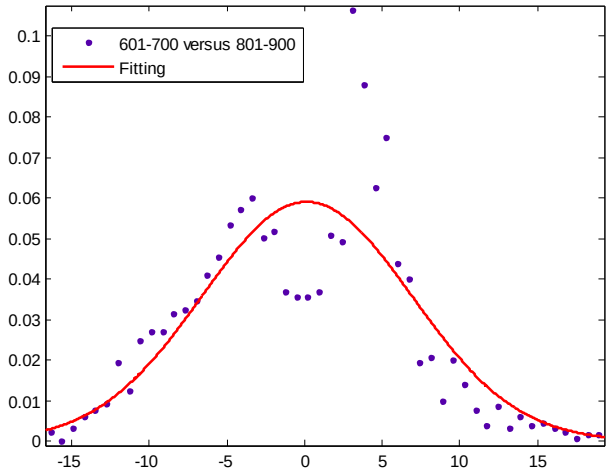
Figure 3.13. Histograms and fitting normal distribution fitting curves for the other cases



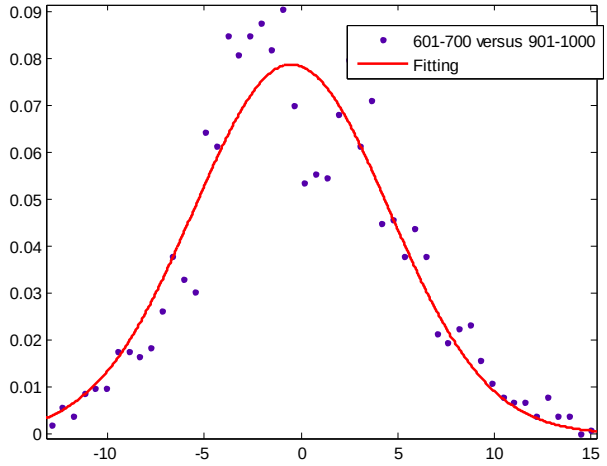
(e) 501-600 versus 801-900



(f) 501-600 versus 901-1,000

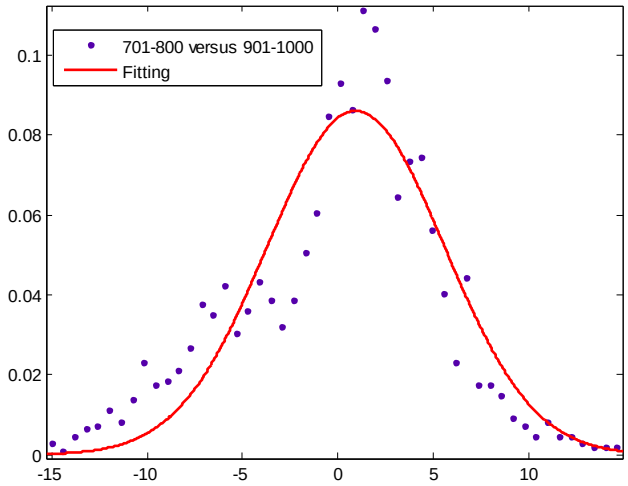


(g) 601-700 versus 801-900



(h) 601-700 versus 901-1,000

Figure 3.13 continued. Histograms and fitting normal distribution fitting curves for the other cases



(i) 701-800 versus 901-1,000

Figure 3.13 continued. Histograms and fitting normal distribution fitting curves for the other cases

Table 3.8. Fitting coefficients of one normal distribution with 95% confidence interval for the other comparisons

	601-700 versus 701-800	**701-800 versus 801-900**	**801-900 versus 901-1,000**
μ	0.6033 (0.278, 0.9286)	1.463 (0.7746, 2.152)	0.3975 (0.2332, 0.5618)
σ	3.344 (3.079, 3.61)	4.359 (3.794, 4.923)	2.563 (2.429, 2.698)
	501-600 versus 701-800	**501-600 versus 801-900**	**501-600 versus 901-1,000**
μ	-2.094 (-2.86, -1.328)	-0.6004 (-0.8045, -0.3963)	-0.1579 (-0.2748, -0.04105)
σ	4.533 (3.907, 5.159)	3.074 (2.907, 3,24)	2.169 (2.073, 2.264)
	601-700 versus 801-900	**601-700 versus 901-1000**	**701-800 versus 901-1,000**
μ	0.1577 (-0.9171, 1.233)	-0.4908 (-0.916, -0.06563)	0.9266 (0.4105, 1.443)
σ	6.752 (5.871, 7.634)	5.063 (4.714, 5.411)	4.636 (4.215, 5.508)

Figure 3.13 and Table 3.8 together indicates that when two potential peaks exist, the estimated variances are greater. Again, these two peaks were likely caused by the underestimate of variance and the auto-correlation together. Indeed, the underestimate of variance phenomena was also a result of the auto-correlation. Therefore, the only reason caused the fitting curve to behave different from expectation was the auto-correlation effect. Therefore, the fitting mechanism was a potential means to detect the auto-correlation as well.

3.6 Summary of Findings

This chapter used the OECD/NEA first benchmark problem, describing a storage fuel pool, to investigate the k_{eff} and flux convergence in criticality Monte Carlo simulations and specifically in MCNP5. Because of the large size of the problem, flux convergence was difficult to reach. Although different initial source distributions yielded similar k_{eff} results, further examination showed that the results were not so convincing since the flux distributions were far away from each other. The only available convergence diagnostics in MCNP5 was the entropy check, which was based on the source distribution. This entropy check mechanism was capable to detect some non-convergence of the flux distribution, but it also gave false positive conclusion in many cases. This weakness of the entropy check was partially due to the lack of the reference criterion and its property that it represented the entire flux distribution with one single number. By using one single entropy value, the diagnostics was easier to understand and apply. However, local flux information was lost as a trade-off. The bounding approach was helpful for the entropy diagnostics by simulating two bounding cases and generating the bounding curves, but it demanded extra computational expense. Another possible solution was visually checking the flux plot, but an experienced user was necessary for this solution, and there is no clear criterion.

To propose an automatic convergence diagnostics using the local flux distribution information, a statistical analysis was conducted. A simplified problem based on the benchmark problem was used as a faster convergence example. The mesh tally was imposed on this problem to obtain flux distributions. The successive flux results were used to yield the statistical data for further analysis. Relying on the curve fitting function of MATLAB, normal distribution function was used to fit the histogram of the data. Reported fitting mean value and variance were far away from expected values and changed dramatically, thus providing a quantitative indication (detection) of non-convergence and/or auto-correlation. The analysis of this behavior was important for further development of new diagnostics.

CHAPTER 4

HIGHER EIGENMODE ACCELERATION METHODOLOGY USING MODIFIED POWER METHOD

4.1 Background

As mentioned in chapter two, the easiest (although ineffective) way for achieving convergence was just to let the simulation run for (more than) a sufficient number of generations. A direct use of this solution was impractical because of the unacceptable computational expense, which pointed to the need for a convergence acceleration methodology that would reduce the computational expense with faster convergence and thus reduce the required number of iterations. This chapter discusses a recently available acceleration methodology based on the power iteration method.

The power iteration method [3] is the basis of the Monte Carlo criticality simulation (and most of deterministic criticality simulations as well). This iterative method for solving eigenvalue problems eliminates the higher harmonics of a well-behaved vector or function. As a result, only the fundamental eigenfunction remains as a steady-state result, which represents a nuclear system in normal operation. At the same time, the fundamental eigenvalue, k_{eff} (effective multiplication factor), is also computed. The convergence rate of the power iteration method is $|k_2|/k_1$, where $|k_2|/k_1$ is the dominance ratio (DR). If the DR is close to one, the power iteration method needs numerous iterations to remove higher components, which results in a slow convergence. This slow convergence is a major obstacle in current criticality simulations for high DR system.

Recently, T. Booth proposed two new methods [37] [38] to compute the second eigenvalue k_2 and eigenfunction Ψ_2 using Monte Carlo simulation. T. Yamamoto [39]

derived an alternative demonstration for the first method in Ref. 37. He claimed that the strict convergence condition in Ref. 37 was not necessary that the method could be applied to more problems. One disadvantage of the first method was that it could only compute the second eigenvalue and eigenfunction at one time. Another drawback was the complexity of the mathematical derivation. The second method T. Booth proposed in Ref. 38 was easier to understand and implement. In addition, this method could compute k_1 (fundamental eigenvalue, all known as k_{eff}), k_2, Ψ_1 (fundamental eigenfunction), and Ψ_2 at the same time. The major characteristic of this method was the increasing of the convergence rate from $|k_2|/k_1$ to $|k_3|/k_1$, whose demonstration could be found in Ref. 40.

4.2 Review of the Modified Power Iteration Method

Recall Eq. (1.3) to (1.5) as the basic concepts for the power iteration method. An operator A with eigenvalue k_i and corresponding eigenfunction ψ_i satisfies the relation

$$A\psi_i = k_i\psi_i \text{, in which } k_1 > |k_2| > |k_3| > \cdots\cdots. \tag{4.1}$$

Any "well-behaved" function ψ can be expanded in eigenfunctions

$$\psi = \sum_i a_i\psi_i \,. \tag{4.2}$$

With renormalization, the power iteration method could estimate the fundamental eigenfunction and eigenvalue as the iteration number n approaches to infinity

$$\lim_{n\to\infty} \frac{1}{k_1^n} A^n\psi = \psi_1 \text{ and } k_1 = \lim_{n\to\infty} \frac{A^n\psi}{A^{n-1}\psi}. \tag{4.3}$$

A parameter x is introduced to the expansion of a well-behaved function Ψ as a part of the coefficient

$$\psi = \sum_i (a_i + b_i x)\psi_i \,. \tag{4.4}$$

After iterations, two limits similar to those in Eq. (4.3) are acquired. The difference is that now the first two eigenfunction components remain rather than only the fundamental one. The first limit estimating the eigenfunction is

$$\lim_{n\to\infty}\frac{1}{k_1^n}A^n\psi=\lim_{n\to\infty}\frac{1}{k_1^n}\left((a_1+b_1x)k_1^n\psi_1+(a_2+b_2x)k_2^n\psi_2\right). \tag{4.5}$$

The second limit estimating the eigenvalue is

$$\lim_{n\to\infty}\frac{A^n\psi}{A^{n-1}\psi}=\lim_{n\to\infty}\frac{(a_1+b_1x)k_1^n\psi_1+(a_2+b_2x)k_2^n\psi_2}{(a_1+b_1x)k_1^{n-1}\psi_1+(a_2+b_2x)k_2^{n-1}\psi_2}. \tag{4.6}$$

The values of the limits depend on the parameter x. For example, if $x=-b_2/a_2$, the coefficient (a_2+b_2x) will equal to zero, causing the fading away of the second eigenfunction component. As a result, Eq. (4.5) turns to

$$\lim_{n\to\infty}\frac{1}{k_1^n}A^n\psi=\lim_{n\to\infty}\frac{1}{k_1^n}(a_1+b_1x)k_1^n\psi_1=(a_1+b_1x)\psi_1. \tag{4.7}$$

By applying renormalization after each iteration, this limit is an accurate estimate of the fundamental eigenfunction. In addition, Eq. (4.6) turns to

$$\lim_{n\to\infty}\frac{A^n\psi}{A^{n-1}\psi}=\lim_{n\to\infty}\frac{(a_1+b_1x)k_1^n\psi_1}{(a_1+b_1x)k_1^{n-1}\psi_1}=k_1, \tag{4.8}$$

which is also an accurate estimate of the fundamental eigenvalue. Likewise, if $x=-b_1/a_1$, the coefficient (a_1+b_1x) will equal to zero. Therefore, the two limits will converge to the second eigenfunction and eigenvalue with proper renormalization technique.

In order to find the proper parameter x, which satisfies the conditions, one divides the entire system into two sub-regions, R_1 and R_2. No strict requirement applies to the choosing of these sub-regions—R_1 and R_2 could overlap; the union of R_1 and R_2 does not need to cover the entire space. In both sub-region, one then estimates the eigenvalue with parameter x using

$$k^{R_1}=\frac{A\psi^{R_1}}{\psi^{R_1}}\ \text{and}\ k^{R_2}=\frac{A\psi^{R_2}}{\psi^{R_2}}. \tag{4.9}$$

To clarify this expression, two new function are introduced

$$\psi'=\sum_i a_i\psi_i\ \text{and}\ \psi''=\sum_i b_i\psi_i. \tag{4.10}$$

If the procedure converges, the estimates of k in R_1 and R_2 are the same. Therefore, setting k^{R_1} and k^{R_2} equal yields a quadratic equation for parameter x

$$\frac{A\psi'_{R_q} + xA\psi''_{R_1}}{\psi'_{R_1} + x\psi''_{R_1}} = \frac{A\psi'_{R_2} + xA\psi''_{R_2}}{\psi'_{R_2} + x\psi''_{R_2}}. \tag{4.11}$$

The two roots of this equation satisfy the two conditions above, which induce Eq. (4.5) and Eq. (4.6) to converge to the fundamental and second eigenfunction and eigenvalue respectively. After one iteration, the two new achieved vectors $A\psi'$ and $A\psi''$, which are employed to initialize the next iteration, function as ψ' and ψ'' in the following calculation. In addition, one root of x, say x_1, leads the estimate of $A\psi' + x_1 A\psi''$ to $\hat{\psi}_1$, and the other root of x, say x_2, leads the estimate of $A\psi' + x_2 A\psi''$ to $\hat{\psi}_2$. The hat in the expressions means that the eigenfuctions are estimates rather than accurate results.

4.3 Application to a Matrix Eigenmode Problem

To show the validity of this method, consider an arbitrary matrix P

$$P = \begin{pmatrix} -1.1500 & -0.6250 & -3.0250 & -1.9500 \\ 0.5167 & 1.3750 & 0.6417 & 0.7167 \\ 2.6833 & -0.8750 & 4.0583 & -0.1167 \\ -0.9833 & 2.1250 & 0.3917 & 4.2167 \end{pmatrix}. \tag{4.12}$$

The eigenvalues of this matrix are 4, 3, 1, and 0.5. Two arbitrary initial vectors (also called trigger vectors) **a**=(1 1 1 1) and **b**=(1 0 1 1), which functioned as ψ' and ψ'' in Eq. (4.10), were used. The parameter x was inserted in front of vector **b**. The first component of the vectors was taken as the first sub-region and the last three components were taken as the second sub-region in calculation for 100 iterations. Figure 4.1 illustrates the evolution of the two eigenvalues for these vectors.

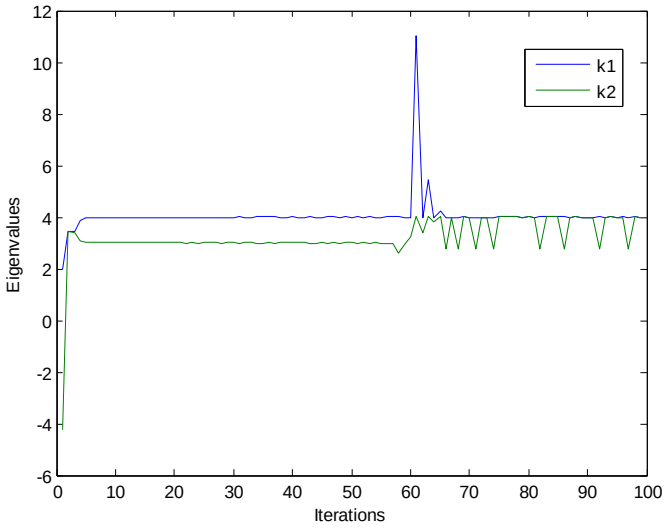


Figure 4.1. Evolution of the eigenvalues with initial vectors a=(1 1 1 1) and b=(1 0 1 1)

The estimates converged to the first two eigenvalues (4 and 3) after ~8 iterations and kept the convergence until ~50th iteration. After that, the green line didn't converge to k_2 anymore and switched to k_1 instead; in the meantime, a huge jump of the blue line appeared. Finally, the blue line kept estimating k_1 and the green line kept oscillating between k_1 and k_2 randomly. A speculation to explain this random oscillation was that it is due to the round-off error.

As an alternative case, the initial vectors were changed to **a**=(1 1 0 0) and **b**=(0 0 1 1) in order to perform the calculation again with 50 iterations shown in Figure 4.2. The estimate of k_2, which converged to the third eigenvalue for some iterations, never converged to the expected second eigenvalue before it collapsed to the fundamental one. Another observation from the calculation was that the calculation was still running after ~33 iterations although the estimated eigenvalues were both infinite, which cannot be shown in Figure 4.2. In principle, both vectors **a** and **b** converged to the fundamental eigenvector. Therefore, in some cases, the entire procedure would not generate a quadratic equation of x and the computation would stop; in other words, a collapse of calculation took place. In other cases, due to the round-off error, the difference between

the two vectors kept randomly changing, which either cause dthe random oscillation as in Fig 4.1 or the a meaningless estimate as in Fig 4.2.

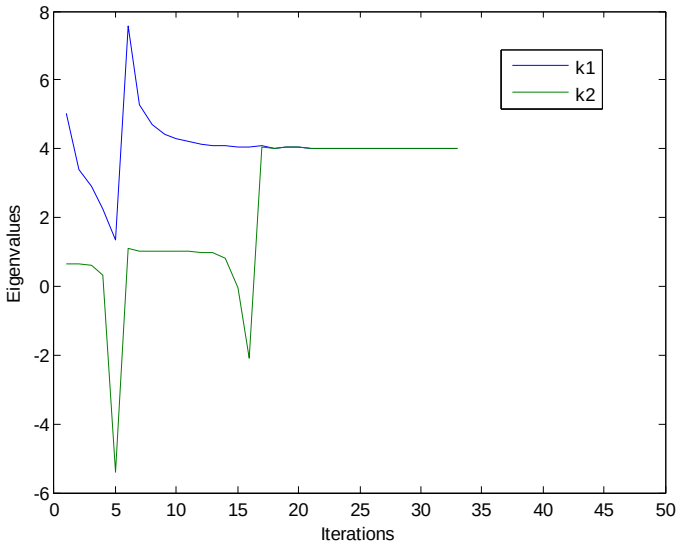


Figure 4.2. Evolution of the eigenvalues with initial vectors a=(1 1 0 0) and b=(0 0 1 1)

As a comparison of the two cases, different choices of the initial vectors also affected the calculation. This is surprising because it is comprehensible that the selection of the initial vectors should not affect the results as long as both of them contain all the eigenfunction components. However, it turns out that the validity of the convergence also depends on the ratio of the first two components in the initial vectors. If the ratio for the two vectors is very close, distinguishing the first two eigenvalues tends to be difficult, causing the failure of the method. As an example, the first to second eigenvector component ratio for vector (1 1 0 0) is 2.1424/3.0838=0.6947, which is close to the ratio for vector (0 0 1 1) that is 4.9990/7.1956=0.6947. Indeed, the difference of the two ratios is on the order of 10^{-4}. This is why no convergence to k_2 took place in Fig 4.2. Another example uses the first set of the initial vectors. The ratio for vector (1 1 1 1) is 7.1414/10.2794=0.6947 and the ratio for vector (1 0 1 1) is 5.7131/9.2515=0.6175. These

two ratios are sufficiently far away from each other, so the two initial vectors induced the successive convergence shown in Fig 4.1.

In summary, depending on the specific selection of initial vectors, two types of collapse take place during the calculation of the modified power iteration method. One collapse is that the estimate of k_2 will eventually converge to k_1 as the shrinking of the second eigenvector components. The other collapse is the effect of the numerical round-off error, which generally occurs after the first collapse. Therefore, aiming to overcome some of these drawbacks, several refinements were proposed, which will be reviewed in the next section.

4.4 Refinements

In Ref. 41, J. Gubernatis and T. Booth proposed two refinements.

4.4.1 The First Refinement

As previously mentioned, each iteration is initialized with vectors P*$\mathbf{a}$ and P*$\mathbf{b}$, which are denoted as $\mathbf{a}_{new}$ and $\mathbf{b}_{new}$. from the previous iteration with parameter x in front of P*$\mathbf{b}$. The first refinement replaces P*$\mathbf{a}$ with P*$\mathbf{a}$+x_2*P*$\mathbf{b}$, which is the latest estimate of Ψ_2, as $\mathbf{a}_{new}$ to trigger the next iteration. This replacement prevents vectors $\mathbf{a}$ and $\mathbf{b}$ from merging together to the fundamental eigenvector after several iterations, which would cause the failure of the method with finite computational accuracy. In deterministic computation, one root x_1 can be used as a convergence criterion because vector $\mathbf{b}$ is converging to the second eigenvector and consequently, x_2 is converging to zero. Figure 4.3 depicts the performance of 50 iterations using this refinement with the initial set of vectors $\mathbf{a}$=(1 1 0 0) and $\mathbf{b}$=(0 0 1 1) that did not produce two eigenvalues before.

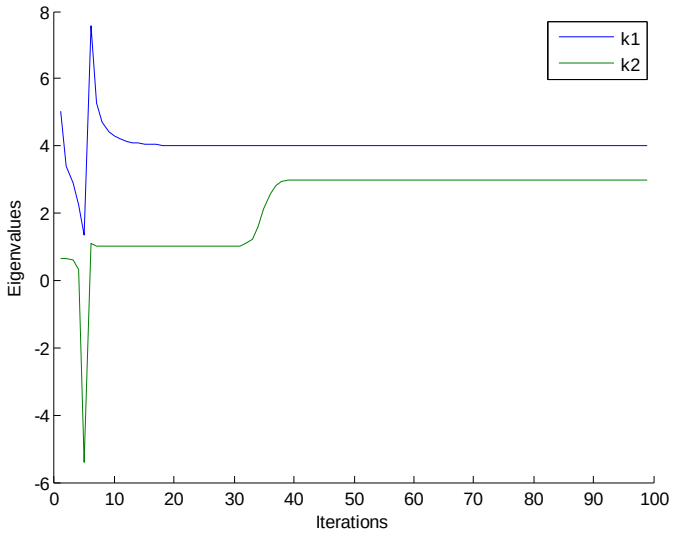


Figure 4.3. Evolution of the eigenvalues for the first refinement with initial vectors a=(1 1 0 0) and b=(0 0 1 1)

This refinement overcame the collapse, which caused the incapability of keeping computing eigenvalues for many iterations. Indeed, this collapse was not significant for a deterministic or matrix problem since generally, the calculation would meet the stop criterion before the collapse occurred. However, the collapse was crucial for Monte Carlo simulations because the more generations were used for the simulation, the better confidence interval the simulation would reach.

4.4.2 The Second Refinement

Although the first refinement resolved the collapse problem, it did not increase the convergence rate to the fundamental eigenfunction, which was as important as the eigenvalues, because no changes applied to vector **b**. The second refinement could increase this convergence rate by updating vector **b** as well. On the basis of the replacement of vector **a**, this refinement replaced vector P***b*** with a similar expression P***a***+x_1*P***b** to represent $\mathbf{b}_{new}$. Figure 4.4 shows the evolution of the eigenvalues for this refinement with the same initial vectors as the first refinement, but only for 50 iterations.

Figure 4.5 illustrates how vector **b** converges to the fundamental mode eigenfunction measured by the error for the power iteration method, the modified power iteration method with the first refinement, and the modified power iteration method with the second refinement for 50 iterations. Here, the error is l-2 norm of the difference between the normalized estimate of the eigenfunction and the accurate eigenfunction.

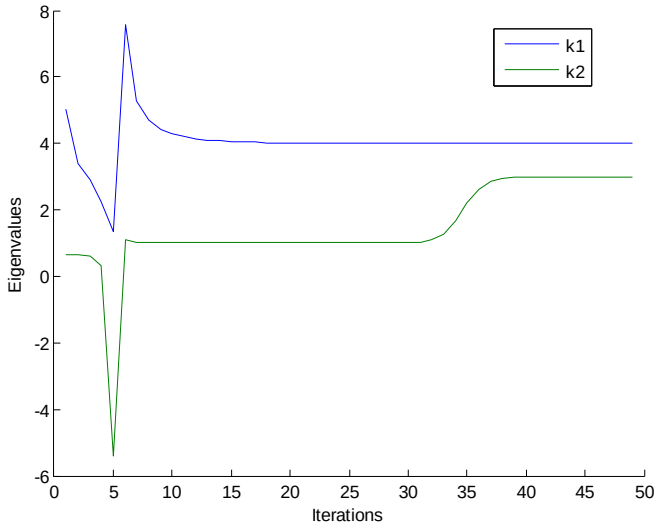


Figure 4.4. Evolution of the eigenvalues for the second refinement with initial vectors a=(1 1 0 0) and b=(0 0 1 1)

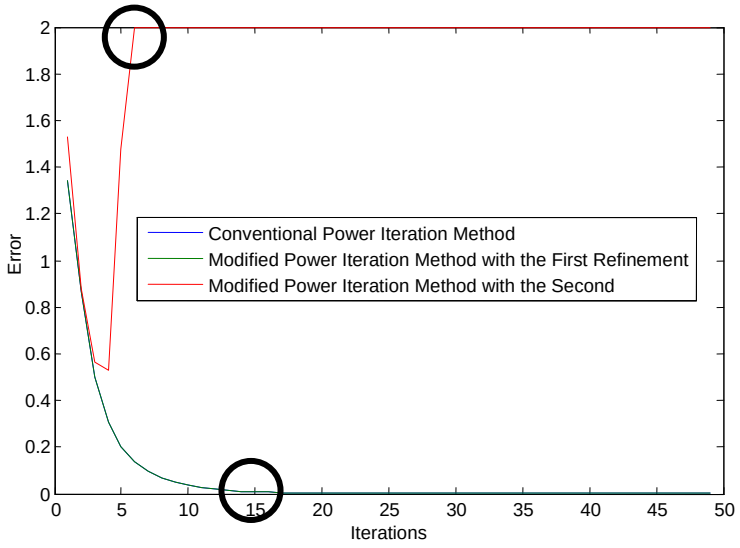


Figure 4.5. Comparison of the convergence to the fundamental mode eigenfunction

Blue and green lines in Figure 4.5 actually overlapped as expected since the first refinement kept the convergence rate. The red line representing the second refinement behaved oddly at first sight, but the underlying cause was easy to grasp. For the eigenvalue problem $A*\Psi=k*\Psi$, where A is an operator or a matrix, k is an eigenvalue (a scalar quantity), and Ψ represents the corresponding eigenfunction or eigenvector. If Ψ is an eigenfunction for k, $(-\Psi)$ is also an eigenfunction for k. Theoretically, power iteration could not guarantee the convergence of the eigenfunction to Ψ or $(-\Psi)$. In Figure 4.5, if the reference eigenfunction was noted as Ψ_1, the odd behavior of the red line was due to its convergence to $(-\Psi_1)$. Therefore, the error value, which was two times of the normalized eigenfunction, was 2. Generally speaking, in deterministic method, this would not be a big issue because one could always obtain Ψ_1 from $(-\Psi_1)$ easily. Again, comparison between the red line and the green line, especially paying attention to when they achieved convergence as marked in the figure, exemplified the improvement of the convergence rate.

Unfortunately, this refinement also contained a drawback in numerical implementations. As the iterations proceeded, vector **b** converged to Ψ_1 and vector **a** converged to Ψ_2, resulting in the parameter x_1 in P*$\mathbf{a}$+x_1*P*$\mathbf{b}$ converging to infinity and eventually out of computational range. If the calculation shown in Figure 4.4 continued, it would collapse after the 69^{th} iteration, when root x_1 grew to 10^{+14}. This was a different collapse from the previous one, which was due to the round-off error. However, similar to the previous one, this collapse was not crucial for deterministic applications since the results should have passed the stopping criteria before the calculation collapsed. Still, when this refinement was applied to the Monte Carlo method, the user had to be aware of the possibility of collapses if he wanted to reduce confidence interval by running more iterations.

4.4.3 A New Refinement

In order to not only preserve the long calculation and fast convergence properties, but also prevent the parameter x from diverging to infinity, this work devised a new improvement described in this section. The new expressions of $\mathbf{a}_{new}$ and $\mathbf{b}_{new}$ trigger the next iteration with

$$\begin{aligned} \vec{a}_{new} &= \frac{P \bullet \vec{a} + x_1 P \bullet \vec{b}}{\widehat{k}_1} + \frac{P \bullet \vec{a} + x_2 P \bullet \vec{b}}{\widehat{k}_2} \\ \vec{b}_{new} &= \frac{P \bullet \vec{a} + x_1 P \bullet \vec{b}}{\widehat{k}_1} - \frac{P \bullet \vec{a} + x_2 P \bullet \vec{b}}{\widehat{k}_2} \end{aligned} . \tag{4.13}$$

This approach employed a linear combination—1 and 1 for $\mathbf{a}_{new}$ and 1, -1 for $\mathbf{b}_{new}$—of the latest estimated eigenfunctions, which were divided by the latest estimated corresponding eigenvalues before a regular renormalization. The reason of the division by $\widehat{k}_1$ or $\widehat{k}_2$ was to keep the second eigenfunction component detectable and prevent the collapse due to the round-off error. In this refinement, neither vector **a** nor vector **b** would converge to eigenfunction itself since they were being updated every iteration. However, the

expressions P*$\mathbf{a}$+x_1*P*$\mathbf{b}$ and P*$\mathbf{a}$+x_2*P*$\mathbf{b}$ converged to the fundamental and second eigenfunctions respectively. The coefficients of the linear combination were artificial, controlled to what numbers the two roots of the parameter x would converge. For the choice in Eq. (4.13), the roots of the parameter x converged to a pair of values with opposite signs, which were determined by the matrix P as well. Figure 4.6 illustrates the evolution of eigenvalues using the same vectors for 200 iterations, and Figure 4.7 depicts the convergence to the fundamental mode eigenvector for the first 30 iterations. The two figures confirmed that this new refinement was valid and met the expectations.

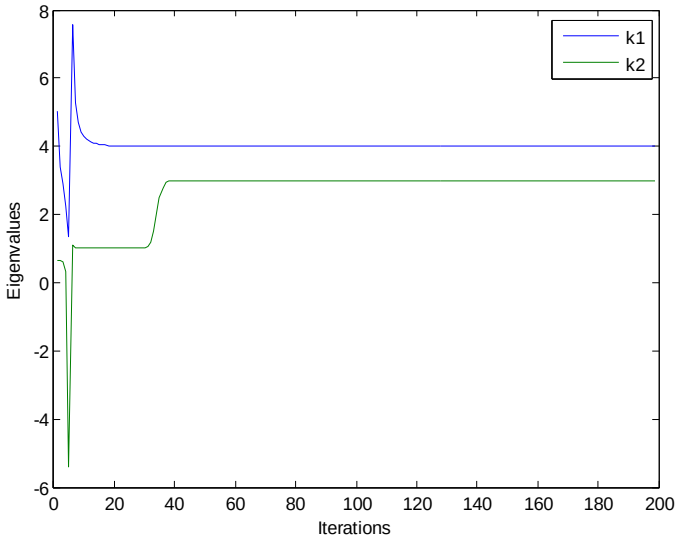


Figure 4.6. Evolution of the eigenvalues for the new refinement with initial vectors a=(1 1 0 0) and b=(0 0 1 1)

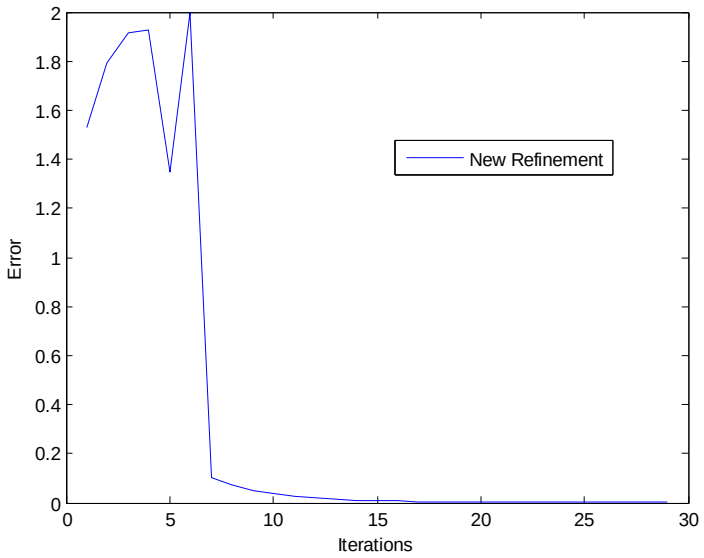


Figure 4.7. Convergence to the fundamental mode eigenfunction using the new refinement

4.5 Implementation into Monte Carlo Simulation

Based on the positive results of applying the modified power iteration method to the matrix problem, implementation into Monte Carlo simulation was attempted. The simulation employed a one-dimensional mono-energetic fissile system described in Ref. 41. The system extended from -10cm to +10cm with reflective boundaries along both x and y directions. In z direction, the system, the width of which was 9 cm (-4.5 cm to +4.5cm) with a vacuum boundary condition, was divided into 100 small 0.09 cm-wide meshes. The fissile material was uniform and isotropic with the following data

$$\Sigma_{total} = 1.0cm^{-1}, \Sigma_{scattering} = 0.8cm^{-1}, \Sigma_{capture} = \Sigma_{fission} = 0.1cm^{-1}, \nu = 3.0. \qquad (4.14)$$

Two sub-regions for two estimates of eigenvalues were -4.5cm to 0cm, noted as the left region, and 0cm to 4.5cm, noted as the right region, along the z direction. The initial two sets of weights or source distributions were

$$A = \begin{cases} 0.6 & for \quad z < 0 \\ 1.4 & for \quad z > 0 \end{cases} \text{ and } B = \begin{cases} 1.6 & for \quad z < 0 \\ 0.4 & for \quad z > 0 \end{cases}. \qquad (4.15)$$

The collision estimator was used to gather the weights deposited in the left and right sub-regions as well as the weights deposited in the 100 small meshes similar to the mesh tally mechanism. After each iteration, a quadratic equation about x was formed based on the deposited weights. Solving this equation gave the two roots of the parameter x leading to the two eigenvalues in Eq. (4.6) and corresponding eigenfunctions in Eq. (4.5). Then, according to Eq. (4.13), two new weights distributions, which should be normalized, were generated based on the latest estimated eigenvalues and eigenfunctions. The normalization used was to maintain the summation of the absolute weights in each mesh as N/100, where N was the total number of particles per generation and 100 represented the number of meshes. The source generating mechanism was called "resampling method" that was based on Yamamoto's method in Ref. 39. Section 4.6 will discuss this source generating method and some other weight cancellation and source sampling approaches. Following the above procedure, a simulation of the problem used 10,000 particles per generation, 130 total cycles, and 30 inactive cycles. The linear combination pair of the new trigger distributions was the same as described in Eq. (4.13).

Table 4.1. Eigenvalue results for the Monte Carlo simulation

k_1	k_2
1.30567 ± 0.00042	0.95428 ± 0.00063

Table 4.1 shows the eigenvalue results and their estimated standard deviations. Compared with a reference deterministic results given in Ref. 40 as $k_1 = 1.30534$ and $k_2 = 0.95488$, the differences between the deterministic and Monte Carlo results for k_1 and k_2 were within the 1σ confidence interval for both, which indicated that the eigenvalue results were consistent. In addition to the eigenvalue results, the eigenfunction information was also available from the 100-mesh weights deposit. Figure 4.8 shows the

normalized eigenfunction results averaged over 100 active generations, which were similar to analytic predictions. [42] The consistence of both the eigenvalue results and the eigenfunction results indicated the validity of the implementation of the modified power iteration method with the new refinement.

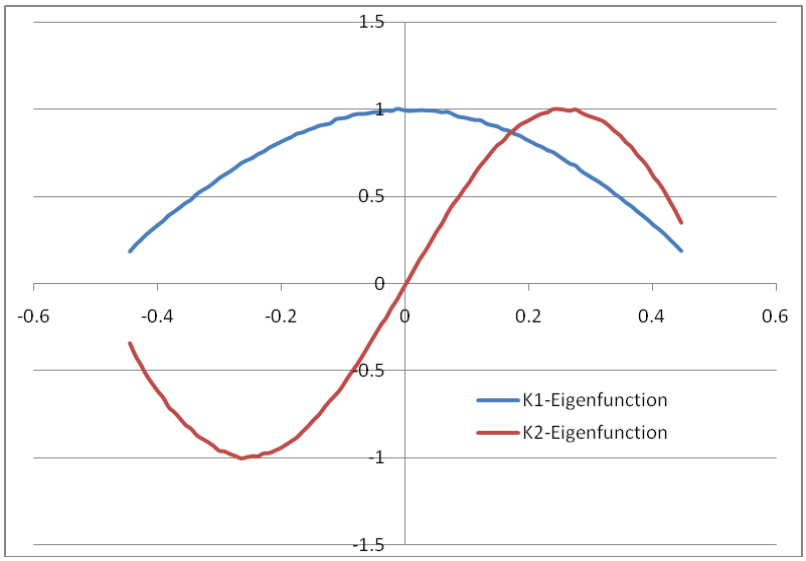


Figure 4.8. Normalized eigenfunction results of the Monte Carlo simulation with modified power iteration method and new refinement

4.6 Weight Cancellation and Source Sampling

For the original form of the modified power iteration method as shown in previous section without any refinement methods, the implementation into Monte Carlo simulation did not need any action for adjusting weights of particles. The two sets of source distributions with positive weights as in a traditional Monte Carlo method should work well. However, because of the limitation of the original form discussed earlier, the refinements, which included summation or subtraction of the source distributions, were necessary. Therefore, a cancellation or accumulation mechanism of the source

distribution was needed in order to represent two proper source distributions for the next iteration.

In Ref. 37, T. Booth proposed to use a cancellation mechanism similar to the point detector concept. For a simulation with N particles per generation, each particle possesses two weights, representing two sets of weight distribution. Before the new iteration begins, N new particles, which will be treated as a point detector, are sampled randomly over the entire system. For each particle generated for an old point carrying two weights, the point fluxes in each new detector position are computed corresponding to the two weights. Therefore, a new set of source points is obtained with two weight sets after each iteration. The weights carried by each point are unbiased, so this mechanism is an exact cancellation method. However, the computational cost for a point detector mechanism is expensive. Moreover, there actually exist N point detectors in a single iteration which is too costly.

In Refs. 39 and 40, two similar methods were proposed for the weights cancellation. By approximating the flux distribution with small meshes, two sets of mesh tallies are calculated corresponding to the two sets of trigger source or weight distributions. Then, the weights cancellation or accumulation is easily implemented with simple summation or subtraction of the two sets of flux or weight distribution. The difference of the two methods is how to generate new sources according to the new flux or weight distribution. In Ref. 39, T. Yamamoto resampled N/M particles, where N represented the total number of particles per generation and M represented the number of meshes, from each mesh evenly with the updated weight to represent the weight distribution. In Ref. 40, T. E. Booth and J. E. Gubernatis used a similar procedure to obtain the mesh-based weights distribution. Instead of resampling source particles, they used the old source points generated after the previous iteration and gave them proper weights. In summary, the basis of both of the methods is the approximation of the weight distribution and no evidence indicates which one is a better approximation. In the

previous Monte Carlo simulation, the resampling method was used for the cancellation of source generation.

4.7 Variance of the Eigenvalues

To examine the impact of this new approach on the variance, 10 repetitions of the Monte Carlo simulation were performed with different random number seeds. Each repetition employed 10,000 particles per generation, 100 inactive generations, and 400 active generations, which were divided into 5 groups of 80 generations each. Only the first 50 generations of each group (80 generations) provided effective data while the remaining 30 generations acted as a gap between data sets to eliminate or significantly reduce auto-correlation of k_{eff} for this problem.[43] Thus, total 50 sets of almost independent data were obtained for analysis.

Before the analysis of variance, some concepts will be reviewed. Suppose one wanted to estimate k_{eff}; by repeating simulations, he would obtain results from N=50 active generations for M repetitions. Here, M=50 was selected. Therefore, n=1,2,…,50 represented active generations and m=1,2,…,50 represented repetitions. The result from the n^{th} generation of the m^{th} repetition could be written as k_n^m. For the traditional Monte Carlo method, only one simulation was performed and it equaled to picking one of the 50 repetitions to compute the k_{eff} with its standard deviation

$$\bar{k}^m = \frac{1}{N}\sum_{n=1}^{N} k_n^m , \tag{4.16}$$

$$\sigma^m = \sqrt{\frac{\left[\left(\sum_{n-1}^{N}\left(k_n^m\right)^2 - N\times\left(\bar{k}^m\right)^2\right)\right]}{N(N-1)}} . \tag{4.17}$$

By using the data from 50 repetitions, an estimate of the mean value of the average k_{eff} and its standard deviation was

$$\bar{k} = \frac{1}{M}\sum_{m=1}^{M} \bar{k}^{m}\,, \tag{4.18}$$

$$\bar{\sigma} = \frac{1}{M}\sum_{m=1}^{M} \bar{\sigma}^{m}\,. \tag{4.19}$$

However, due to the correlation between sequence generations, this estimated standard deviation would be underestimated. [15] One way to estimate the true variance was to use the data from the 50 repetitions as

$$\sigma_{true} = \sqrt{\frac{\left(\sum_{m=1}^{M}\left(\bar{k}^{m}\right)^{2}\right) - M \times \left(\bar{k}\right)^{2}}{M-1}}\,. \tag{4.20}$$

To eliminate the effects of the resampling method used here, a simulation with traditional Monte Carlo method used for comparison also employed the same resampling method. Table 4.2 lists the results.

Table 4.2. Comparison of the results from 50 repetitions

	Traditional Monte Carlo using resampling	**Modified Monte Carlo with new refinement**	
		k_1	**k_2**
$\bar{k}$	1.30548	1.30542	0.95499
$\bar{\sigma}$	0.00055	0.00048	0.00090
σ_{true}	0.00069	0.00053	0.00071

The first column of the table (last two rows) showed the underestimate due to the correlation effect: reported variance was always smaller then the true variance. The second column illustrated the same behavior of k_1 from the modified power method with the new refinement. Comparison of the expected value of the reported variance from the traditional Monte Carlo simulation and the modified Monte Carlo method showed the

better estimate from the modified method. The decreasing of the reported variance was probably because of the faster removal of the second eigenfunction component, which was also an estimate. This same reason could explain the similar behavior of the true variance. In addition to the decrease of variance, the modified method also reduced the underestimation effect, which increased the reported to true σ ratio from 0.8 to 0.9. The real reason for this advantage was not certain, but we speculate that it may be due to the updating of the source weights for each iteration by a linear combination. The least expected result in Table 4.2 was the third column which indicated the larger reported and true σ than in the other two cases. Moreover, the reported variance in this case is greater than the true variance. While we had not performed sufficient research yet to explain this behavior, we suspect that it was related to the second eigenmode features, and this uncommon result could be a possible research topic in the future.

4.8 Summary of Findings

This chapter first reviewed the principle of the modified power iteration method, which could compute the first and second eigenvalues and eigenfunctions simultaneously. This method could also increase the convergence rate to the fundamental eigenmode. A matrix problem was used to illustrate the validity of the modified method. Although this method worked well in its original form, some drawbacks came along with the method. Therefore, two refinements previously published by other authors were reviewed and exemplified first, and a new refinement was proposed next. This new refinement was implemented into a Monte Carlo simulation. The eigenvalue and eigenfunction results from the simulation were consistent with the analytical expectation and deterministic calculations. Analysis based on the reported variance, weight cancellation, and source generating was performed in order to understand the characteristics of the modified method and improve its robustness.

In summary, this modified power iteration method was proved valid for both deterministic method and Monte Carlo method. The advantages of this method—such as the ease of understanding and implementation, obtaining more information at one time, and faster convergence—were quite impressive. However, since this is a relatively new area of research, many questions regarding behaviors or features of this method were still open. Therefore, this field is attractive and has potential to improve the efficiency of Monte Carlo criticality simulations.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This thesis described the potential obstacles in the Monte Carlo criticality simulations. Among these obstacles, this thesis focused on the criticality convergence diagnostic and convergence acceleration issues. By applying the entropy check used in MCNP5 to the OECD/NEA first benchmark problem (storage fuel pool), which has the loosely coupled slow convergence properties, chapter three illustrated the validity of the method in some cases. However, due to the definition of the entropy value, local information is missing, which caused the method to fail to detect the non-convergence for some cases. The bounding property would help for the convergence diagnostics, but it needs unpractical extra computational expense. Chapter three proposed potentially more reliable diagnostics based on the analysis of the local distribution. The normal distribution fitting technique carried more information about the source distribution evolution than the entropy value. Therefore, this fitting technique may be expected to provide a more robust diagnostics. However, how to utilize the resulting data to discern non-converged flux distribution from the converged one was not quite clear because the fitting was contaminated by the loose coupling property of the problem. A robust convergence criterion based on first principles is future research topic.

In the modified power iteration method, obtaining the fundamental and second eigenvalues and eigenfunctions is possible using the Monte Carlo method. A significant feature of this method is the improvement of the convergence rate. Chapter four reviewed the concepts of the method and some refinements aimed to stabilize the simulation. A new refinement was proposed for numerical stability and to maintain the improvement of the convergence rate at the same time. Some preliminary results obtained by applying the modified method to a simple problem indicated its capability. The analysis of the

variance reported along with the eigenvalue results exhibited some unexpected enhancement. Therefore, more research is needed on the prediction of the variance, especially for the second eigenvalue. In addition, by introducing negative-weighted particles in the simulation, a proper way to cancel the positive and negative weights is established, crucial for the accuracy. Although Chapter 4 included an approximated mechanism for the cancellation, it would not function well for a huge complex system. Even if the approximation mechanism were very accurate, the computational expense will be intolerable to maintain the accuracy. As a result, a weight cancellation and source generating mechanism without approximation is needed for the further implementation of the method.

In summary, several fundamental issues appeared related to convergence diagnostics and acceleration of Monte Carlo simulations of nuclear systems. To ensure efficiency and accuracy of such simulations, more research is necessary to develop for some more robust algorithm.

REFERENCES

[1] E. E. Lewis and W. F. Miller, Jr., *Computational Methods of Neutron Transport*, American Nuclear Society, Inc. (1993).

[2] I. Lux and L. Koblinger, *Monte Carlo particle transport methods: neutron and photon calculations*, CRC Press (1991).

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences) (3rd Edition)* (1996).

[4] R. Walpole, R. Myers, S. Myers, and K. Ye, *Probability and statistics for engineers and scientists*, Pearson Education, Inc. (2007).

[5] T. Yamamoto, T. Nakamura, and Y. Miyoshi, "Fission Source Convergence of Monte Carlo Criticality Calculations in Weakly Coupled Fissile Arrays," *Journal of Nuclear Science and Technology*, **37**, 41-52 (2000).

[6] R. N. Blomquist and E. M. Gelbard, "Alternative Implementations of the Monte Carlo Power Method," *Nuclear Science and Engineering*, **141**, 85-100 (2002).

[7] R. N. Blomquist, M. Armishaw, D. Hanlon, N. Smith, Y. Naito, J. Yang, Y. Mioshi, T. Yamamoto, O. Jacquet, and J. MISS, "Source Convergence in Criticality Safety Analyses," *NEA Report No. 5431*, Nuclear Energy Agency, Organisation for Economic Co-operation and Development (Nov, 2006).

[8] T. Ueki, "Information Theory and Undersampling Diagnostics for Monte Carlo Simulation of Nuclear Criticality", *Nuclear Science and Engineering*, **151**, 283-292 (2005).

[9] T. Ueki, "On-the-Fly Diagnostics of Particle Population in Iterated-Source Monte Carlo Methods", *Nuclear Science and Engineering*, **158**, 15-27 (2008).

[10] E. W. Larsen and J. Yang, "A Functional Monte Carlo Method for k-Eigenvalue Problems" *Nuclear Science and Engineering*, **159**, 107-126 (2008).

[11] G.E. Whitesides, "A difficulty in computing the k-effective of the World," *Transactions of the American Nuclear Society*, **14**, 680 (1971).

[12] R. N. Blomquist, etc. "Source Convergence in Criticality Safety Analysis," *NEA Report ISBN 92-64-02304-6* (2006).

[13] D. B. MacMillan, "Monte Carlo Confidence Limits for Iterated-Source Calculations", *Nuclear Science and Engineering*, **50**, 73-87 (1973).

[14] E. M. Gelbard and R. Prael, "Computation of Standard Deviations in Eigenvalue Calculations", *Progress in Nuclear Energy*, **24**, 237 (1990).

[15] T. Ueki, T. Mori, and M. Nakagawa, "Error Estimations and Their Biases in Monte Carlo Eigenvalue Calculations", *Nuclear Science and Engineering*, **125**, 1-11 (1997).

[16] H. J. Shim and C. H. Kim, "Real Variance Estimation Using an Intercycle Fission Source Correlation for Monte Carlo Eigenvalue Calculations", *Nuclear Science and Engineering*, **162**, 98–108 (2009).

[17] T. Ueki and B. R. Nease, "Time Series Analysis of Monte Carlo Fission Sources—II: Confidence Interval Estimation", *Nuclear Science and Engineering*, **153**, 184-191 (2006).

[18] R. A. Forster, S. P. Pederson, and T. E. Booth, "Ten new checks to assess the statistical quality of Monte Carlo solutions in MCNP," *8th International Conference on Radiation Shielding* (April 1994).

[19] B. Petrovic, "Numerical Demonstration of Source Convergence Issues in Monte Carlo Eigenvalue Simulations," *Transactions of the American Nuclear Society*, **84**, 173-175 (2001).

[20] T. Ueki and F. B. Brown, "Stationarity Modeling and Informatics-Based Diagnostics in Monte Carlo Criticality Calculations," *Nuclear Science and Engineering*, **149**, 38–50 (2005).

[21] X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-particle Transport Code, Version 5," *Los Alamos National Laboratory*, (2003)

[22] T. Ueki, "Stationarity Diagnostics with Relative Entropy and Wilcoxon Signed Rank in Iterated-Source Monte Carlo Methods," *Nuclear Science and Engineering*, **160**, 242–252 (2008).

[23] P. K. Romano, "Application of the Stochastic Oscillator to Assess Source Convergence in Monte Carlo Criticality Calculations," *International Conference on Mathematics, Computational Methods & Reactor Physics* (2009).

[24] E. Dumonteil etc., "Source convergence diagnostics using Boltzmann entropy criterion Application to different OECD/NEA criticality benchmarks with the 3-D Monte Carlo code Tripoli-4," *International Conference on the Physics of Reactors* (2006).

[25] M. T, Wenner and A. Haghighat, "A Combined Diagnostic Approach for Monte Carlo Source Convergence Identification," *International Conference on Mathematics, Computational Methods & Reactor Physics* (2009).

[26] H. J. Shim and C. H. Kim, "Stopping Criteria of Inactive Cycle Monte Carlo Calculations," *Nuclear Science and Engineering*, **157**, 132-141 (2007).

[27] T. Yamamoto and Y. Miyoshi, "Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt's Method," *Journal of Nuclear Science and Technology*, **Vol. 41**, No. 2, 99-107 (2004).

[28] F. Brown, "Wielandt Acceleration for MCNP5 Monte Carlo Eigenvalue Calculations," *Joint International Topical Meeting on Mathematics Computation and Supercomputing in Nuclear Applications* (2007).

[29] T. Yamamoto, T, Nakamura and Y. Miyoshi, "Fission Source Convergence of Monte Carlo Criticality Calculations in Weakly Coupled Fissile Arrays," *Journal of Nuclear Science and Technology*, **Vol. 37**, No. 1, 41-52 (2000).

[30] T. Kuroishi and Y. Nomura, "Development of Fission Source Acceleration Method for Slow Convergence in Criticality Analyses by Using Matrix Eigenvector Applicable to Spent Fuel Transport Cask with Axial Burnup Profile," *Journal of Nuclear Science and Technology*, **Vol. 40**, No. 6, 433-440 (2003).

[31] S. Yun and N. Z. Cho, "Monte Carlo Anchoring Method for Loosely-Coupled k-Eigenvalue Problems," *International Conference on Mathematics, Computational Methods & Reactor Physics* (2009).

[32] D. P. Griesheimer and B. E. Toth, "A Novel Source Convergence Acceleration Scheme for Monte Carlo Criticality Calculations, Part I: Theory," *Joint International Topical Meeting on Mathematics Computation and Supercomputing in Nuclear Applications* (2007).

[33] R. Blomquist, A. Nouri, M. Armishaw, O. Jacquet, Y. Naito, and T. Yamamoto, "OECD/NEA Source Convergence Benchmark Program: Overview and Summary of Results," *Proc. 7th Intl. Conf. on Nuclear Criticality Safety*, **Vol I**, 278-282 (2003).

[34] NEA Expert Group on Source Convergence in Criticality-Safety Analysis, internet source from http://www.nea.fr/html/science/wpncs/convergence/specifications/index.html.

[35] Taro Ueki and Forrest B. Brown, "Informatics Approach to Stationarity Diagnostics of the Monte Carlo Fission Source Distribution,'' *Transactions of the American Nuclear Society*, **89**, 458 (2003).

[36] Y. Naito and J. Yang, "The Sandwich Method for Determining Source Convergence in Monte Carlo Calculation," *Journal of Nuclear Science and Technology*, **Vol. 41**, No. 5, p. 559–568 (2004).

[37] T. E. Booth, "Computing the Higher k-Eigenfunctions by Monte Carlo Power Iteration: A Conjecture," *Nuclear Science and Engineering*, **143**, 291–300 (2003).

[38] T. E. Booth, "Power Iteration Method for the Several Larg,est Eigenvalues and Eigenfunctions," *Nuclear Science and Engineering*, **154**, 48–62 (2006).

[39] T. Yamamoto, "Convergence of the Second Eigenfunction in Monte Carlo Power Iteration," A*nnals of Nuclear Energy*, **36**, 7-14 (2009).

[40] T. E. Booth and J. E. Gubernatis, "Improved Criticality Convergence via a Modified Monte Carlo Power Iteration Method," *International Conference on Mathematics, Computational Methods & Reactor Physics*, (2009).

[41] J.E. Gubernatis and T.E. Booth, "Multiple Extremal Eigenpairs by the Power Method," *Journal of Computational Physics*, **227**, 8508–8522 (2008).

[42] James J. Duderstadt and L.J. Hamilton, *Nuclear Reactor Analysis*, John Wiley & Sons, (1976).

[43] Taro Ueki, Forrest B. Brown, D. Kent Parsons and Drew E. Kornreich. "Autocorrelation and Dominance Ratio in Monte Carlo Criticality Calculations," *Nuclear Science and Engineering*, **145**, 279–290 (2003).