DEVELOPMENT OF A MULTIPLE PERTURBATION MONTE CARLO METHOD FOR EIGENVALUE PROBLEMS AND IMPLEMENTATION ON PARALLEL PROCESSORS

by

Amitava Majumdar

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Scientific Computing)
in The University of Michigan
1996

Doctoral Committee:

Professor William R. Martin, Chairperson
Assistant Professor James P. Holloway
Professor Edward W. Larsen
Professor John C. Lee
Professor Trevor N. Mudge

To my parents Bimal and Gouri Majumdar

ACKNOWLEDGEMENTS

I would like to thank Prof. William Martin for all the support, encouragement and knowledge that I have received from him all these years at The University of Michigan. Prof. Martin has given me freedom to pursue research (in my own little way) when I was in the right direction; he has stepped in and put me back to the right path when I was following the wrong directions. This combination of freedom and guidance is the very best support a graduate student can ever expect from his or her advisor. I will be forever grateful to Prof. Martin for everything that he has done for me.

A special thanks to Prof. John Lee. While working under him on different projects, I have learned how to do research and this experience will be most valuable to me throughout my working life. I would also like to thank Prof. Edward Larsen. Everything I know about numerical transport theory, I have learned from his excellent lectures. Thank you Prof. Larsen for giving me this rare opportunity. I thank Prof. James Holloway for always answering any question I had about academic and non-academic matters. Thanks to Prof. Trevor Mudge for being in my doctoral committee. It is a rare privilege to learn from a teacher like Prof. Ziya Akcasu - thank you Prof. Akcasu.

I am most grateful to my parents for all their support. Without their encouragement this thesis would not have existed. It is impossible to express in mere words my feelings for them. Everything that I have accomplished has been possible because of

them.

I am grateful to my aunt Catherine Majumdar and uncle Debu Majumdar. They have encouraged and guided me from the very first day of my graduate student life. Without their support I would not have any graduate degree.

A very special thanks to Mieko Ueno for supporting me in her own quite way all these stressful years. She is the only person in front of whom I did not hide my stresses and tensions. Thank you for standing by me. I will be there when you receive your Ph.D.

I would like to thank late Prof. Harry Charyulu of Idaho State University. His guidance was instrumental in my graduate studies. Also thanks to Dr. Henry Makowitz and Dr. Michael Lineberry of Idaho National Engineering Laboratory; and Prof. D. T. Neil, Prof. Al. Wilson and Prof. Al Stephens of Idaho State University.

Thanks to Dr. Hal Marshall who has taught me about parallel computers and has always answered my questions. Thanks to Dr. Dave Woodcock for also helping me. I would like to thank the Center for Parallel Computing for letting me use and learn parallel computing on different parallel machines.

Thanks to several friends who have played with me, laughed with me and have taught me all the bad and good things about life. Without them life would not be fun. I would like to thank Todd Urbatsch, Djordje Tomasevic, Todd and Stephanie Wareing, Moaied Miften, Yuji Fujii and Yuni Dewaraja, Tarek Al-Nahlawi, Todd and Teri Palmer, Ayman Hawari, Siaka Yusuf, Eric Smith, Chris and Michaelle Gesh, Scott and Anna Turner, Kasem Abotel, Carol Tallman, Eric Hawkes, Peggy Jo Christenson, Marie-Anne Descalle, Hongjuan Wang, Denise Parevanti, Joe Schumer, Bob Grove, Paul Keller, Taro Ueki, Jun Miyamoto, Danny Ray Tolar, Brian Guthrie,

Marc Cooper, Patricia Lopez, Ricardo de Barros, Robert Rulko, Musa Yavuz, Helen Lum, Sue Greenwood, Karen Balson, Ann Bell, Alice Cook, Diana Corey, Pam Derry and Mariann Marshall from The University of Michigan; Masanao Moriwaki and Natsuko Kodama, Norio Suzuki, Takaaki Mochida, and Konosuke Onishi from Japan; Kambiz Khosro-shahroudi, Bill Rhodes, Cindy Charyulu, PJ, Kuang Yi Chen, Xia Fang Qi and Daisy Wang from Idaho State University; Krishna Thakur, Subhasis Kabiraj, Ballal Chatterjee, Prabal Chatterjee, Pranabendu Chakrabarty, Tamal Ghosh and Shaileswar Bhattacharyaa from Calcutta; Debashasis Roy, Sarajit RoyChoudhury, Ashim Pal, Ashoke Chakraborty, Pintu Mukherjee and Sharmistha Ganguly from Durgapur; my junior high school teacher from Durgapur, Mr. Sanyal for teaching me English; my cousins Debi, Mou, Raj, Minakshi, Rajeev, Nikhil, Sharmistha, Sujata, Tata, and Krishna-da.

Watching sports outside the Nuclear Engineering Department has made my life as interesting as doing research and learning inside the department. My thanks to Fav-Five, Desmond Howard, Walter Smith, ex-Piston Isiah Thomas (for shaking my hand and waving me goodbye afterwards at the Detroit Metro airport on Jan, 5, 1996 around 2:30 PM), and all the male and female athletes and coaches of The University of Michigan.

Go blue and *peace*.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

LIST OF APPENDICES

Appendix

CHAPTER I

INTRODUCTION

1.1 Numerical Solution of the Boltzmann Transport Equation

The Boltzmann transport equation describes neutral and charged particle transport phenomena. Numerical solution of the Boltzmann transport equation finds application in different fields such as nuclear reactor design, radiation shielding calculations, radiative transfer in stellar atmospheres, semiconductor device design, radiation oncology, and high energy physics, to name a few. There are two classes of computational techniques that are used to solve the transport equation. In the first class, deterministic methods, the transport equation is discretized using a variety of methods and then solved directly or iteratively. Different types of discretization give rise to different deterministic methods [Lew93, Dud79], such as discrete ordinates (S_N), spherical harmonics (P_N), collision probabilities, nodal methods, and others. The second class of techniques, Monte Carlo methods, constructs a stochastic model in which the expected value of a certain random variable is equivalent to the value of a physical quantity to be determined [Car75, Ham64, Lux91, Rub81, Spa69]. The expected value is estimated by the average of many independent samples representing the random variable. Random numbers, following the distributions of the variable

to be estimated, are used to construct these independent samples. There are two different ways to construct a stochastic model for Monte Carlo calculations. In the first case the physical process is stochastic and the Monte Carlo calculation involves a computational simulation of the real physical process. In the other case, a stochastic model is constructed artificially, such as the solution of deterministic equations by Monte Carlo.

Both deterministic and Monte Carlo methods have errors, but the source of errors is different for each method. In the treatment of deterministic computational methods the computing errors are systematic. They arise from the discretization of the time-space-angle-energy phase space and the approximate geometry. The present state of the art does not allow full representation of complicated three-dimensional geometries for deterministic transport methods. Monte Carlo methods, on the other hand, can treat continuous energy, space, and angle, and hence avoid discretization errors. The errors in Monte Carlo methods take the form of stochastic uncertainties. Estimation of the statistical uncertainty of Monte Carlo results requires understanding of properties of random variables such as expectation values, variance, and the central limit theorem. Deterministic methods are computationally fast but may sacrifice accuracy; whereas Monte Carlo methods are computationally slow yet arbitrarily accurate.

We conclude these introductory remarks on the numerical solution of the Boltzmann transport equation, by noting that the rest of this chapter is divided into following sections: in section 1.2 we explain the motivation and objective for this research work. In section 1.3 we introduce the ideas of Monte Carlo eigenvalue calculation, which are vital to Monte Carlo eigenvalue perturbation calculations. In section 1.4 different Monte Carlo perturbation methods for the calculation of re-

action rates and eigenvalues are introduced. Section 1.5 introduces parallel Monte Carlo algorithms. Section 1.6, the last section of this chapter, gives an outline of the remainder of this dissertation.

1.2 Objective and Motivation for this Work

The main objective of this research work is to develop a multiple perturbation Monte Carlo technique for nuclear reactor criticality problems. Even though the eigenvalue (or the multiplication factor) of the neutron transport equation can be estimated very efficiently by Monte Carlo methods, the calculation of small reactivity effects due to realistic cross section perturbations is much more difficult. For small perturbations, a direct correlated simulation is necessary instead of taking the difference between the results of two independent Monte Carlo simulations. The ability to calculate multiple eigenvalue perturbations from a single Monte Carlo simulation is extremely useful. Since Monte Carlo methods are computationally slow, this capability gives rise to an efficient Monte Carlo perturbation technique. There are quite a few cases in computational reactor physics for which it is desired to know the perturbations in eigenvalue due to cross section changes. These include calculations of perturbed eigenvalues due to different soluble boron concentrations, different number of absorber control rods in assemblies, and different assembly loading patterns for a global core, for example.

1.3 Introduction to Monte Carlo Eigenvalue Calculation

The multiplication factor (K) is defined as the dominant eigenvalue of the neutron transport equation. The eigenvalue K can also be expressed as the ratio of the number of neutrons in one generation to the number in the previous generation in

a system containing fissionable material and in the absence of any external source. Monte Carlo eigenvalue simulation starts with hundreds or thousands of neutrons and follows these neutrons through many generations [Goa59, Lie68, Mih67, Whi66]. These neutron generations are referred to as batches, cycles, or stages. Within each batch, a random walk simulation, from birth to death, of all the neutrons is done. The simulation of a neutron lifetime from birth by fission, to death by capture or leakage, is called a "history". Appropriate probability density functions, representing physical processes such as fission, scattering, transmission, leakage etc., are sampled to determine the different states of a random walk simulation. If fission occurs, the site of the fission and the number of neutrons produced at that site are stored for the next batch. After the simulation of all the histories for a particular batch is completed, the fission production is normalized to maintain the original number of starting fission neutrons. Most codes require that the starting number of fission neutrons be approximately the same for all generations. This prevents population extinction if $K < 1$ and memory overflow if $K > 1$, and it makes the coding simpler.

One problem associated with Monte Carlo eigenvalue calculations is that the spatial source distribution for the first batch is not known *a priori* [Men68]. The usual procedure for handling this situation is to start with some arbitrary source distribution for the first batch, and then for each subsequent batch, select source sites based on the fission neutron production distribution obtained from the previous batch. This iterative procedure requires that a number of initial batches be discarded to eliminate the effect of the arbitrary starting distribution. It is important that enough initial batches be discarded so that the fundamental mode source distribution is reached before batch eigenvalue estimates are accumulated [Whi71].

In a Monte Carlo eigenvalue calculation there is a tradeoff between the number

of batches versus the number of neutrons per batch. It is difficult to determine what combinations of batches and histories per batch will provide the most accurate result for a fixed number of neutrons (i.e., neutrons per batch times the number of batches). If a very large number of neutrons per batch is used for an eigenvalue calculation, then the variance in the eigenvalue estimator for each batch will be small. However, the fundamental mode distribution of fission neutrons may not be reached due to the relatively fewer number of batches used. The other alternative is to follow a small number of neutron histories for many batches. This will allow the neutron source distribution to reach the fundamental mode, but a large variance will arise in the fission production estimator for each batch.

Bias is another important issue involved with Monte Carlo eigenvalue calculations [Bow83, Eno90, Gel74, Gel91, Zol83]. It is known that the calculation of K using Monte Carlo batches produces biased results. This is due to the need to generate and maintain the fundamental mode eigenfunction, which is usually achieved by a batch-to-batch settling process involving some kind of normalization of the neutron population at the end of each batch. With some experience in Monte Carlo eigenvalue computation and using the relationship [Bri86, Gel94] between bias and standard deviation as guidance, the effect of bias can be made insignificant for most cases. The estimators for the variance of the eigenvalue over the batches can also be biased [Gel81, Gel90a, Mac73, Moo76, Gas75], especially for systems with high dominance ratios, due to the correlations between the neutron histories from one batch to another.

1.4 Introduction to Monte Carlo Perturbation Calculation

Small changes in a system are called *perturbations*. In a straightforward perturbation computation, two independent simulations are performed and the difference of the two results are calculated. This type of approach, using Monte Carlo methods, can be inefficient and even impossible to solve in some cases. For small perturbations, the relative error of the difference of two independent simulations can be much larger than the relative error of the unperturbed or perturbed quantity [Spa69]. Hence, it is necessary to employ special Monte Carlo techniques that calculate the difference of two responses, independent of their statistical uncertainties. These Monte Carlo perturbation techniques include correlated tracking, derivative operator sampling, the importance function approach [Rie86, Lux91] and the linear perturbation theory [Bel70] approach. The linear perturbation theory requires information about the angular neutron flux and hence has not been studied in this work. These other three methods employ different estimators and their uncertainties may differ considerably, depending on the problem tested. The first two methods may be applied to reaction rate perturbation calculations as well as eigenvalue perturbation calculations. The importance function approach is mainly used for eigenvalue perturbation calculations. The usual source iteration method for Monte Carlo eigenvalue calculations encounters difficulty, due to the propagation of perturbed weights from one generation to the next [Rie84, Spa69]. To avoid this difficulty, an adjoint function must be used, and this gives rise to the importance function approach for eigenvalue perturbation calculations. For eigenvalue perturbation calculations, using correlated and derivative operator sampling, these two methods are applied to the Green's function or fission matrix approach for eigenvalue calculations. In the fission matrix

approach, the homogeneous neutron transport equation is first discretized by subdividing the fissionable region into a mesh of volume elements. Next, a random walk simulation is done to estimate the elements of the fission matrix that contains the mutual fission probabilities for these volume elements. The dominant eigenvalue of this fission matrix gives the multiplication factor of the system. In this thesis work we have combined the correlated sampling technique and the fission matrix method to calculate multiple eigenvalue perturbations from a single Monte Carlo simulation. Below, we briefly explain the correlated tracking, derivative operator sampling, and the importance function approach.

1.4.1 Correlated Sampling

In a straightforward Monte Carlo approach, where a perturbation is evaluated by taking difference of two independent simulations, the relative variance of the difference tends to infinity as the perturbation tends to zero. To address this, the correlated sampling technique [Rie84, Blo83] forces both the perturbed and unperturbed histories to follow the same transition points in phase space. Using the known transition and collision kernels, the actual simulation can be done in either the unperturbed or the perturbed system. A correlated simulation is done by modifying the appropriate weight factors of the other system. This can be understood as if the unperturbed and perturbed particles are migrating in parallel along the same trajectories. This technique forces the responses of the perturbed and unperturbed histories to be strongly correlated. As a result, their difference is expected to have a smaller uncertainty than the corresponding difference in the uncorrelated game. It can be shown that for sufficiently small perturbations, this method leads to finite relative variance of the differential effect.

For the calculation of eigenvalue perturbations, along with the unperturbed (or perturbed) fission matrix, another fission matrix for the perturbed (or unperturbed) system is generated from the correlated histories reacting with the perturbed (or unperturbed) system. The difference of the dominant eigenvalues of these two fission matrices gives the required ΔK.

The source of error in this approach is due to the modifications of the appropriate weight factors. These modifications force particles in one system to simulate the physics that is consistent with the other system. Modification of weight factors are needed to account for the distance to collision, scattering event, fission reaction etc.

1.4.2 Derivative Operator Sampling

In nuclear design it is important to know the effect of uncertainties in key parameters on reaction rates. This gives the sensitivity of reaction rates to small changes (or perturbations) to these key parameters. For complex systems like nuclear reactors, it is often necessary to resort to Monte Carlo methods for this kind of sensitivity study. The uncertainty in a reaction rate, due to small changes in a system parameter, can be defined as the derivative of the reaction rate w.r.t. the system parameter. Correlated simulation estimates the change due to a given variation of parameter(s), whereas a differential Monte Carlo simulation estimates the change due to arbitrary (but small) variations in the system parameters. Hence, the derivatives are characteristic of the sensitivity of the reaction rate to variations in the parameters. In the case of several parameters, multivariate Taylor series applies and the partial derivatives of the reaction rate w.r.t. the various parameters are estimated.

For eigenvalue perturbation calculations, the scores of first and second order derivatives of the unperturbed system are stored in separate matrices. After the

fission matrix of the unperturbed system is completed, the fission matrix of the perturbed system is calculated by a Taylor series approximation for the perturbation of the system.

An unbiased procedure of estimating the first derivative, for reactivity changes due to small variations in system parameters, was first proposed by Mikhailov [Mik67] and independently by Miller [Mil67] and Takahashi [Tak70]. A constructive derivation of a multiparameter second-order derivative estimation procedure was shown by Hall [Hal80, Hal82].

Both the correlated sampling and derivative operator sampling have the advantage that they require little additional computing effort to calculate the perturbation effects. No calculation for importance parameters are required, as is the case for the importance function approach, which is described next.

1.4.3 Importance Function Approach

This method is utilized to calculate perturbations in the reactivity of a multiplying system due to changes in cross sections. This formulation makes full use of the properties of the importance (adjoint) function [Mat72, Hof72a, Hof78]. Development of this perturbation theory is started from two equations, proposed by Ussachoff [Uss55], representing the homogeneous transport equation for the unperturbed system and the corresponding adjoint equation for the perturbed system. From these two equations an expression for ΔK is derived in terms of the importance function of a fission neutron, the fundamental mode fission neutron production function, and the fission kernel. These three kernels are then expressed in terms of the Green's functions for the nonmultiplying systems. Application of the difference flux concept [Bra70, Hof72b], to the Boltzmann transport equations satisfied by these

Green's function, allows the expression of ΔK in terms of the unperturbed and perturbed fission cross sections, the angular flux generated in the unperturbed system by the fission source, and the angular flux generated in the perturbed system by the perturbation source [Mat72].

A Monte Carlo simulation starts by guessing the initial fission source distribution in the unperturbed system and the importance function in the perturbed system. Simulations of the fission neutrons in the unperturbed system gives the angular flux in the unperturbed system, the perturbation source, an uncorrected importance function, and the fission source distribution for the next generation. Simulation of the perturbed source particles gives the angular flux in the perturbed system. After several of these iterations, the fundamental mode eigenfunction is achieved, and estimations for ΔK can be accumulated.

1.5 Monte Carlo Particle Transport on Parallel Computers

Monte Carlo particle transport methods are extensively used, due to the generality and accuracy of these methods. In Monte Carlo methods hundreds of thousands or millions of particle histories are simulated using random numbers, highly accurate representations of particle reaction probabilities and exact models for 3-D geometries. The principal limitations for Monte Carlo methods are the requirement to simulate many particles to achieve an acceptable statistical uncertainty. This requirement provides ample incentive to utilize the computational power of modern vector and parallel supercomputers. Monte Carlo particle transport algorithms are inherently parallel because each particle history can be simulated independently and concurrently on separate processors. Monte Carlo particle transport codes have been successfully implemented on a number of different computational platforms [Mar91].

For adaptation to vector computers, the computational algorithm should be changed from a history-based scheme to an event-based scheme [Bro81, Bro85, Bro86, Cha85, Mar86, Mar87a]. Even though this requires restructuring of all data and extensive recoding, successful vectorized codes have provided gains of 10 - 100X or greater in computational speed. Moreover, once vectorized, it is relatively easy to parallelize a Monte Carlo code across multiple vector processors [Bob84]. Many research organizations have resisted the vectorization of production Monte Carlo codes, due to the considerable investments in time and manpower involved. On the other hand, the relative ease of parallelization and continuing decrease in costs of parallel computers, both massively parallel processors (MPPs) and distributed workstations, makes it attractive to adapt Monte Carlo codes to parallel computers [Mar87b, Mar93]. Monte Carlo algorithms are ideally suited for distributed and shared memory MIMD (Multiple Instruction, Multiple Data) parallel processors, because of the inherent parallelism involved in the fundamental algorithm. If enough memory is available to each processor, the Monte Carlo code and data can be replicated, and each processor can independently follow a portion of the total particles. This is possible due to the statistical independence of the particle histories. Monte Carlo particle transport codes have been demonstrated to be efficient and effective on both MPPs and distributed workstations. Two important issues involving parallel random number generators and reproducibility of results on parallel computers will be discussed in a later chapter of this thesis. We will also discuss our effort to implement various Monte Carlo algorithms on the two parallel computers (KSR-1 and IBM-SP2) operated by the Center for Parallel Computing at the University of Michigan and on the BBN TC2000 at the Lawrence Livermore National Laboratory.

1.6 Outline of Remaining Chapters

This section summarizes the contents of the remainder chapters of this dissertation, which includes an additional five chapters, two appendices, and a bibliography.

In chapter II we derive equations for Monte Carlo eigenvalue calculation methods. It is important to understand the basic Monte Carlo eigenvalue algorithms before one studies Monte Carlo eigenvalue perturbation methods. Both the source iteration technique and the fission matrix approach are described in this chapter. All the statistical quantities associated with Monte Carlo eigenvalue calculation are also presented. Numerical results are shown for both the source iteration and the fission matrix methods. We also investigate two variations of the fission matrix algorithm, the *cycle fission matrix* algorithm and the *cumulative fission matrix* algorithm.

Chapter III explains in detail the correlated sampling technique. We show that subtracting two independent Monte Carlo simulations to evaluate perturbation effects is problematical for realistic problems. Next, the mathematical basis behind the correlated sampling technique is established. We also introduce the idea of performing the Monte Carlo simulation in an artificial reference system which is different from both the unperturbed and perturbed systems. We show that combining the correlated sampling and source iteration techniques to calculate perturbation effects fails, whereas combining the correlated sampling and fission matrix techniques can successfully evaluate perturbation effects in eigenvalue. Numerical results are shown to support all the theories developed in this chapter.

Chapter IV explains how the ideas of the fission matrix method, correlated sampling, and the artificial reference system generated in the last two chapters can be combined to develop a *multiple perturbation* Monte Carlo method. This method al-

lows the calculation of multiple perturbations of the eigenvalue of the Boltzmann transport equation from a single Monte Carlo simulation, and results in significant savings in overall computational effort. The extension of this method to the multigroup case is also explained. Numerical examples are provided to show the accuracy and efficiency of this multiple perturbation method.

Chapter V deals with the implementation of particle transport Monte Carlo algorithms on parallel computers. We explain the basic concepts behind Monte Carlo parallel algorithms used to simulate particle transport. Quantities such as speedup, efficiency, parallel and serial fractions, and communication times are defined specifically for Monte Carlo parallel algorithms. Results of implementing a photon transport algorithm on the KSR-1 and BBN Butterfly computers, and three Monte Carlo eigenvalue algorithms (source iteration, fission matrix and correlated sampling) on the IBM-SP2 are shown.

Chapter VI contains the conclusions of this research and recommendations for future work. We explain what has been achieved in this dissertation research. We summarize the different theories that lead to the development of the numerical technique for multiple eigenvalue perturbation calculations. Areas that need more study and could lead to further research topics are also addressed.

In appendix A we perform a computational parametric study on the choice of the artificial reference system different from both the unperturbed and perturbed systems. In this dissertation work, the cross sections of the artificial reference system were chosen as the average of the cross sections for the unperturbed and of all perturbed systems. This parametric study of appendix A is based only on computational experiments. We believe that this study requires more theoretical investigation, which we leave as one of the future extensions of this research.

In appendix B we perform accuracy test for the TWODANT code. We vary the mesh size and the quadrature sets for a given problem to determine the accuracy of TWODANT eigenvalue results. These results are used in determining the number of significant figures necessary in TWODANT calculated ΔKs against which the Monte Carlo results are compared.

CHAPTER II

MONTE CARLO EIGENVALUE CALCULATION

2.1 Introduction

This chapter explains how Monte Carlo techniques can be used to calculate the multiplication factor K. The theories and the resulting algorithms, described in this chapter, are the basic building blocks of the later chapters, in which we investigate Monte Carlo eigenvalue perturbation methods. The multiplication factor can be defined in various ways, such as: the dominant eigenvalue of the neutron transport equation, the quantity by which $\bar{\nu}$ (the average number of neutrons per fission) must be divided to keep a non-critical system exactly critical, and the ratio between the number of neutrons in successive generations. A *generation* can be defined as the life of a neutron from birth by fission to death by leakage or absorption (both capture and fission). For critical systems $K = 1$, for subcritical systems $K < 1$, and for supercritical systems $K > 1$.

This chapter consists of four more sections. In the second section we give a mathematical basis for eigenvalue calculations. The third section describes how this mathematical basis can be transformed into Monte Carlo algorithms for the eigenvalue calculation. In the fourth section, we describe various numerical results.

We conclude in the fifth section with some numerical experiments to investigate the effect of two variations of the fission matrix algorithm.

2.2 Eigenvalue Calculations - Mathematical Basis

The time-independent Boltzmann transport equation for neutrons is an equation for the neutron angular density. This equation is derived by applying neutron conservation to an infinitesimal element of volume, direction and energy [Bell70, Lew93]. The integro-differential form of the time-independent transport equation in a medium with volume V surrounded by a surface S, as shown in figure 2.1, can be written as,

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

$$\vec{r} \in V, \quad 0 < E < \infty, \quad |\vec{\Omega}| = 1. \tag{2.1}$$

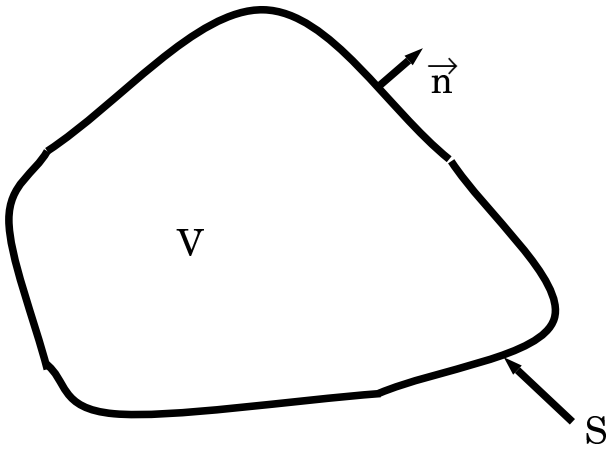


Figure 2.1: Spatial Domain V for the Transport Equation with Surface S.

The boundary condition, if $\vec{n}$ is an outward normal to S, is given as,

$$\psi(\vec{r}, \vec{\Omega}, E) = \psi^b(\vec{r}, \vec{\Omega}, E), \quad \vec{\Omega} \cdot \vec{n} < 0, \quad \vec{r} \in S. \tag{2.2}$$

In the transport equation, $\vec{r}$ denotes spatial variables, E denotes energy, $\vec{\Omega}$ denotes

angular variables, $\psi(\vec{r},\vec{\Omega},E)$ is the angular flux, $\Sigma_t(\vec{r},E)$ is the macroscopic total cross section, and $Q(\vec{r},\vec{\Omega},E)$ is the source emission density.

In general, the source emission density $Q(\vec{r},\vec{\Omega},E)$ consists of three contributors:

$$Q = Q_{ex} + Q_s + Q_f. \tag{2.3}$$

The above three contributors are due to the external source, scattered particles, and fission neutrons, respectively.

For *nonmultiplying* systems $Q_f = 0$, and the two contributions to the source emission density are due to the external source and the scattered particles. Hence the transport equation in a nonmultiplying medium is given by,

$$\vec{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\vec{\Omega},E) + \Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E) =$$

$$Q_{ex}(\vec{r},\vec{\Omega},E) + \int dE' \int d\Omega' \Sigma_s(\vec{r},E'\rightarrow E,\vec{\Omega}'\cdot\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}'), \tag{2.4}$$

where $\Sigma_s(\vec{r},E'\rightarrow E,\vec{\Omega}'\cdot\vec{\Omega})$ is the macroscopic scattering cross section.

For *multiplying* media, the contribution Q_f due to the neutrons emitted from fission reactions must be included. We will assume that all neutrons are produced instantaneously at the time of fission and neglect the effect of delayed neutrons. This is a valid assumption, except for time dependent problems of neutron kinetics. This assumption is appropriate for steady state problems with fixed sources and for criticality calculations, where only the critical state and the flux distribution are of interest. The transport equation in a multiplying medium without delayed neutrons takes the following form:

$$\vec{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\vec{\Omega},E) + \Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E) = Q_{ex}(\vec{r},\vec{\Omega},E) +$$

$$\int dE' \int d\Omega' \Sigma_s(\vec{r},E'\rightarrow E,\vec{\Omega}'\cdot\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}') + \chi(E)\int dE'\nu\Sigma_f(\vec{r},E')\int d\Omega'\psi(\vec{r},E',\vec{\Omega}'). \tag{2.5}$$

Here, $\nu(E)$ is the mean number of fission neutrons produced in a fission caused by a neutron with energy E. The effect of delayed neutrons is included in the total $\bar{\nu}$. Also $\Sigma_f(\vec{r}, E)$ is the *macroscopic fission cross section* and $\chi(E)$ denotes the *fission spectrum.* The transport equation is a hyperbolic equation in both the time dependent and steady state forms. In a nonmultiplying medium with a steady state source, there exists a steady state flux distribution that satisfies this equation [Cas67]. In multiplying systems, the transport equation is studied by introducing the concept of criticality. Physically, a system containing fissionable material is critical if a self-sustaining time independent chain reaction in the absence of external neutron sources can be maintained. Mathematically, a system is critical if it has a time-independent, nonnegative solution to the transport equation without external source, i.e.

$$\vec{\Omega} \cdot \vec{\nabla}\psi(\vec{r}, \vec{\Omega}, E) + \Sigma_t(\vec{r}, E)\psi(\vec{r}, \vec{\Omega}, E) =$$

$$\int dE' \int d\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \cdot \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}') + \chi(E) \int dE' \nu\Sigma_f(\vec{r}, E') \int d\Omega' \psi(\vec{r}, E', \vec{\Omega}'). \tag{2.6}$$

In general, it is difficult to find that combination of cross sections and geometry that will allow equation (2.6) to be satisfied. Therefore, for criticality calculations, the above equation is cast into the form of an eigenvalue problem. The eigenvalue provides a measure of the criticality of the system. Two formulations for criticality are the *time-absorption* or *α-eigenvalue* formulation and *the multiplication factor formulation* with the K eigenvalue [Bell70]. The K eigenvalue approach is discussed in this thesis.

We have stated before that K can be defined as the quantity by which ν must be divided to keep a system critical, i.e., ν can be adjusted to obtain a time independent solution to equation (2.6). Hence for the K eigenvalue problem ν is replaced by ν/K

and the transport equation becomes,

$$\vec{\Omega} \cdot \vec{\nabla}\psi(\vec{r},\vec{\Omega},E) + \Sigma_t(\vec{r},E)\psi(\vec{r},\vec{\Omega},E) =$$

$$\int dE' \int d\Omega' \Sigma_s(\vec{r},E' \rightarrow E, \vec{\Omega}'\cdot\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}') + \frac{\chi(E)}{K} \int dE' \nu\Sigma_f(\vec{r},E') \int d\Omega' \psi(\vec{r},E',\vec{\Omega}'). \tag{2.7}$$

In an eigenvalue problem it is always possible to find the largest value of K that will give a nonnegative *fundamental mode* solution to equation 2.7. The system is critical if this largest value of K is equal to unity. If K $<$ 1, then ν, the actual number of neutrons per fission available, is less than ν/K, the number of neutrons per fission necessary to maintain criticality, and the system is subcritical. Similarly
for K $>$ 1, the actual number of neutrons per fission, ν, is more than ν/K, the number of neutrons per fission required to maintain the system critical, and hence the system is supercritical.

Equation (2.7) is the integro-differential form of the transport equation. This equation can also be expressed in integral form using the Green's function. The Green's function $G(\vec{r}',\vec{\Omega}',E' \rightarrow \vec{r},\vec{\Omega},E)$ is the neutron angular flux at $\vec{r},\vec{\Omega},E$ due to a unit point source at $\vec{r}'$ emitting 1 neutron/sec in the direction $\vec{\Omega}'$ with energy E'. Using the Green's function we can write down the K eigenvalue equation as follows:

$$\psi(\vec{r},\vec{\Omega},E) =$$

$$\int\int\int d\vec{r}' d\vec{\Omega}' dE' G(\vec{r}',\vec{\Omega}',E' \rightarrow \vec{r},\vec{\Omega},E) \frac{1}{K}\chi(E') \int\int dE'' d\vec{\Omega}'' \nu\Sigma_f(\vec{r}',E'')\psi(\vec{r}',\vec{\Omega}'',E''). \tag{2.8}$$

Both equation (2.7) and (2.8) can be expressed in terms of the fission source density, $\int\int \nu\Sigma_f(\vec{r},E')\psi(\vec{r},\vec{\Omega}',E')d\Omega' dE'$, which gives the spatial distribution of fission neutrons. From that form, both equations lead to the same matrix eigenvalue problem,

to which the power iteration method can be applied to calculate K, as explained below.

For the time-independent, nonmultiplying transport equation given in equation (2.4), the transport operator is defined as [Lew93]:

$$H\psi = [\vec{\Omega} \cdot \vec{\nabla} + \Sigma_t(\vec{r}, E)]\psi(\vec{r}, \vec{\Omega}, E)$$

$$- \int dE' \int d\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \cdot \vec{\Omega})\psi(\vec{r}, E', \vec{\Omega}'). \tag{2.9}$$

Expressing the *fission source density* as,

$$F\psi = \int \int \nu\Sigma_f(\vec{r}, E')\psi(\vec{r}, \vec{\Omega}', E')d\Omega' dE', \tag{2.10}$$

the multiplication eigenvalue problem, equation (2.7), can be written in operator notation as

$$H\psi = \frac{1}{K}\chi F\psi. \tag{2.11}$$

Inverting H and operating on both sides by F, we obtain the following form of the eigenvalue problem:

$$F\psi = \frac{1}{K}[FH^{-1}\chi]F\psi. \tag{2.12}$$

Using the following definition for the transport operator

$$A \equiv FH^{-1}\chi, \tag{2.13}$$

the eigenvalue equation becomes,

$$KS = AS, \tag{2.14}$$

where S (= Fψ) is the *source eigenvector* and A is the *fission matrix*. The element A(l,m) of the fission matrix represents the probability that a neutron starting in cell

m will generate a fission neutron appearing in cell l. In integral form, the above equation can be expressed as,

$$KS(\vec{r}) = \int S(\vec{r'})A(\vec{r},\vec{r'})dr', \tag{2.15}$$

where $A(\vec{r},\vec{r'})$ is the operator that describes the probability that a neutron originating at $\vec{r'}$ produces a fission neutron at $\vec{r}$ and $S(\vec{r})$ is the source distribution. The above equation can also be derived from the integral transport form of the K eigenvalue equation (equation (2.8)) by using the operator F and the following property of the Green's function:

$$\int\int\int d\vec{r'}d\vec{\Omega}'dE'G(\vec{r'},\vec{\Omega}',E' \rightarrow \vec{r},\vec{\Omega},E) = H^{-1}. \tag{2.16}$$

The multiplication eigenvalue problem of equation (2.14) is then solved by the method of power iteration [Var62, Wac66, Nak77].

2.3 Monte Carlo K Calculation

Source iteration and fission matrix methods are two techniques used for Monte Carlo K calculations. There are three different, but correlated, estimators [Bri88] that can be used to estimate K using the two algorithms. They are *collision*, *absorption* and *track length* estimators. These estimators are used to estimate K for a particular batch. Averaging over all the active batches gives the final estimate of K for each of the three estimators. *Active batches* are the ones that are used to accumulate K after the fundamental mode source distribution has been reached.

2.3.1 Monte Carlo K Estimators

An *estimator* is a specific function of the random samples, of a random variable, that statistically represents a true unknown mean. If x is a random variable with an

associated distribution and an unknown mean, then the function X(x_1, x_2, x_3,...,x_n) is an estimator of the unknown mean. The set {x_1, x_2, x_3,...,x_n} consists of n independent random samples selected from the probability density distribution of x. A good estimator should be unbiased, consistent, and efficient [Bow72]. An estimator is *unbiased* if its expected value equals the true mean, μ, i.e.,

$$E(X) = \mu, \tag{2.17}$$

for all X. *Consistency* applies to a single sample where the sample size, n, becomes large. An estimator is *consistent* if it approaches, in a statistical sense, the true mean as n gets large. An efficient estimator is the one, among a group of unbiased estimators, that produces the minimum variance for a given sample size n. Mathematical expression for each of the three estimators of K are given here. We use the superscripts C, A and TL for the collision, absorption and track length estimators respectively.

Collision Estimator:

The collision estimate of K for an active batch is:

$$K^c = \frac{1}{N} \sum_j \sum_i W_i \frac{\bar{\nu}\Sigma_f}{\Sigma_t} \tag{2.18}$$

where,

i = all collisions for a particle in regions where fission is possible;

j = all source particles for a batch;

N = number of source particles for a batch; and

W_i = weight of particle entering collision.

Then the quantity $W_i \frac{\bar{\nu}\Sigma_f}{\Sigma_t}$ in equation (2.18) is the expected number of neutrons to be produced from a fission process in collision i. Hence K^c is the mean number of fission neutrons produced per generation (or batch).

Absorption Estimator:

The absorption estimator of K for an active batch:

$$K^A = \frac{1}{N} \sum_i W_i \frac{\bar{\nu}\Sigma_f}{\Sigma_a}, \tag{2.19}$$

where,

i = an analog capture event; and

Σ_a = macroscopic absorption cross section.

Track Length Estimator:

The track length estimator for K is given by,

$$K^{TL} = \frac{1}{N} \sum_j \sum_i dW_i \frac{\bar{\nu}\Sigma_f}{\Sigma_t} \tag{2.20}$$

where,

d = track length segment;

i = all track length segments,d, for a particle; and

j = all source particles for a batch.

In criticality calculations at the end of each batch (or fission generations) an estimate of K is produced by each of these three estimators. The final K estimator of each type is the average over many batch K estimates. Recent studies [Urb95a] have suggested that a combination of the three estimators to be the best K estimate available. In our research work we have used the collision estimator for all criticality and perturbation calculations. It should be emphasized that for Monte Carlo criticality calculations, the final result is not a point estimate of K, but rather a confidence interval.

2.3.2 Monte Carlo K algorithms

The source iteration and fission matrix algorithms for Monte Carlo criticality calculations are discussed below using the collision estimator. For simplicity of no-

tation, we assume a homogeneous system made up of a single isotope material that has cross sections $\bar{\nu}\Sigma_f$, Σ_t and Σ_s.

Source Iteration Algorithm:

The source iteration method is formulated by solving the source equation (equation (2.15)),

$$S^n(\vec{r}) = \int S^{(n-1)}(\vec{r}')A(\vec{r},\vec{r}')dr' \tag{2.21}$$

and calculating K as the ratio of source neutrons of two successive neutron generations [Uss58, Men68] as follows:

$$K = \frac{\int S^n(\vec{r})dr}{\int S^{(n-1)}(\vec{r})dr}. \tag{2.22}$$

$S^n(\vec{r})$ is the source distribution of the nth generation of neutrons and $A(\vec{r},\vec{r}')$ describes the probability that a neutron originating at $\vec{r}'$ produces a fission neutron at $\vec{r}$. The Monte Carlo source iteration algorithm is given below:

1. N source neutrons are started with an initial guessed spatial distribution. Each neutron has a starting weight W = 1.0.

2. At every collision point where fission is possible we perform the following operations.

   (a) Accumulate the tally for the collision estimator,

$$TALLY_j = TALLY_j + W_i\frac{\bar{\nu}\Sigma_f}{\Sigma_t}$$

   where,

   i = each collision by the neutron, and

   j = all neutrons, j = 1,2,....,N.

(b) We record the location and the number (n'_i) of source neutrons to be started from that location (where collision occurred) for the next generation:

$$n'_i = INTEGER(\frac{W_i \bar{\nu} \Sigma_f}{\Sigma_t} \frac{1}{K} + \xi),$$

where ξ is a random number between 0 and 1, and K is the last batch's eigenvalue or a guessed one for the first batch.

$$n'_j = \sum_i n'_i.$$

(c) For survival biasing, we set the new weight after the i^{th} collision, W_{i+1}, to,

$$W_{i+1} = W_i \frac{\Sigma_s}{\Sigma_t}.$$

(d) We also apply the Russian roulette technique [Lew93] to eliminate low weight neutrons.

3. After all the N neutron histories have been tracked, the eigenvalue for that generation or batch is calculated as;

$$K = \frac{\sum_{j=1}^{N} TALLY_j}{N}.$$

4. The next batch is started with N′ source neutrons, where

$$N' = \sum_j n'_j.$$

For this next batch each source particle has a starting weight,

$$W = \frac{N}{N'}.$$

This completes the source iteration algorithm. It should be noted that the number of neutrons, started in each batch, times the weight of each neutron, for that batch,

is constant for the all batches, i.e., W*N′ is same for all the batches. N′ is close to N due to the K normalization done in step 2(b). If the initial guess of K is too low or too high, the number of fission neutrons produced for the next batch will be too high or too low relative to the desired nominal number N. The number of fission neutrons, n_i' stored at each collision is rounded up or down (step 2(b)) to an integer (including zero) with a probability proportional to its deviation from that integer.

Fission Matrix Algorithm:

The fission matrix method [Kap58, Mih67] solves the eigenvalue equation

$$KS(\vec{r}) = \int S(\vec{r'})A(\vec{r},\vec{r'})dr', \tag{2.23}$$

which holds for every generation. Equation (2.23) can be discretized and is then reduced to the matrix equation,

$$KS = AS, \tag{2.24}$$

where S is the source eigenvector and A is the fission matrix. The element A(l,m) of the fission matrix describes the probability that a neutron starting in cell m will generate a fission neutron in cell n [Ave58]. Assuming that the physical space is discretized into M cells, the Monte Carlo fission matrix algorithm can be described as:

1. N source neutrons are started with an initial guessed spatial distribution. Each neutron has a starting weight W = 1.0. An array C(m) keeps track of the number of neutrons that start from cell m where m = 1,2,...,M.

2. At every collision point where fission is possible, the following operations are performed:

(a) If the collision occurs in cell l, accumulate into matrix B(l,m) the contribution to K due to a neutron that started from cell m, i.e.,

$$B(l,m) = B(l,m) + W_i \frac{\bar{\nu}\Sigma_f}{\Sigma_t}$$

where

i = every collision that a neutron encounters and $\bar{\nu}\Sigma_f$ and Σ_t are macroscopic cross sections for cell l.

(b) Record the location and the number (n'_i) of source neutrons to be started from that location (where collision occurred) for the next generation.

$$n'_i = INTEGER(\frac{W_i \bar{\nu}\Sigma_f}{\Sigma_t}\frac{1}{K} + \xi),$$

where ξ is a random number between 0 and 1, and K is the previous batch eigenvalue (or a guessed one for the first batch).

$$n'_j = \sum_i n'_i.$$

(c) For survival biasing, adjust the new weight after the i^{th} collision,

$$W_{i+1} = W_i \frac{\Sigma_s}{\Sigma_t}.$$

(d) Apply the Russian roulette technique [Lew93] to eliminate low weight neutrons.

3. After all the N neutron histories have been tracked, compute the fission matrix A(l,m) element,

$$A(l,m) = \frac{B(l,m)}{C(m)}.$$

The element A(l,m) gives the expected number of fission neutrons produced in cell l due to a unit source neutron in cell m. The dominant eigenvalue, or K, of the fission matrix A can be calculated by using a matrix iterative algorithm.

4. Start the next batch with N′ source neutrons, where

$$N' = \sum_j n'_j,$$

where each source particle has a starting weight,

$$W = \frac{N}{N'}.$$

This completes the fission matrix algorithm. As with source iteration, the quantity W*N′ is preserved for all batches. There are two ways to formulate the fission matrix algorithm. These two formulations are explained in detail in section 2.5. The numerical examples of section 2.4 are solved using the cycle fission matrix approach, where the elements of the matrix B(l,m) and the array C(m) are zeroed out at the begining of every batch.

The following information is provided to a Monte Carlo eigenvalue simulation algorithm:

- the nominal number of source histories per batch;
- the initial guess of K;
- the number of initial batches, I_i, to be discarded before tally accumulation;
- the total number of batches, I_t, for the problem; and
- an initial spatial distribution for the source.

The number of active batches, I_a that are used to evaluate the mean and the standard deviation for K is then given by,

$$I_a = I_t - I_i. \tag{2.25}$$

2.3.3 Statistics for Monte Carlo K Calculation

A Monte Carlo eigenvalue simulation provides the following estimate of the average eigenvalue,

$$K = \frac{1}{I_a} \sum_{i=1}^{I_a} K_i, \tag{2.26}$$

where K_i is the batch eigenvalue. The *sample variance* in this estimate of K is

$$\sigma_s^2 = \frac{\sum_{i=1}^{I_a} {K_i}^2}{I_a - 1} - \frac{(\sum_{i=1}^{I_a} K_i)^2}{I_a(I_a - 1)}. \tag{2.27}$$

The *standard deviation* is then

$$\sigma = \frac{\sigma_s}{I_a^{\frac{1}{2}}}, \tag{2.28}$$

which is provided along with the average eigenvalue estimate. The expressions for the variance and standard deviations assume that the batch eigenvalues are independent, hence no batch-to-batch correlations exist after the fundamental eigenmode is reached.

2.3.4 Issues Related to Monte Carlo K Calculation

The first issue that needs to be addressed regarding Monte Carlo K calculations is that of source convergence. Before accumulating any K tally data, enough batches must be performed and discarded to allow the source neutron distribution to attain the fundamental mode [Whi71]. Good spatial sampling is important for attaining and maintaining the fundamental eigenmode. Maintaining the fundamental eigenmode may be difficult, especially for systems with high dominance ratio, due to the batch-to-batch correlations in the spatial distributions of fission neutrons. This is due to the fact that for systems with high dominance ratio, there is less neutron communication between different regions of the system and spatial correlations between batches may prevail.

The second issue deals with the assumption that the batch eigenvalues are independent. For systems with a high dominance ratio, batch-to-batch correlations among fission neutron distribution exist, and this assumption is invalid. This results in an underestimation of the standard deviation [Moo76, Gel90a]. Various studies [Mac73, Gas75] have been done to account for this phenomenon.

The third issue involves the choice of the optimum number of batches versus the optimum number of neutrons per batch [Lew93], as illustrated in figures 2.2 and 2.3. Figure 2.2 pertains to the case where a large number of neutrons per batch has been followed for a few batches, while figure 2.3 shows the case where a small number

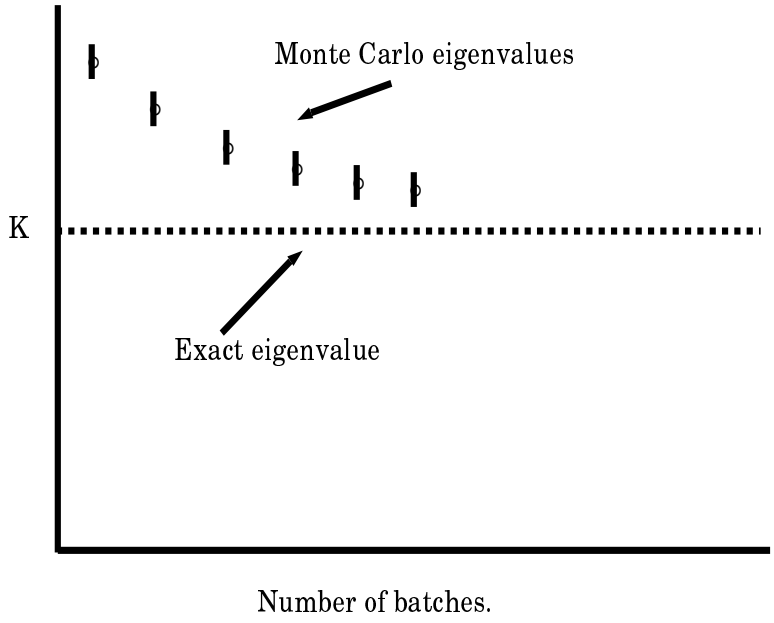


Figure 2.2: Few Batches with Large Number of Histories per Batch.

of neutrons per batch was followed for a large number of generations. Note that in figure 2.2, the fundamental eigenmode is not reached, even though the variance in the eigenvalue estimate is small. On the other hand, figure 2.3 shows that the fundamental

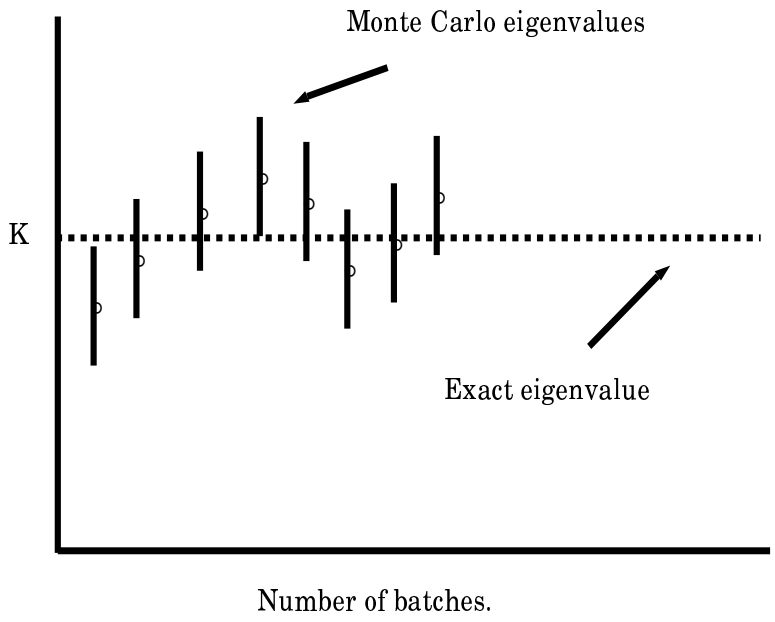


Figure 2.3: Large Number of Batches with Few Histories per Batch.

eigenmode is reached, albeit with a large variance in the eigenvalue estimator.

Lastly, we address the issue of *bias* in Monte Carlo eigenvalue calculation [Gel91]. The bias is a result of the fission source normalization done after each batch. Numerical experiments [Bow83] suggest that biases in fluxes and eigenvalues are negligibly small for most practical cases. The conclusion that, in practice, the eigenvalue bias is negligible and smaller than a single standard deviation, also has strong theoretical support [Gel90b]. A detailed analysis of bias [Bri86, Gel94] gives the following equation;

$$E\{K_0\} = \lambda_0 - (\frac{N_G}{2\lambda_0})(\sigma_t{}^2 - \sigma_a{}^2), \tag{2.29}$$

where,

K_0 is the biased Monte Carlo eigenvalue computed by averaging over generations or batches;

λ_0 is the true eigenvalue;

$\sigma_t{}^2$ is the true variance in K_0;

${\sigma_a}^2$ is the apparent variance in K_0, computed by assuming that estimates of K from different generations are independent; and

N_G is the active number of generations.

Thus the bias, $\Delta\lambda$, is given by,

$$\Delta\lambda = -\frac{N_G}{2\lambda_0}({\sigma_t}^2 - {\sigma_a}^2), \tag{2.30}$$

and the relative bias (compared to the standard deviation) is bounded by

$$\frac{|\Delta\lambda|}{\sigma_t} < \frac{N_G}{2}\frac{\sigma_t}{\lambda_0}. \tag{2.31}$$

For typical eigenvalue calculations one prefers $\frac{\sigma_t}{\lambda_0} < 0.0025$, and hence if N_G is less than 800 the bias will be less that a standard deviation [Gel90a]. It should be noted that the bias itself is independent of N_G since ${\sigma_t}^2$ is proportional to $\frac{1}{N_G}$. Therefore, an adequate number of histories per batch should be tracked; and a number of batches should be simulated which is statistically large enough, but not so large such that the bias in the estimate of K is a significant fraction of the standard deviation.

2.4 Numerical Results

In the following subsections we provide various numerical results for Monte Carlo eigenvalue calculations and compare them to benchmark and discrete ordinates (S_N) results. The comparison of results verifies the accuracy of our Monte Carlo algorithms and codes. The benchmark [Kap74] results are for one energy group, homogeneous slabs and are based on analytical transport theory. The discrete ordinate results were obtained using the TWODANT [O'De82] code. Monte Carlo eigenvalue results are given for both the source iteration and the fission matrix algorithms. All the test problems show here are for systems with isotropic scattering.

2.4.1 Numerical Results I

Numerical results of this section show the batch eigenvalues, the batch averaged eigenvalues, and the standard deviation for the batch averaged eigenvalues. This first numerical test problem [Kap74] is for a single energy group, critical, homogeneous slab, with isotropic scattering, and vacuum boundary conditions on both ends. The half thickness of the slab is 8.3295135616 cm and the cross sections are Σ_t= 1.0 cm^{-1}, $\Sigma_s = 0.9$ cm^{-1} and $\bar{\nu}\Sigma_f = 0.11$ cm^{-1}, as shown in figure 2.4. The Monte Carlo cases shown here are for 120 batches, with the initial 40 batches discarded, each batch consisting of 2000 neutrons. Figure 2.5 shows the plot of batch eigenvalue, batch averaged eigenvalue, and the standard deviation of the batch averaged eigenvalue using source iteration. The value of the batch averaged K after 80 active batches is .99988 $\pm$0.56E-3. Figure 2.6 shows the same results utilizing the fission matrix approach, where the value of K is .99983 $\pm$0.55E-3. It should be noticed that for both cases the estimated eigenvalue is within one standard deviation of the benchmark (K = 1.0) result. From now on, only the batch averaged eigenvalue and its associated standard deviation will be provided.

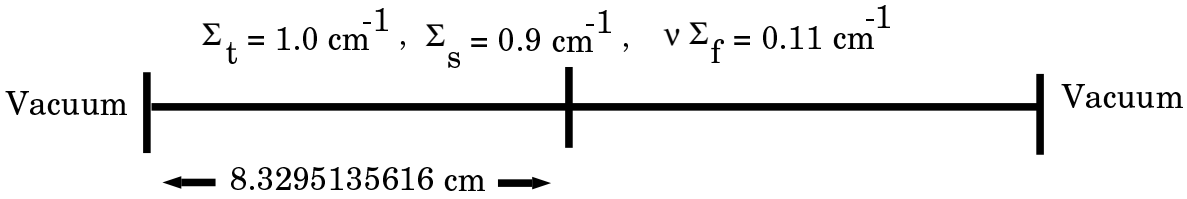


Figure 2.4: Configuration of Homogeneous Slab.

Table 2.1 compares the source iteration and the fission matrix results for different

single energy critical homogeneous slab problems [Kap74] with vacuum boundary conditions on both ends. These problems are similar to that shown in figure 2.4 except different cross sections and half thicknesses. All the Monte Carlo results were produced with 40 inactive batches, 80 active batches and 2000 neutrons per batch. We notice that as the half thickness decreases, in terms of mfps, the standard deviation of the eigenvalue increases, even though all the problems are simulated with the same number of batches and neutrons per batch. This is due to the fact that neutrons leak out more easily from smaller systems and hence fewer neutrons contribute to the eigenvalue estimators.

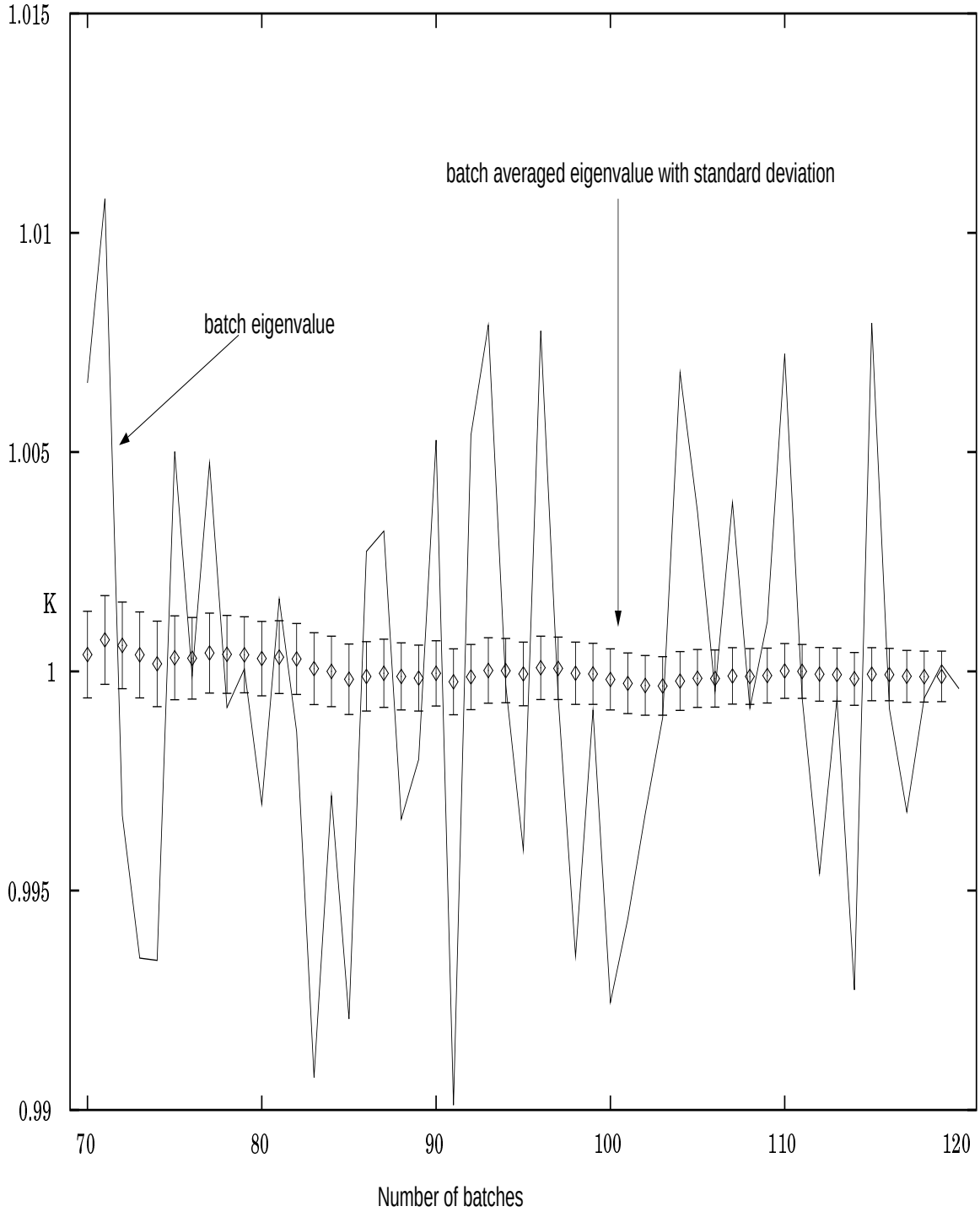


Figure 2.5: Eigenvalue Calculation Using Source Iteration Algorithm.

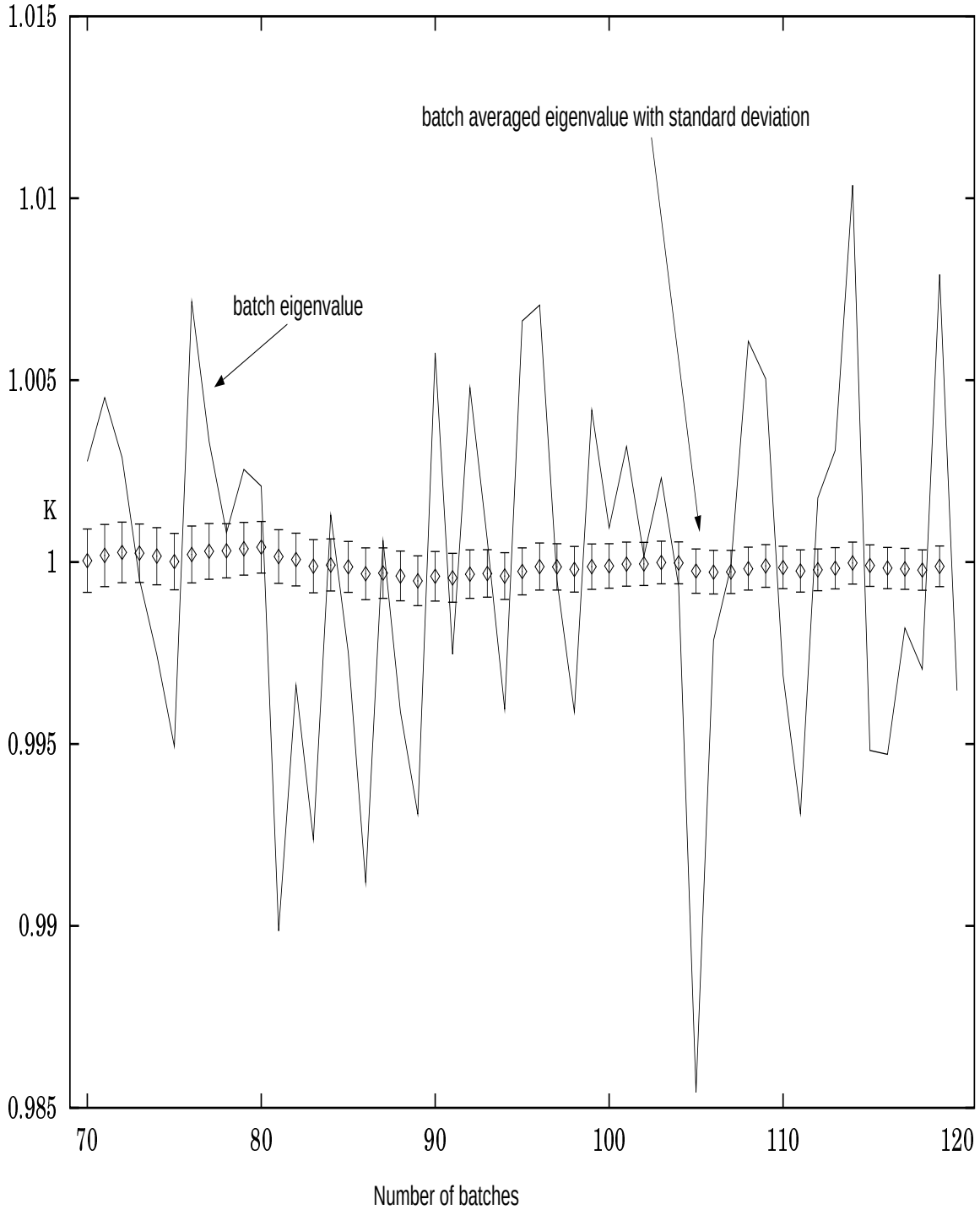


Figure 2.6: Eigenvalue Calculation Using Fission Matrix Algorithm.

Half thickness (mfp)	$\frac{\Sigma_s+\bar{\nu}\Sigma_f}{\Sigma_t}$	Source iteration	Fission matrix
5.6655054562	1.02	0.99930 ±.85E-3	0.99970 ±.91E-3
3.300263772	1.05	1.0005 ±.12E-2	1.0005 ±.14E-2
2.113309666	1.10	0.9999 ±.15E-2	1.0008 ±.15E-2
1.289379285	1.20	1.0004 ±.19E-2	1.0007 ±.19E-2
0.93772556	1.30	0.9976 ±.23E-2	1.0000 ±.22E-2
0.73660355	1.40	1.0049 ±.21E-2	1.0052 ±.23E-2
0.51196298	1.60	1.0052 ±.28E-2	1.0023 ±.29E-2
0.31102598	2.00	0.9987 ±.37E-2	1.0036 ±.36E-2

Table 2.1: Results for Homogeneous Critical Slabs.

Next in table 2.2 we show the results for single energy group heterogeneous slab test problems with configurations as shown in figure 2.7 and 2.8. The Monte Carlo eigenvalue results ($I_i = 40$, $I_a = 80$, 2500 neutrons per batch) are compared to that of the TWODANT code with S_{32} quadrature sets and inner and outer iteration convergence criteria of 10^{-12}.

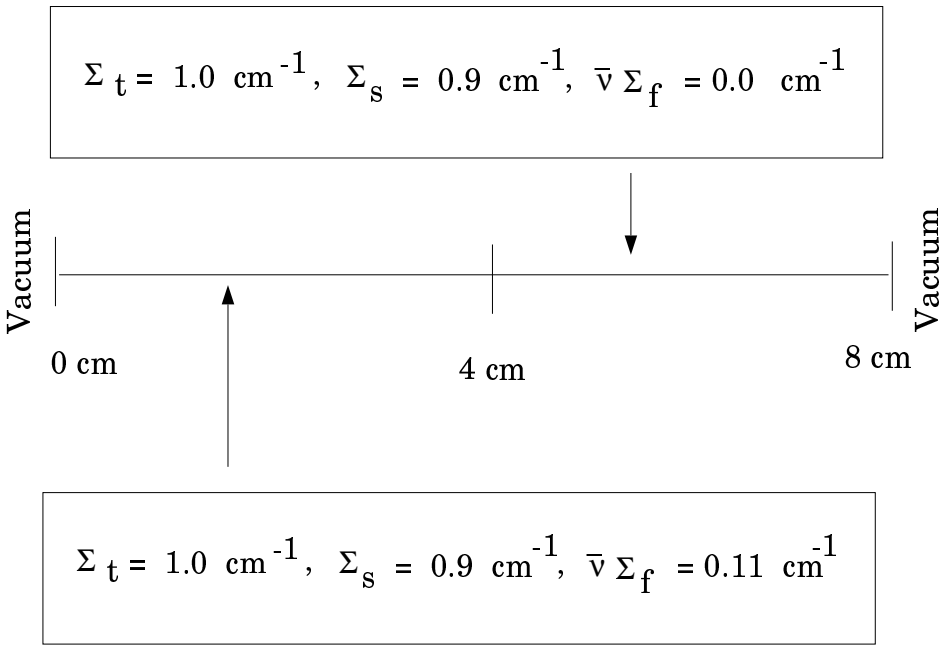
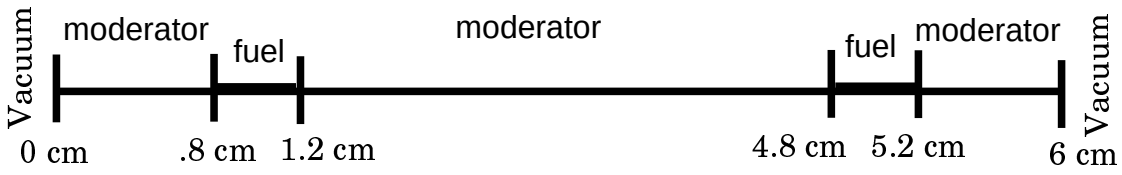


Figure 2.7: Configuration of Heterogeneous Slab 1.

Problem	TWODANT	Source iteration	Fission matrix
Hetero-slab 1	0.847805	0.84829 ± .63E-3	0.84805 ± .68E-3
Hetero-slab 2	1.00010	1.0012 ± .17E-2	1.0061 ± .17E-2
Hetero-slab 3	0.925597	0.9194 ± .16E-2	0.9307 ± .20E-2

Table 2.2: Results for Heterogeneous Slab Problems.

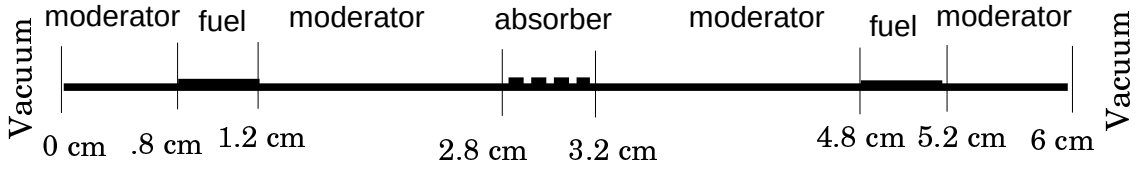


Heterogeneous slab 2

$\Sigma_t = .717\ cm^{-1}$, $\Sigma_s = .397\ cm^{-1}$, $\bar{\nu}\Sigma_f = .756\ cm^{-1}$
fuel

$\Sigma_t = 5.0\ cm^{-1}$, $\Sigma_s = 4.9\ cm^{-1}$, $\bar{\nu}\Sigma_f = 0.0\ cm^{-1}$
absorber

$\Sigma_t = 2.35\ cm^{-1}$, $\Sigma_s = 2.3\ cm^{-1}$, $\bar{\nu}\Sigma_f = 0.0\ cm^{-1}$
moderator



Heterogeneous slab 3

Figure 2.8: Configuration of Heterogeneous Slabs 2 and 3.

2.4.2 Numerical Results II

In this section we provide results for two dimensional XY geometry cases in one and two energy groups. Test problems 1 and 2 are for one energy group cases.

Configurations for problems 1 and 2 are specified in figure 2.9. The first test problem is a 8 cm by 8 cm square region with homogeneous cross sections of Σ_t=1.0 cm^{-1}, $\bar{\nu}\Sigma_f = 0.11$ cm^{-1}, and $\Sigma_s = 0.9$ cm^{-1}; and with vacuum boundary conditions on all sides. The second test problem is an 8 cm by 8 cm square region with an interior 4 cm by 4 cm square region and vacuum boundary conditions on all sides. The inner square region has cross sections of Σ_t=1.0 cm^{-1}, $\bar{\nu}\Sigma_f = 0.21$ cm^{-1}, and $\Sigma_s = 0.9$ cm^{-1}, and the rest of the system has cross sections of Σ_t=1.0 cm^{-1}, $\bar{\nu}\Sigma_f = 0.11$ cm^{-1}, and $\Sigma_s = 0.9$ cm^{-1}. Test problems 1 and 2 were simulated for 120 active batches, 40 inactive batches and with 4000 neutrons per batch. Test problem 3 is a homogeneous square region with vacuum boundary conditions on all sides and two energy groups. Cross sections for test problem 3 and the configuration is shown in figure 2.10. Test problem 3 was simulated with 120 active batches, 40 inactive batches and 2500 neutrons per batch. Results for the two dimensional problems are shown in table 2.3. The reference TWODANT results are for S_{16} quadrature sets and with inner and outer iteration convergence criteria of 10^{-12}.

Problem	TWODANT	Source iteration	Fission matrix
Test Problem 1	0.640792	0.64089 ± .73E-3	0.64130 ± .83E-3
Test Problem 2	1.004494	1.00502 ±.89E-3	1.00609 ±.82E-3
Test Problem 3	1.489619	1.4897 ±.17E-2	1.4880 ±.20E-2

Table 2.3: Results for Two-Dimensional Problems.

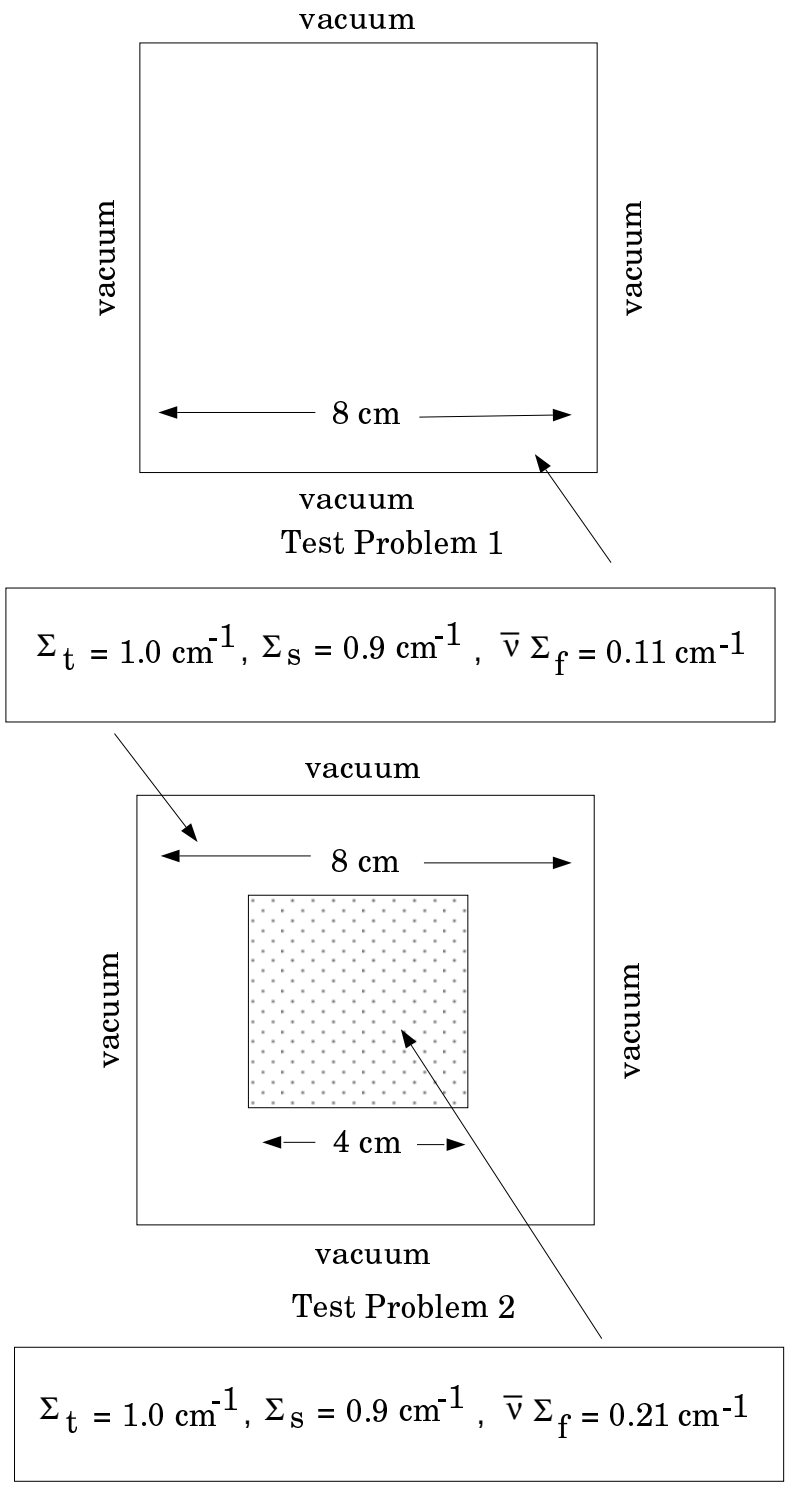


Figure 2.9: Configuration of Two-Dimensional One Energy Group Problems.

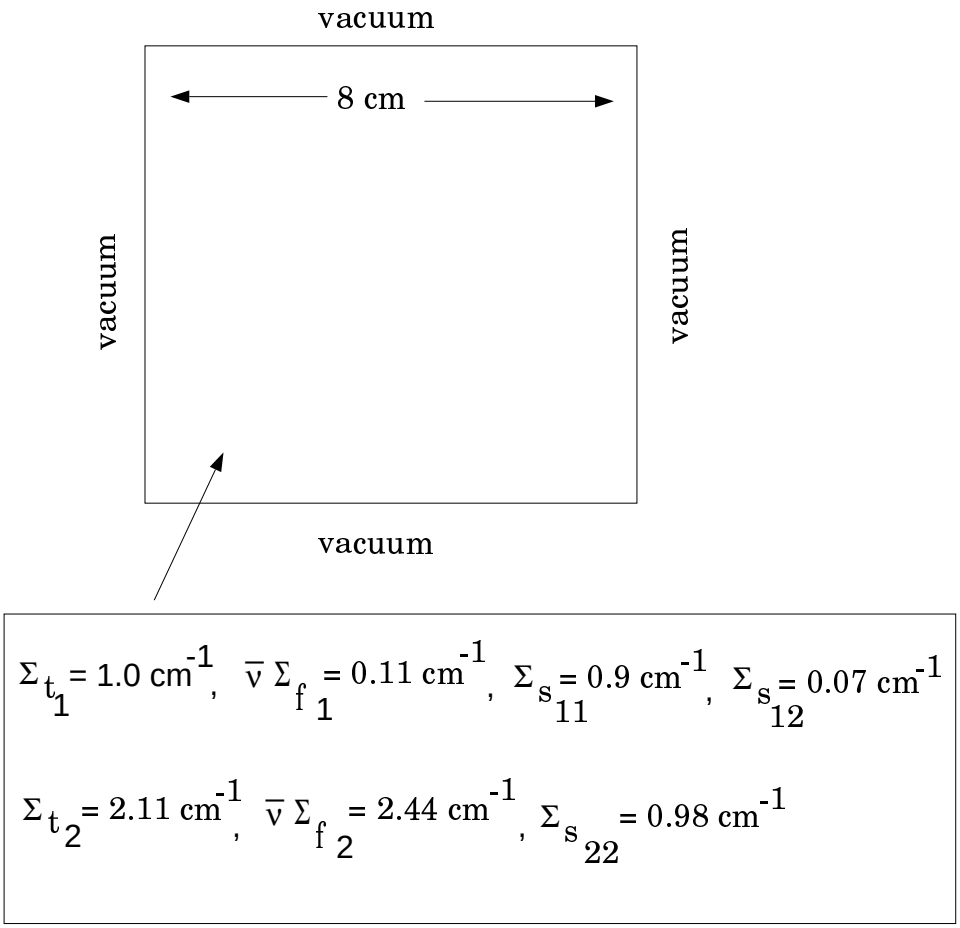


Figure 2.10: Configuration of Two-Dimensional Two Energy Group Problem.

2.5 Cycle Versus Cumulative Fission Matrix Algorithm

We have mentioned in section 2.3.2. that there are two variations of the fission matrix algorithm. In this section we introduce these two variations and perform numerical experiments to study the effect of these two algorithms on different test problems. In section 2.3.2 the fission matrix algorithm was described. We have shown that for every collision that occurs in a cell l, we accumulate in a matrix B(l,m) the contribution to K due to a neutron that started from cell m. An array C(m) stores the number of neutrons that started from cell m. After all histories are

tracked, the fission matrix A(l,m) is computed as

$$A(l,m) = \frac{B(l,m)}{C(m)}. \tag{2.32}$$

The dominant eigenvalue, or K, of the fission matrix A is evaluated numerically.

The first variation of the fission matrix algorithm, called the cycle fission matrix (cyfm) algorithm, zeroes out all elements of the matrix B(l,m) and the array C(m) at the begining of each batch. Hence, the fission matrix is formed only from the contribution of all neutrons of a particular batch. The batch K is calculated at the end of each batch, along with the batch averaged K over all batches, and the standard deviation of the batch averaged K. Fission source normalization is done using the batch K. Since the fission matrix is computed for every batch, independent of any contribution from previous batches, the batch Ks are independent and we can calculate the batch averaged K and the associated standard deviation.

The second variation, called the cumulative fission matrix (cumfm) algorithm, does not zero out elements of the matrix B(l,m) and the array C(m) at the begining of each batch. Contributions to both B(l,m) and C(m) are accumulated over batches and the fission matrix A(l,m) is computed from them. At the end of each batch we calculate the batch K. Since contributions to the fission matrix are accumulated over batches, the batch eigenvalues are not independent in a statistical sense, and hence the batch averaged eigenvalue and the associated standard deviation are not computed. The batch K is used for fission source normalization.

We now perform computational experiments to see the effect of these two fission matrix algorithms on different test problems. We will specifically look at the effect of these algorithms on loosely coupled versus tightly coupled problems. Effects of different batch sizes, inactive and active batches will also be studied. The test

cases [Urb95b] are slab geometry problems, with isotropic scattering and vacuum boundaries.

The first test problem is a tightly coupled problem consisting of a 60 mfp homogeneous slab with $\Sigma_t = 1.0$, $\Sigma_s = 0.7$, and $\nu\Sigma_f = 0.3071$. A fine mesh S_{32} calculation gives a K of 1.02070 for this test problem. Results for this test problem are shown in table 2.4 for different batch sizes and for different numbers of inactive and active batches. We observe from table 2.4 that both fission matrix algorithms perform equally well. We cannot differentiate one being superior to the other algorithm for this test problem. For this test problem the effect of different batch sizes and different numbers of inactive and active batches is not significant.

The second test problem, which is a loosely coupled problem, is a lattice of fuel regions and absorber regions. The fuel regions are each 2 cm thick and are separated by 1 cm thick regions of absorbing material. The material at the boundaries is another 1 cm slab of absorbing material. There are 19 fuel regions and the thickness of the entire slab is 58 cm. Cross sections of the fuel regions are Σ_t=1.0, Σ_s=0.7, and $\nu\Sigma_f$=0.3071; and cross sections of the absorbing regions are Σ_t=1.0, Σ_s=0.001, and $\nu\Sigma_f$=0.0. A fine mesh TWODANT calculation gives a K of 0.59852 for this problem. Results for this test problem are shown in table 2.5 for different batch sizes and for different numbers of inactive and active batches. We observe from table 2.5 that the cumulative fission matrix algorithm performs better than the cycle fission matrix algorithm. The test problem is a loosely coupled problem due to the strong absorbing materials separating fuel regions. Sampling only with neutrons of each batch does not provide good enough sampling for fission reaction, and hence the cycle fission matrices do not represent the loosely coupled system well. Even for the last case of table 2.5, i.e., with 5000 neutrons per batch, the eigenvalue has not

reached the

Fine mesh	TWODANT	K =	1.02070
#of neutrons/batch	I_i, I_a	cyfm	cumfm
500	I_i=0,I_a=80	1.02053±.27E-3	1.01923
500	I_i=20,I_a=60	1.02096 ±.29E-3	1.01923
1000	I_i=0,I_a=80	1.02072±.21E-3	1.01995
1000	I_i=20,I_a=60	1.02123±.19E-3	1.01995
1500	I_i=0,I_a=80	1.02026±.19E-3	1.01991
1500	I_i=20,I_a=60	1.02079±.16E-3	1.01991
2000	I_i=0,I_a=80	1.01989±.15E-3	1.01946
2000	I_i=20,I_a=60	1.02019 ±.13E-3	1.01946
3000	I_i=10,I_a=150	1.020080±.91E-4	1.02011
3000	I_i=50,I_a=150	1.02024 ±.10E-3	1.02011
4000	I_i=10,I_a=150,	1.020324±.73E-4	1.01993
4000	I_i=50,I_a=150,	1.020564±.70E-4	1.01993
5000	I_i=10,I_a=150,	1.020732±.69E-4	1.02023
5000	I_i=50,I_a=150,	1.021019±.59E-4	1.02023

Table 2.4: Cycle (batched averaged) and Cumulative Fission Matrix Results for the First Test Problem.

Fine mesh	TWODANT	K =	0.59852
# of neutrons/batch	I_i, I_a	cyfm	cumfm
500	I_i=0,I_a=80	0.791± .11E-1	0.6005
500	I_i=20,I_a=60	0.792± .12E-1	0.6005
1000	I_i=0,I_a=80	0.6971±.77E-2	0.59925
1000	I_i=20,I_a=60	0.7024±.94E-2	0.59925
1500	I_i=0,I_a=80	0.6504±.29E-2	0.59872
1500	I_i=20,I_a=60	0.6512±.37E-2	0.59872
2000	I_i=0,I_a=80	0.6466±.58E-2	0.59953
2000	I_i=20,I_a=60	0.6512±.75E-2	0.59953
3000	I_i=10,I_a=150	0.6268±.16E-2	0.60026
3000	I_i=50,I_a=150	0.6289±.22E-2	0.60026
4000	I_i=10,I_a=150,	0.61862±.99E-3	0.59909
4000	I_i=50,I_a=150,	0.6189±.12E-2	0.59909
5000	I_i=10,I_a=150,	0.6176±.12E-2	0.59883
5000	I_i=50,I_a=150,	0.6188±.15E-2	0.59883

Table 2.5: Cycle (batched averaged) and Cumulative Fission Matrix Results for Second Test Problem.

correct fine mesh result for the cycle fission matrix algorithm. We observe that as the number of neutrons per batch is increased, the cycle fission matrix eigenvalue approaches the correct result. This indicates that a large number of neutrons per batch is necessary for the cycle fission matrix algorithm to perform well for this test problem. In figure 2.11 we plot the batch eigenvalue and the batch averaged eigenvalue of the cycle fission matrix algorithm and the batch eigenvalue of the

cumulative fission matrix algorithm for the first test problem with 5000 neutrons per batch and I_i=50 and I_a=100. Only the last 40 batches are shown in figure 2.11. From this figure we notice that the batch eigenvalue of the cumulative fission matrix is far superior to the batch eigenvalue of the cycle fission matrix, as expected, whereas the batch averaged eigenvalue of the cycle fission matrix and the batch eigenvalue of the cumulative fission matrix show comparable accuracy. Figure 2.12 shows similar plot for second test problem. From figure 2.12 it is clear that the cumulative fission matrix algorithm yields superior results to the cycle fission matirx algorithm for loosely coupled problems.

Numerical experiments with other test problems have also shown similar behaviour. We can make a few conclusions from these numerical experiments. As expected, the batch eigenvalue of the cumulative fission matrix algorithm always provides superior result compared to the batch eigenvalue of the cycle fission matrix algorithm. The batch eigenvalue of the cumulative fission matrix algorithm and the batch averaged eigenvalue of the cycle fission matrix algorithm provide comparable results for tightly coupled problems, where there is enough neutron communication between different regions. For a loosely coupled system, the cumulative fission matrix eigenvalue result is superior to the batch averaged eigenvalue of the cycle fission matrix algorithm. For this kind of system, the cycle fission matrix algorithm should be used with caution.

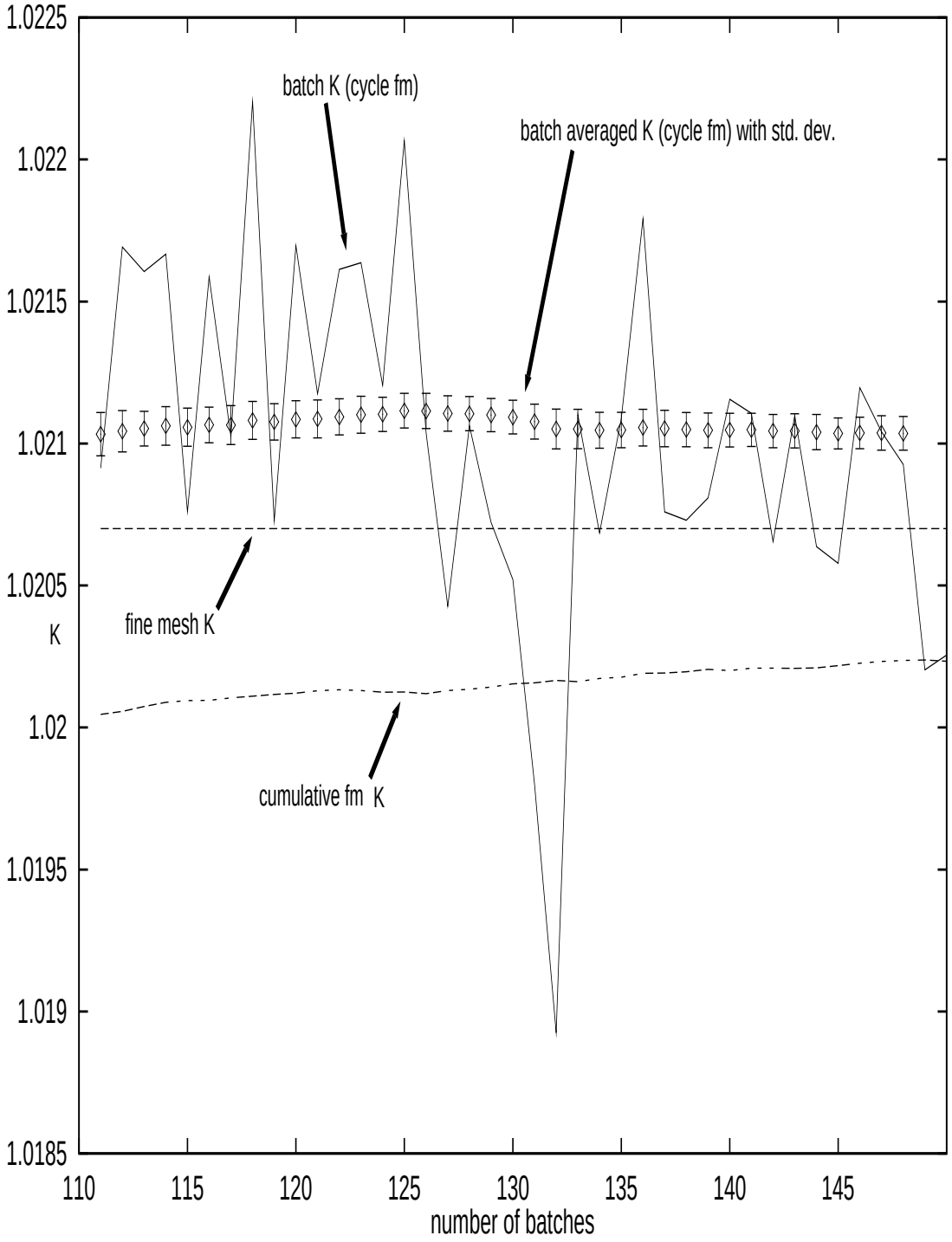


Figure 2.11: Fission Matrix (Cycle and Cumulative) K for First Test Problem.

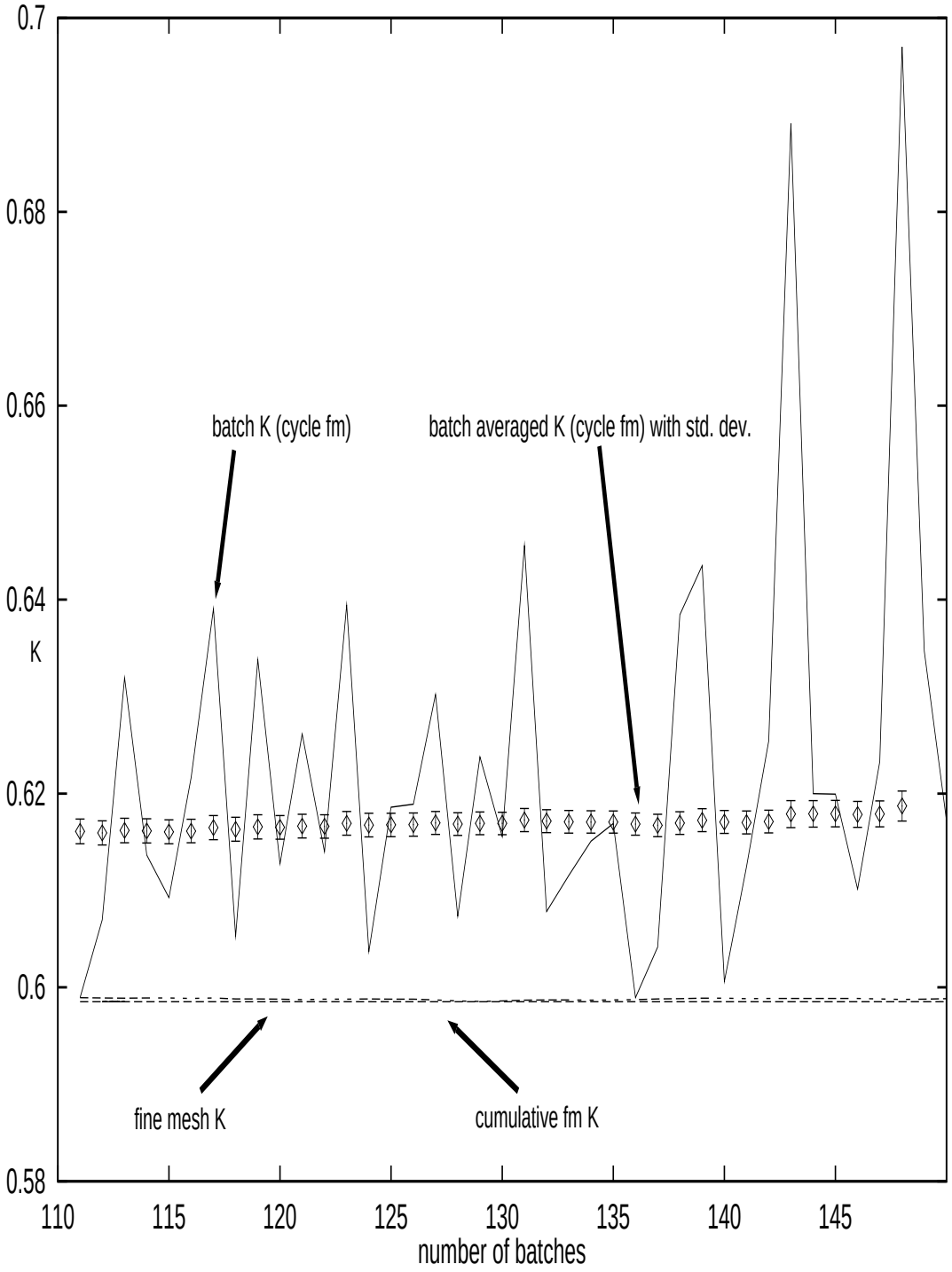


Figure 2.12: Fission Matrix (Cycle and Cumulative) K for Second Test Problem.

CHAPTER III

CORRELATED SAMPLING MONTE CARLO

3.1 Introduction

In this chapter, the correlated sampling technique, for Monte Carlo eigenvalue perturbation calculations, is described. Several issues that lead to the combination of the fission matrix and correlated sampling methods as a Monte Carlo eigenvalue perturbation technique are discussed. The idea of performing a Monte Carlo simulation in an artificial reference system, different from both the unperturbed and perturbed systems, is introduced. Numerical results are provided in support of all the theories developed in this chapter.

3.2 Difficulties of Monte Carlo Eigenvalue Perturbation

In a straightforward Monte Carlo approach, a perturbation effect would be evaluated by performing two independent simulations for the perturbed and unperturbed systems. The perturbation result is then calculated by taking the difference between these two independent simulations. In this approach, the variance of the result increases as the perturbation becomes smaller, and hence the differential effect becomes difficult to observe due to a very large uncertainty [Rie86]. This can be explained by deriving the expression of the variance of a differential effect.

Let x be a random variable with an associated distribution and an unknown mean. We denote the function X(x_1, x_2, x_3,...) as an estimator of the unknown mean. The expected value of X is given by,

$$E(X), \tag{3.1}$$

and the variance by,

$$V(X) = E(X - E(X))^2. \tag{3.2}$$

X may depend on several variables and parameters. Let us assume a small variation in a parameter (or parameters) resulting in a different unknown mean, which is estimated by the function X^*, with expectation value,

$$E(X^*), \tag{3.3}$$

and variance,

$$V(X^*) = E(X^* - E(X^*))^2. \tag{3.4}$$

Therefore the expectation of the differential effect due to the small variation in the parameter(s) is given by,

$$E(X - X^*) = E(X) - E(X^*), \tag{3.5}$$

and the variance of the differential effect by,

$$V(X - X^*) = V(X) + V(X^*) - 2cov(X, X^*). \tag{3.6}$$

If the two estimates E(X) and E(X^*) are obtained by two independent Monte Carlo simulations (i.e., in the absence of any correlations between the two simulations), the covariance term in equation (3.6) vanishes and the variance is

$$V(X - X^*) = V(X) + V(X^*). \tag{3.7}$$

This is larger than either of the variances separately and may be larger than the expression given in equation (3.6) [Spa69]. Hence, it is possible to reduce the variance of the differential effect, specially for small perturbations, if one has a strong positive correlation between the unperturbed and perturbed Monte Carlo simulations.

3.2.1 Numerical Examples

Numerical examples are provided here to demonstrate that straightforward subtraction of two independent Monte Carlo simulations is problematical for calculating small perturbation effects. All the test problems correspond to a homogeneous, one energy group, slab, with thickness of 16 cm and vacuum boundary conditions on both ends. Table 3.1 shows the unperturbed and perturbed cross sections and ΔK results calculated by two independent Monte Carlo simulations and the reference TWODANT simulation. Monte Carlo results are for 30 inactive batches, 70 active batches and 2000 neutrons per batch. The TWODANT results were generated using S_{32} quadrature sets and inner and outer iteration convergence criteria of 10^{-12}.

Unperturbed cross sections: Σ_t=1.0 cm^{-1}, $\nu\Sigma_f$=0.11 cm^{-1}, Σ_s=0.9 cm^{-1}			
Problem #	Perturbed cross sections	TWODANT ΔK	Monte Carlo ΔK
1	$\Delta\Sigma_t$=.001, $\Delta\Sigma_a$=.001	-.008807	-.00843 $\mp$.89E-3
2	$\Delta\Sigma_t$=.0001, $\Delta\Sigma_a$=.0001	-.000888	.00033$\mp$.89E-3
3	$\Delta\Sigma_f$=.0001	.000903	.00124 $\pm$.86E-3

Table 3.1: Perturbation Results from Independent Monte Carlo Simulations.

Results of table 3.1 show, with the exception of the first test problem, that subtracting two independent Monte Carlo runs to calculate small perturbation effects

can yield significant errors. Similar observations can be found in other references [Wal94, Gal95] also. Later in this chapter (in section 3.3.4), test problems 1, 2 and 3 are solved using the combined correlated sampling fission matrix (CSFM) approach (with the same number of batches and neutrons per batch as in the cases of table 3.1) and the results are significantly more accurate than in table 3.1. Also, later in this chapter, it is shown that when the correlated sampling technique is applied with the source iteration method, it encounters severe difficulties; and it is found that the combined CSFM method yields far superior results in Monte Carlo calculations of perturbation effects.

3.3 Correlated Sampling Technique

Correlated sampling techniques force all the histories corresponding to the perturbed system to follow the same transition points in phase space as the unperturbed histories. Appropriate weight factors are then used to adjust the particle weights at the transition points. This can be explained mathematically by looking at the integral form of the neutron transport equation (equation 2.1), expressed in terms of collision density and its solution by the Neumann series [Spa69, Lux91]. The collision density equation is given by,

$$\varphi(x) = \int_V \varphi(y)\kappa(y,x)dy + Q(x), \tag{3.8}$$

where x and y are the coordinates of a particle in the six-dimensional phase space, $\kappa(x,y)$ is the transport kernel from y to x, $\varphi(x)$ is the collision density of particles entering a collision in x, and Q(x) is the external particle source in x. The Neumann series solution of equation (3.8) is given by,

$$\varphi(x) = \sum_{n=0}^{\infty} \varphi_n(x)$$

$$= Q(x) + \sum_{n=1}^{\infty} \int_V \cdots \cdots \int_V \kappa(u_n, x)\kappa(u_{n-1}, u_n).....\kappa(u_1, u_2)du_n.....du_1 Q(u_1), \quad (3.9)$$

where $\kappa(u_{i-1}, u_i)$ is the probability that a particle entering into a collision at $\vec{r}_{i-1}$ with energy $\vec{E}_{i-1}$ will appear at $\vec{r}_i$ with energy $\vec{E}_i$,

$$\kappa(u_{i-1}, u_i) = \kappa(\vec{r}_{i-1}, \vec{E}_{i-1} \rightarrow \vec{r}_i, \vec{E}_i). \quad (3.10)$$

Here

$$\varphi_0(x) = Q(x), \quad (3.11)$$

is the direct source contribution term. For n = 1 we get the once-collided term,

$$\varphi_1(x) = \int_V \kappa(u_1, x)Q(u_1)du_1. \quad (3.12)$$

Similarly for n = 2,3... etc. the twice, thrice ... etc. collided terms can be found. The transport kernel $\kappa(u_{i-1}, u_i)$ is expressed in terms of the product of a collision kernel, C($\vec{E}_{i-1} \rightarrow \vec{E}_i|\vec{r}_{i-1}$), and a translation kernel, T($\vec{r}_{i-1} \rightarrow \vec{r}_i|E_i$), as shown below,

$$\kappa(u_{i-1}, u_i) = \kappa(\vec{r}_{i-1}, \vec{E}_{i-1} \rightarrow \vec{r}_i, \vec{E}_i) = C(\vec{E}_{i-1} \rightarrow \vec{E}_i|\vec{r}_{i-1})T(\vec{r}_{i-1} \rightarrow \vec{r}_i|E_i), \quad (3.13)$$

or,

$$\kappa_{i-1} = C_{i-1}T_{i-1}. \quad (3.14)$$

The kernel C_{i-1} denotes the probabilities of particles that are coming out of a collision in $\vec{r}_{i-1}$ with direction $\vec{\Omega}$ and energy E_i, i.e., $\vec{E}_i = \vec{\Omega}_i E_i$. The collision kernel can be represented explicitly as,

$$C(\vec{E}_{i-1} \rightarrow \vec{E}_i|\vec{r}_{i-1}) = \sum_j p_j C_j(\vec{E}_{i-1} \rightarrow \vec{E}_i|\vec{r}_{i-1}), \quad (3.15)$$

where p_j denotes the probability of scattering collision of type j, and C_j is the corresponding collision kernel. Each C_j can be normalized to the mean number of secondaries, ν, per event,

$$\int C_j(\vec{E}_{i-1} \rightarrow \vec{E}_i|\vec{r}_{i-1})d\vec{E}_i = \nu_j. \tag{3.16}$$

For elastic scattering events, $\nu = 1$; for fission $\nu > 1$. The probabilities p_j can be written as,

$$p_j = \frac{\Sigma_{sj}(\vec{r}_{i-1}, \vec{E}_{i-1})}{\Sigma_t(\vec{r}_{i-1}, \vec{E}_{i-1})}, \tag{3.17}$$

where Σ_{sj} is the macroscopic scattering cross section for scattering type j.
The kernel T_{i-1} represents the probability for the transport of particles from $\vec{r}_{i-1}$ to the next collision in $\vec{r}_i$. For example, if $\Sigma_t(\vec{r}_{i-1}, \vec{E}_i)$, the total macroscopic cross section at $(\vec{r}_{i-1}, \vec{E}_i)$, is spatially constant along the direction $\vec{\Omega}_i$ then,

$$T(\vec{r}_{i-1} \rightarrow \vec{r}_i|E_i) = \Sigma_t(E_i)exp(-\Sigma_t(E_i)d), \tag{3.18}$$

where d is the distance from $\vec{r}_{i-1}$ to $\vec{r}_i$.

To develop expressions for correlated sampling tracking, we will denote the transport kernel of the unperturbed system by,

$$\kappa_i^{up} = \kappa(u_i, u_{i+1}; \Sigma_x^{up}), \tag{3.19}$$

and that for the perturbed system by,

$$\kappa_i^{p} = \kappa(u_i, u_{i+1}; \Sigma_x^{p}). \tag{3.20}$$

Σ_x^{up} and Σ_x^p denote a generic cross section for the unperturbed and the perturbed systems, respectively, and the perturbation in cross section can be expressed as,

$$\Delta\Sigma_x = \Sigma_x^p - \Sigma_x^{up}. \tag{3.21}$$

Now the collision densities for the unperturbed and the perturbed systems are respectively,

$$\varphi^{up}(x) = Q(x) + \sum_{n=1}^{\infty} \int_V \cdots \cdots \int_V (\prod_{i=1}^{n} \kappa_i^{up} du_i) Q(u_1), \tag{3.22}$$

and

$$\varphi^{p}(x) = Q(x) + \sum_{n=1}^{\infty} \int_V \cdots \cdots \int_V (\prod_{i=1}^{n} \kappa_i^{p} du_i) Q(u_1). \tag{3.23}$$

The difference between the two collision densities is a function of the cross section change $\Delta\Sigma_x$. As shown before, independent simulation of the unperturbed and perturbed systems and straightforward subtraction of the results is not sufficient for calculating perturbation effects, especially for small perturbations. In the correlated sampling method, the perturbed histories are forced to follow the same trajectories as the unperturbed histories including the same transition points in phase space [Rie84]. A weight factor is used for the perturbed histories to account for the resulting biasing due to the forced transition. The weight factor for the perturbed system is given by,

$$F^p(u_{i+1}, u_i; \Delta\Sigma_x) = F_i^p = \frac{\kappa_i^p}{\kappa_i^{up}}. \tag{3.24}$$

Now the perturbation effect is given by,

$$\Delta\varphi(x; \Delta\Sigma_x) = \varphi^p(x) - \varphi^{up}(x) =$$

$$\sum_{n=1}^{\infty} \int_V \cdots \cdots \int_V (\prod_{i=0}^{n} F_i^p - 1)[\prod_{i=1}^{n} \kappa_i^{up} du_i Q(u_1)]. \tag{3.25}$$

We notice in the expression for $\Delta\varphi(x)$ that the summation expression for the unperturbed collision density, $\varphi^{up}(x)$, has been multiplied by the weight factor

$$(\prod_{i=0}^{n} F_i^p - 1).$$

$\Delta\varphi$ is tallied at each collision point and contributes to the calculation of the perturbation.

3.3.1 Correlated Sampling in a Reference System

The correlated sampling technique described in the previous section assumes that the actual simulation is done in the unperturbed system and the adjusting weight factors F_i^ps ($= \frac{\kappa_i^p}{\kappa_i^{up}}$) account for the simulation of the correlated perturbed particles. The weight carried by the perturbed particles would be a product of these adjusting weight factors F_i^ps. This requires that the adjusting weight factors for the correlated game must always remain finite, which implies,

$$\kappa_i^p > 0, \kappa_i^{up} > 0. \tag{3.26}$$

In some actual simulations these conditions may not be satisfied. For example, in calculating perturbation effects due to voids, a particular phase-space transition probability may vanish in the perturbed system but exist in the unperturbed one. To account for this difficulty, a Monte Carlo perturbation approach has been developed [Rie88] where the actual simulation is done in an artificial reference system different from both the unperturbed and perturbed systems. The probability density functions describing the transition process in the reference system are a weighted mean of the probability density functions of the unperturbed and perturbed systems. The important characteristic of the reference system is that all particle reactions that can take place in the unperturbed and perturbed systems will also have a non-zero probability in the reference system. Thus, for example, a voided region would not be admissible in the reference system if the unperturbed or perturbed systems were not voided. However, the transpose is acceptable, i.e. one could have a voided region in the perturbed system, in which case the weights would be appropriately modified if a collision occurred in the "voided" region. The responses for the unperturbed and perturbed systems are calculated by correlating them to the reference system's

simulation. Also by the proper choice of the reference (ref) system, it is possible to avoid large fluctuations of the adjusting weight factors, $\frac{\kappa_i^{up}}{\kappa_i^{ref}}$ and $\frac{\kappa_i^{p}}{\kappa_i^{ref}}$, and reduce uncertainty of the differential effect. The reference system [Maj94] may be chosen freely, but it also depends on the type of problem solved [Lux89]. Since the choice of the reference system will have an effect on the variance, it may be subject to optimization methods[Lar95]. This by itself leads to a topic of future research. In appendix A, some preliminary computational parametric studies are performed to observe the effect of different reference systems on the result of a perturbation problem. In this research work, the arithmetic average of the parameters of the unperturbed and perturbed systems has been used as the parameters for the reference system. One of the requirements of the reference system is that $\frac{\kappa_i^{up}}{\kappa_i^{ref}}$ and $\frac{\kappa_i^{p}}{\kappa_i^{ref}}$ should be as constant as possible. We will now explicitly derive the adjusting weight factors for the unperturbed and perturbed systems and introduce the concept of a δ-scatter in the forward direction to avoid large fluctuations in the adjusting weight factors.

The cross sections for the reference, unperturbed and perturbed systems are Σ_x^{ref}, Σ_x^{up} and Σ_x^{p}, respectively, where x denotes a different cross section type. According to our choice of the reference system, we have,

$$\Sigma_x^{ref} = \frac{1}{2}[\Sigma_x^{up} + \Sigma_x^{p}]. \tag{3.27}$$

The spatial collision distance, d, is sampled from $\Sigma_t^{ref} exp(-\Sigma_t^{ref} d)$ in the reference system. Then the adjusting weight factors (WF) for the unperturbed and perturbed systems are given by,

$$WF^{up} = \frac{\Sigma_t^{up} exp(-\Sigma_t^{up} d)}{\Sigma_t^{ref} exp(-\Sigma_t^{ref} d)} = \frac{\Sigma_t^{up}}{\Sigma_t^{ref}} exp[(\Sigma_t^{ref} - \Sigma_t^{up})d], \tag{3.28}$$

and

$$WF^{p} = \frac{\Sigma_t^{p} exp(-\Sigma_t^{p} d)}{\Sigma_t^{ref} exp(-\Sigma_t^{ref} d)} = \frac{\Sigma_t^{p}}{\Sigma_t^{ref}} exp[(\Sigma_t^{ref} - \Sigma_t^{p})d], \tag{3.29}$$

respectively. The weight which is carried along by the unperturbed and perturbed particles is the product of these adjusting weight factors, WF^{up} and WF^{p}, respectively.

Now, assuming $\Sigma_t^p > \Sigma_t^{up}$, the adjusting weight factor for the unperturbed history has a positive exponent and may assume large values if d is large. Similarly, the factor for the perturbed history may assume a very small value. To avoid these large fluctuations, a δ-scatter in the forward direction is added to all three total cross sections [Rie88]. Introduction of a pseudo-collision with a δ-scatter in the forward direction increases the number of collisions without changing the expectation values. For each of these three cases the δ-scatter is chosen to yield the same total cross section plus δ-scatter :

$$\Sigma_t^{ref} + \delta^{ref} = \Sigma_t^{up} + \delta^{up} = \Sigma_t^{p} + \delta^{p}. \tag{3.30}$$

Using these δ-scatters,

$$WF^{up} = \frac{\Sigma_t^{up} + \delta^{up}}{\Sigma_t^{ref} + \delta^{ref}} exp[(\Sigma_t^{ref} + \delta^{ref} - \Sigma_t^{up} - \delta^{up})d] = 1, \tag{3.31}$$

and

$$WF^{p} = \frac{\Sigma_t^{p} + \delta^{p}}{\Sigma_t^{ref} + \delta^{ref}} exp[(\Sigma_t^{ref} + \delta^{ref} - \Sigma_t^{p} - \delta^{p})d] = 1. \tag{3.32}$$

Thus the large fluctuations in WF^{up} and WF^{p} are avoided. The reference system's cross sections are chosen using equation (3.27) and a δ-scatter for the reference system (δ^{ref}) is selected. Next equation (3.30) can be used to determine the δ-scatters for the unperturbed and perturbed systems.

3.3.2 Variance Reduction Using δ-scatter

Variance reduction techniques can be applied to the reference system's particles. We have applied survival biasing and Russian roulette [Lew93] to the reference system's particles and correlated the unperturbed and perturbed systems' particles to

the reference system's particles. After determining the site of collision, the survival chance of the reference system particle is sampled from,

$$p_s^{ref} = \frac{\Sigma_s^{ref} + \delta^{ref}}{\Sigma_t^{ref} + \delta^{ref}}, \tag{3.33}$$

where Σ_s^{ref} is the macroscopic scattering cross section of the reference system. Next, the survival chance of the unperturbed and perturbed particles are sampled from,

$$p_s^{up} = \frac{\Sigma_s^{up} + \delta^{up}}{\Sigma_t^{ref} + \delta^{ref}}, \tag{3.34}$$

and

$$p_s^{p} = \frac{\Sigma_s^{p} + \delta^{p}}{\Sigma_t^{ref} + \delta^{ref}}, \tag{3.35}$$

respectively. The weights of the reference, unperturbed and perturbed particles are modified at the site of the j^{th} collision as follows,

$$W_j^{ref} = W_{j-1}^{ref} p_s^{ref}, \tag{3.36}$$

$$W_j^{up} = W_{j-1}^{up} p_s^{up}, \tag{3.37}$$

$$W_j^{p} = W_{j-1}^{p} p_s^{p}. \tag{3.38}$$

After the weight adjustment, Russian roulette is applied to the reference system particles. In this procedure, one checks to determine if the reference system particle weight has fallen below some minimum value, in which case the reference system particle as well as the unperturbed and perturbed particles are terminated. In this research work we have simulated isotropically scattering systems. After each collision event the new angular distribution for a neutron is chosen isotropically.

3.3.3 Correlated Sampling and Source Iteration

Most Monte Carlo codes determine the eigenvalue of a system by the source iteration method. Application of the source iteration method encounters difficulties

when used to calculate eigenvalue perturbations because perturbed weights are propagated from one generation to another. Experience has shown [Rie89, Rie84, Gal95] that the statistical fluctuations in the propagated weights grow considerably, and for most cases any useful information regarding the perturbation is lost over many generations. We have solved the same problems given in table 3.1 by applying the correlated sampling technique to the source iteration method of eigenvalue calculation. The actual Monte Carlo simulation is done in the reference system, with cross sections determined as the average of the unperturbed and perturbed systems. As discussed earlier, a δ-scatter cross section is added to the total cross section for each of the reference, unperturbed, and perturbed systems to avoid fluctuations in the adjusting weight factors. For illustration, the value of δ^{ref} is given along with the perturbation results in table 3.2. The standard deviation of a single generation ΔK is given by,

$$\sigma_s = \sqrt{\frac{\sum_{n=1}^{I_a} \Delta K_n^2}{I_a - 1} - \frac{\left(\sum_{n=1}^{I_a} \Delta K_n\right)^2}{I_a(I_a - 1)}} \quad , \tag{3.39}$$

where I_a is the active number of fission generations up to and including the current generation. The standard deviation of the mean is,

$$\sigma = \frac{\sigma_s}{I_a{}^{\frac{1}{2}}} \quad , \tag{3.40}$$

and is provided with the numerical Monte Carlo results in table 3.2.

Unperturbed cross sections: Σ_t=1.0 cm^{-1}, $\nu\Sigma_f$=0.11 cm^{-1}, Σ_s=0.9 cm^{-1}			
Problem #	Perturbed cross sections	TWODANT ΔK	Monte Carlo ΔK
1	$\Delta\Sigma_t$=.001, $\Delta\Sigma_a$=.001	-.008807	-.007401 $\mp$.12E-4 δ^{ref}=.0005
2	$\Delta\Sigma_t$=.0001, $\Delta\Sigma_a$=.0001	-.000888	-.0007462$\mp$.12E-5 δ^{ref}=.0001
3	$\Delta\Sigma_f$=.0001	.000903	.00082298 $\pm$.78E-6 δ^{ref}=0.

Table 3.2: Perturbation Results Using Source Iteration and Correlated Sampling.

Similar to the table 3.1 results, the Monte Carlo runs for table 3.2 are for 30 inactive batches, 70 active batches, and 2000 neutrons per batch. We observe from table 3.2 that for the problems analyzed, the source iteration method has difficulty estimating the differential effect in eigenvalue. As mentioned before in section 3.2.1, the problems of table 3.2 will be solved (in section 3.3.4) using the CSFM method and will provide significantly improved results.

3.3.4 Correlated Sampling Fission Matrix Method

We have shown in previous sections of this chapter that either (i) subtracting two independent Monte Carlo simulations or (ii) combining source iteration and correlated sampling fails to estimate small perturbation effects with reasonable accuracy. Several references [Rie88, Sei91, Gal95] point out that for the calculation of Monte Carlo perturbation effects it is necessary to utilize the fission matrix method to

perform the eigenvalue calculation. To that end, two fission matrix equations,

$$K^{up}S^{up} = A^{up}S^{up}, \tag{3.41}$$

and

$$K^{p}S^{p} = A^{p}S^{p}, \tag{3.42}$$

for the unperturbed and perturbed systems respectively are needed to formulate and solve for the eigenvalue perturbations:

$$\Delta K = K^{p} - K^{up}. \tag{3.43}$$

In the above equations, K is the eigenvalue, S is the fission source vector, and A is the fission matrix. One approach [Sei91, Gal95] utilizes the linearity of the transport equation and splits the fission matrix into two parts:

$$A^{up} = A_0 + A_1. \tag{3.44}$$

The fission matrix A_0 is formed by the particles that do not go through a perturbed cell during a random walk simulation, whereas the fission matrix A_1 is formed by the particles that arrive from any perturbed cell. Since the perturbation is confined to the perturbed cells, the matrix A_0 is equal in both the unperturbed and perturbed problems. The matrix A_1 is evaluated only for the perturbed problem. During particle tracking in the unperturbed system, all necessary information for particles that enter any perturbed cell are saved. At the end of the simulation in the unperturbed system, this information is then used to calculate to the matrix A_1 for the perturbed system.

We have taken a different approach in which the correlated sampling technique is applied to the fission matrix method [Rie88]. The actual Monte Carlo simulation

is done in an artificial reference system with cross sections that are linear combination of the unperturbed and perturbed systems. The δ-scatter cross sections in the forward direction are used for the reference, unperturbed, and perturbed systems as explained in section 3.3.1 of this chapter. For the determination of the fission matrix of the reference system, the first generation is started with an assumed source distribution and $\mathrm{W}^{ref} = 1$, where W^{ref} is the weight of a reference system particle. Likewise, the initial weights of the unperturbed (W^{up}) and perturbed (W^{p}) particles are also set to unity. Particle tracking is carried out only in the reference system and the weights of the unperturbed and perturbed systems are modified by multiplying them with appropriate adjusting weight factors, WF^{up} and WF^{p}, respectively, during the simulation. At the end of the first generation, source normalization is done for the reference system particles to stabilize the neutron population. The source normalization has no direct effect on other systems, except that the next generation particles in the unperturbed and perturbed systems start out with the same weights as that of the reference system particles. For the second generation, the fission neutron production distribution obtained from the first generation is used for the reference system particles. The particles in the unperturbed and perturbed systems also use the same source distribution as that of the reference system for the second generation. This process is continued for specified number of generations. A few of the initial generations are discarded to avoid bias due to the initial source guess.

Now the fission rate is determined. Recall from our discussion on the fission matrix algorithm of chapter 2 that the fission rate is determined by the probability that a particle starting in volume element l generates $\mathrm{a}_{l,m}$ particles in element m, where $a_{l,m}$ is an element of the fission matrix A. For the reference system, the matrix

element $a_{l,m}^{ref}$ is scored as,

$$a_{l,m}^{ref} = a_{l,m}^{ref} + W^{ref} \frac{[\nu^{ref} \Sigma_f^{ref}]}{[\Sigma_t^{ref} + \delta^{ref}]} \quad , \tag{3.45}$$

where ν^{ref} is the number of particles emerging from a fission reaction in the reference system and Σ_f^{ref} is the macroscopic fission cross-section for the reference system. At the same time, the fission matrix elements for the unperturbed and perturbed fission matrices, A^{up} and A^{p}, are estimated as,

$$a_{l,m}^{up} = a_{l,m}^{up} + W^{up} \frac{[\nu^{up} \Sigma_f^{up}]}{[\Sigma_t^{ref} + \delta^{ref}]} \quad , \tag{3.46}$$

and

$$a_{l,m}^{p} = a_{l,m}^{p} + W^{p} \frac{[\nu^{p} \Sigma_f^{p}]}{[\Sigma_t^{ref} + \delta^{ref}]} \quad . \tag{3.47}$$

The dominant eigenvalues K^{ref}, K^{up}, and K^{p} of matrices A^{ref}, A^{up} and A^{p}, respectively, are determined numerically, after which ΔK can be calculated as,

$$\Delta K = K^{p} - K^{up} \quad . \tag{3.48}$$

As before, the standard deviation of the single generation ΔK is given by,

$$\sigma_s = \sqrt{\frac{\sum_{n=1}^{I_a} \Delta K_n^2}{I_a - 1} - \frac{(\sum_{n=1}^{I_a} \Delta K_n)^2}{I_a (I_a - 1)}} \tag{3.49}$$

and the standard deviation of the mean is given by

$$\sigma = \frac{\sigma_s}{I_a^{\frac{1}{2}}} \quad . \tag{3.50}$$

This σ is provided with the numerical Monte Carlo results. Variance reduction schemes are applied as explained in section 3.3.2 of this chapter. The perturbation problems of table 3.1 and 3.2 are now solved with the combined CSFM technique as described above. These results are shown in table 3.3. Similar to table 3.1 and 3.2,

the Monte Carlo runs for table 3.3 utilize 30 inactive batches, 70 active batches and 2000 neutrons per batch.

Unperturbed cross sections: Σ_t=1.0 cm^{-1}, $\nu\Sigma_f$=0.11 cm^{-1}, Σ_s=0.9 cm^{-1}			
Problem #	Perturbed cross sections	TWODANT ΔK	Monte Carlo ΔK
1	$\Delta\Sigma_t$=.001, $\Delta\Sigma_a$=.001	-.008807	-.0089069 $\mp$.82E-5 δ^{ref}=.0005
2	$\Delta\Sigma_t$=.0001, $\Delta\Sigma_a$=.0001	-.000888	-.00089601$\mp$.89E-6 δ^{ref}=.0001
3	$\Delta\Sigma_f$=.0001	.000903	.00090325 $\pm$.51E-6 δ^{ref}=0.

Table 3.3: Perturbation Results Using Fission Matrix and Correlated Sampling.

Comparing the results of table 3.1 and table 3.2 with that of table 3.3 we observe that the CSFM technique provides significantly improved results compared with direct subtraction or the source iteration method with correlated sampling.

3.3.5 Multigroup Energy Transfer

For energy dependent problems with multigroup cross sections, the scattering matrix for the reference system may be determined by taking the average of the unperturbed and perturbed scattering matrices. For example, suppose the scattering matrices for the unperturbed and perturbed systems are given respectively by,

$$\Sigma_s^{up} = \begin{pmatrix} \Sigma_{11}^{up} & \Sigma_{12}^{up} & \ldots \\ \Sigma_{21}^{up} & \Sigma_{22}^{up} & \ldots \\ & \ldots\ldots\ldots & \end{pmatrix}, \tag{3.51}$$

and

$$\Sigma_s^p = \begin{pmatrix} \Sigma_{11}^p & \Sigma_{12}^p & \ldots \\ \Sigma_{21}^p & \Sigma_{22}^p & \ldots \\ & \ldots\ldots\ldots & \end{pmatrix} \quad . \tag{3.52}$$

Then the scattering matrix for the reference system is,

$$\Sigma_s^{ref} = \frac{1}{2}\Big(\Sigma_s^{up} + \Sigma_s^p \Big). \tag{3.53}$$

For multigroup problems, survival biasing is applied at the site of the collision to the reference, unperturbed and perturbed particles as explained in section 3.3.2. Then the reference scattering matrix is used to determine the outgoing energy group of the reference particle, using the probability of a reference particle to scatter from energy group i to j :

$$p_{ij}^{ref} = \frac{\Sigma_{12}^{ref}}{\Sigma_{11}^{ref} + \Sigma_{12}^{ref}}. \tag{3.54}$$

The unperturbed and perturbed particles are constrained to follow the same energy group transfer as the reference particle.

3.4 Numerical Results

In this section, numerical results are given for different eigenvalue perturbation problems, utilizing the combined Monte Carlo CSFM method. The Monte Carlo ΔK results are compared to that of the discrete ordinates code TWODANT [O'De82]. The TWODANT code is used to calculate the unperturbed and perturbed eigenvalues separately; these are then subtracted to determine ΔK. The TWODANT code results for slab geometry are generated with the S_{32} quadrature set and for X-Y geometry with the S_{16} quadrature set. The inner and outer iteration convergence criteria are set to 10^{-12}. The cross sections of the reference systems for all test

problems are chosen as the average of that of the unperturbed and perturbed systems. The percentage error given with all the results is calculated as follows:

$$\%error = \left(\frac{TWODANT(\Delta K) - MonteCarlo(\Delta K)}{TWODANT(\Delta K)} \right) \%. \tag{3.55}$$

Test Problem 1: heterogeneous slab, one energy group.

This case is a one energy group heterogeneous slab problem with vacuum boundary conditions on both ends. Dimensions and unperturbed cross sections for the slab are shown in figure 3.1. A perturbation is made in the 0.4 cm region in the middle. The ΔK results for two perturbations are shown in table 3.4 along with the corresponding TWODANT ΔKs. The Monte Carlo runs utilize 140 active batches, 60 inactive batches and 8000 neutrons per batch. The standard deviation of the eigenvalue of the reference system is 0.71E-3 for the first problem of table 3.4.

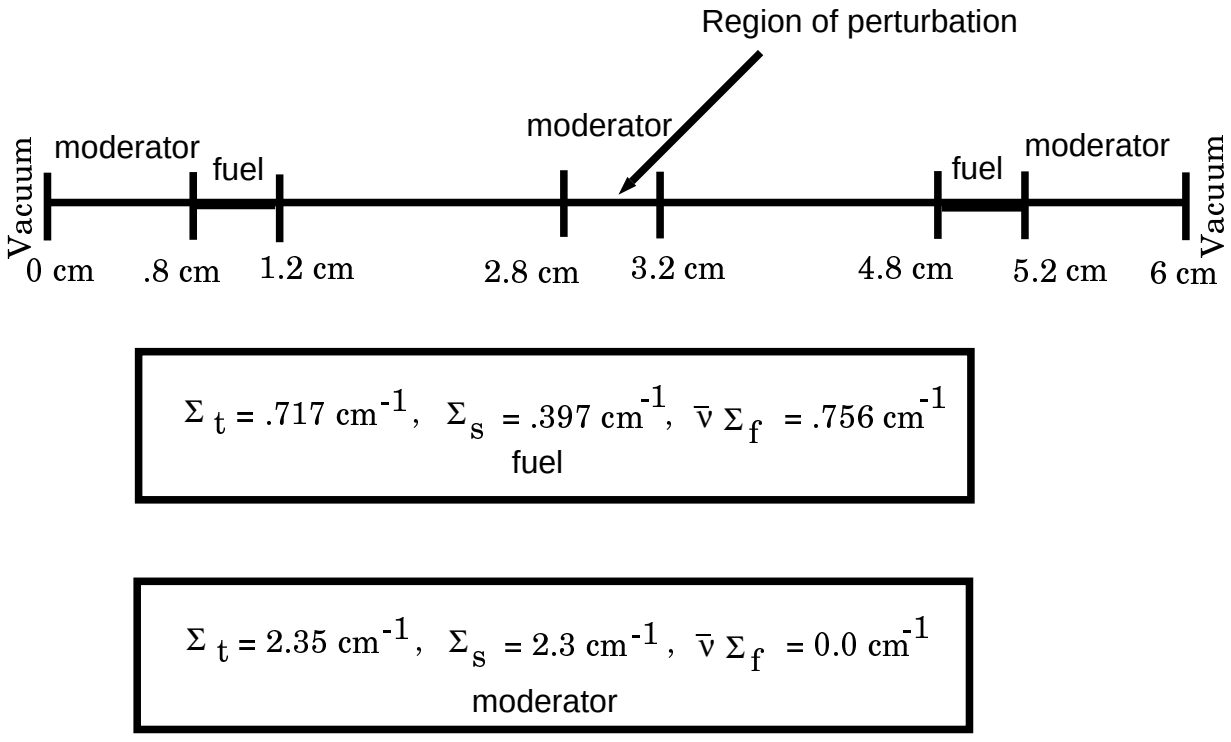


Figure 3.1: Configuration of Heterogeneous Slab for Test Problem 1.

Heterogeneous slab, Σ_t=.717, Σ_a=.32, $\nu\Sigma_f$=.756 Σ_t=2.35, Σ_a=0.05,$\nu\Sigma_f$=0.0	.8≤X≤1.2 all other X	T=6.0cm and 4.8≤X≤5.2	
Perturbation in middle 0.4cm	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\Sigma_t$=.15, $\Delta\Sigma_a$=0.149, $\Delta\Sigma_s$=.001	-0.019438	-0.018991∓ .61E-4 δ^{ref}=.15	2.3
$\Delta\Sigma_t$=.05, $\Delta\Sigma_a$=0.049, $\Delta\Sigma_s$=.001	-0.007613	-0.007509∓ .24E-4 δ^{ref}=.025	1.4

Table 3.4: Perturbation Results for Test Problem 1 (heterogeneous slab, 1 group).

Test Problem 2a: homogeneous X-Y geometry, one energy group.

This case is a one energy group homogeneous X-Y geometry problem with vacuum boundary conditions on all sides. Dimensions and unperturbed cross sections for the problem are shown in figure 3.2. Cross section perturbations are made over the entire square region. The ΔK results for three different perturbations are shown in table 3.5, along with the corresponding TWODANT ΔKs. The first Monte Carlo perturbation case utilizes 70 active batches, 30 inactive batches, and 2000 neutrons per batch, while the second and third cases utilize 70 active batches, 30 inactive batches, and 4000 neutrons per batch.

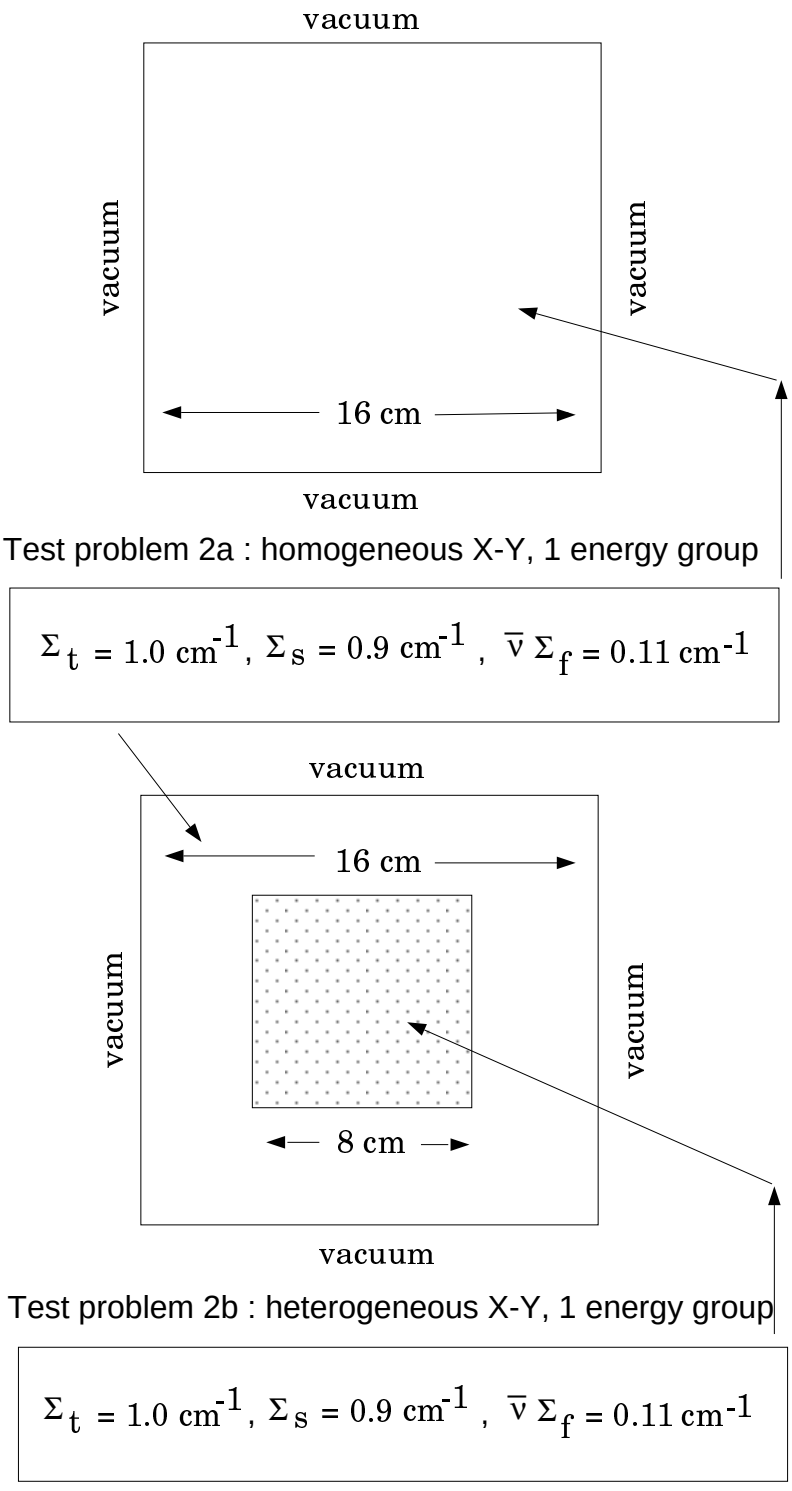


Figure 3.2: Configurations of X-Y geometry for Test Problems 2a and 2b.

Unperturbed Σ_s=0.9,	cross-sections Σ_a=0.1,	Σ_t=1.0, $\nu\Sigma_f$=0.11	
Perturbed cross-sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_f$= -.002	-0.016475	-0.016469∓.16E-4 δ^{ref}=0	.04
$\Delta\Sigma_t$=.001,$\Delta\Sigma_a$=.001	-0.0072755	-0.0074001∓.76E-5 δ^{ref}=.0005	1.7
$\Delta\Sigma_t$= .002, $\Delta\Sigma_a$= .042 $\Delta\Sigma_s$=-.04,$\Delta\nu\Sigma_f$=.039	0.005958	0.00585±.11E-3 δ^{ref}= .00105	1.8

Table 3.5: Perturbation Results for Test Problem 2a (homogeneous X-Y, 1 group).

Test Problem 2b: heterogeneous X-Y geometry, one energy group.

This case is a one energy group heterogeneous X-Y geometry problem with vacuum boundary conditions on all sides. Dimensions and unperturbed cross sections for the problem are shown in figure 3.2. Cross section perturbations are made in the middle square region. The ΔK results for three different perturbations are shown in table 3.6, along with the corresponding TWODANT ΔKs. The first Monte Carlo perturbation case utilizes 70 active batches, 30 inactive batches and 4000 neutrons per batch, while the second and third Monte Carlo perturbation case employ 100 active batches, 40 inactive batches and 4000 neutrons per batch.

Unperturbed Σ_s=0.9,	cross-sections Σ_a=0.1,	Σ_t=1.0, $\nu\Sigma_f$=0.11	
Perturbed cross-sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_f$= .0009	0.0044562	0.0043765±.92E-5 δ^{ref}=0	1.8
$\Delta\Sigma_t$=.001,$\Delta\Sigma_a$=.001	-0.0044909	-0.0043577 ∓.88E-5 δ^{ref}=.001	3.0
$\Delta\Sigma_t$= .002, $\Delta\Sigma_a$= .042 $\Delta\Sigma_s$=-.04,$\Delta\nu\Sigma_f$=.039	0.003884	0.003781 ± .89E-4 δ^{ref}= .0015	2.6

Table 3.6: Perturbation Results for Test Problem 2b (heterogeneous X-Y, 1 group).

Test Problem 3a: homogeneous X-Y geometry, two energy group.

This case is a two energy group homogeneous X-Y geometry (10 cm X 10 cm) problem with vacuum boundary conditions on all sides. Cross section perturbations are done over the whole square region. Unperturbed and perturbed cross sections and ΔK results for a perturbation case is shown in table 3.7 along with the TWODANT ΔKs. The Monte Carlo perturbation case has 110 active batches, 30 inactive batches and 2000 neutrons per batch.

Unperturbed	cross-sections		
Σ_{t1}=1.0,	$\nu\Sigma_{f1}$=.11,	Σ_{s11}=.9,	Σ_{s12}=.07
Σ_{t2}=2.11,	$\nu\Sigma_{f2}$=2.44,	Σ_{s21}=0	Σ_{s22}=.98
Perturbed cross-sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\Sigma_{t1}$=.001,$\Delta\Sigma_{a1}$ = .001	0.122134	0.12152±.14E-3 δ^{ref}=.001	0.5

Table 3.7: Perturbation Results for Test Problem 3a (homogeneous X-Y, 2 group).

Test Problem 3b: Millstone Reactor Assembly.

This problem is a 10 cm X 10 cm square region with reflecting boundary conditions on all sides. The unperturbed cross section correspond to a Millstone [NRC85] reactor fuel assembly with 2.9 weight percent (w/o) enrichment, without burnable poison (bp) pins at hot full power condition with 1398 ppm critical boron concentration, while the perturbed cross section correspond to a 2.9 w/o, 20 burnable poison assembly at the same conditions. These cross sections were generated using the CPM-2 [Jon87] code. Cross section perturbations are made over the entire square region. The cross sections are given in table 3.8 and the ΔK result is given in table 3.9. The Monte Carlo run utilizes 140 active batches, 60 inactive batches, and 6000 neutrons per batch. The standard deviation of the reference system eigenvalue is 0.16E-3.

0 bp : Σ_{t1}=.25285,	$\nu\Sigma_{f1}$=.00642,	Σ_{s11}=.22674,	Σ_{s12}=.01678
Σ_{t2}=.85205,	$\nu\Sigma_{f2}$=.12351,	Σ_{s21}=0	Σ_{s22}=.76202
20 bp : Σ_{t1}=.24303,	$\nu\Sigma_{f1}$=.00642,	Σ_{s11}=.21684,	Σ_{s12}=.01602
Σ_{t2}=.79575,	$\nu\Sigma_{f2}$=.12377,	Σ_{s21}=0	Σ_{s22}=.69020

Table 3.8: Millstone 2.9 w/o Two Group Cross Sections for Test Problem 3b.

TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
-0.165136	$-0.159771 \mp$ .32E-4 δ^{ref1}=.006,δ^{ref2}=.03	3.2

Table 3.9: Perturbation Results for Test Problem 3b.

Test Problem 4: variation of test problem 1.

This is the same as test problem 1, except that the 0.4 cm region of moderator in the middle is replaced with an absorber. Unperturbed cross sections of the absorber are Σ_t = 5.0 cm^{-1}, and Σ_s = 0.1 cm^{-1}. Perturbed cross sections of the absorber

are $\Sigma_t = 5.0$ cm^{-1}, and $\Sigma_s = 0.3$ cm^{-1}. The results are shown in table 3.10. The Monte Carlo runs utilize 140 active batches, 60 inactive batches and 8000 neutrons per batch.

Heterogeneous slab,		T=6.0cm	
Σ_t=.717, Σ_a=.32, $\nu\Sigma_f$=.756	.8≤X≤1.2	and 4.8≤X≤5.2	
Σ_t=5.0, Σ_a=4.9,$\nu\Sigma_f$=0.0	2.8≤X≤3.2		
Σ_t=2.35, Σ_a=0.05,$\nu\Sigma_f$=0.0	all other X		
Perturbation in middle 0.4cm	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\Sigma_a$=-0.2, $\Delta\Sigma_s$=0.2	0.00031	0.0002674∓ .35E-5 δ^{ref}=0.0	13.7

Table 3.10: Perturbation Results for Test Problem 4 (heterogeneous slab, 1 group).

Test Problem 5: source convergence problem.

This test problem specifically looks into the case where the shapes of the unperturbed and perturbed scalar fluxes are significantly different. The unperturbed problem is a 16 mfp one group homogeneous slab with $\Sigma_t = 1.0$, $\Sigma_s = 0.9$, and $\nu\Sigma_f = 0.11$ and vacuum boundary conditions on both ends. For the perturbed problem the middle 4 cm region of the slab is replaced with a material representing moderator with $\Sigma_t = 2.35$, $\Sigma_s = 2.3$, and $\nu\Sigma_f = 0.0$. The perturbed and unperturbed scalar fluxes from TWODANT are shown in figure 3.3.

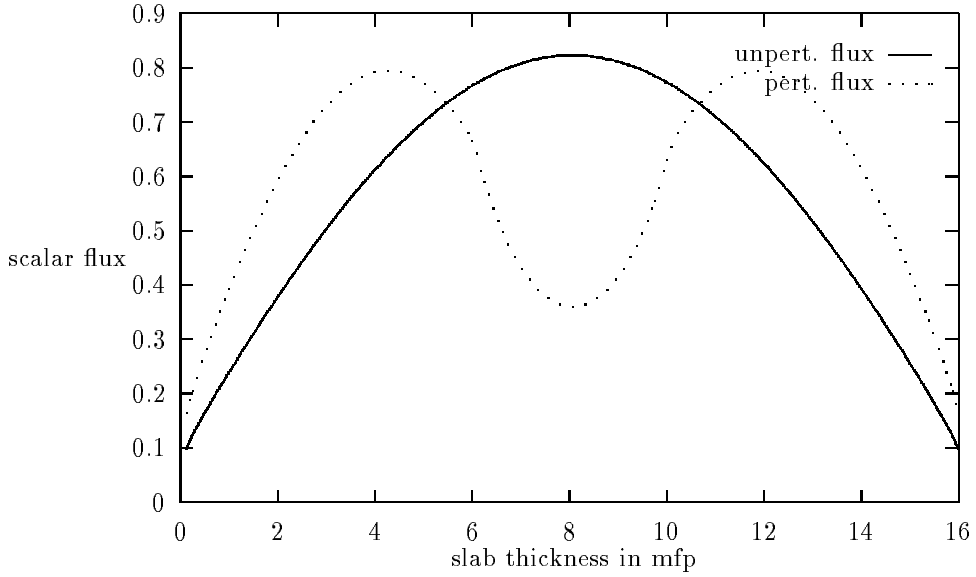


Figure 3.3: Perturbed and Unperturbed Scalar Fluxes from TWODANT.

Results of TWODANT and CSFM Monte Carlo ΔKs are shown in table 3.11. The Monte Carlo case has 140 active batches, 60 inactive batches and 8000 neutrons per batch.

TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
-0.168909	-0.16695$\mp$ .29E-3 δ^{ref}=1.35	1.2

Table 3.11: Perturbation Results for Test Problem 5.

3.5 Summary and Discussion of Numerical Results

The results of table 3.1 show that subtracting the results of two independent Monte Carlo simulations, to compute small perturbation effects can encounter problems. We have shown in section 3.3.3 that the Monte Carlo source iteration method, when combined with the correlated sampling technique, also has difficulty estimating small perturbation results. This led us to combine the correlated sampling technique with the fission matrix approach for eigenvalue calculations. The results of table 3.3 show that the combined CSFM approach can provide significantly improved results compared with direct subtracting or the source iteration method combined with correlated sampling.

In section 3.4 we have shown various numerical results for eigenvalue perturbation problems using the CSFM method. We have compared these Monte Carlo ΔKs to that of the TWODANT code. For test problems 1, 2a, 2b, 3a, and 3b we observe that ΔKs calculated with the Monte Carlo and S_N methods always agree within 4% of each other. Even though we have not included the standard deviation of the reference eigenvalue for all the test problems (except for problems of table 3.4 and 3.9), this standard deviation is always approximately an order of magnitude higher than that of the ΔK. Extension to multigroup problems was straightforward and multigroup problems have shown good agreement with the corresponding TWODANT results also.

We notice that the error for test problem 4 is 13.7% which is significantly larger than that of other test problems of section 3.4. If we compare the configuration of test problem 4 with that of test problem 1 we observe that in test problem 4 the middle 0.4 cm region has been replaced by a strong absorber material. In both

problems perturbation was done in the middle 0.4 cm region. Test problem 1 shows 2.3% error, whereas test problem 4 shows 13.7% error. Insertion of a strong absorber in the middle region of the slab, converts the test problem 4 into a loosely coupled system and hence there is less communication of neutrons between the two sides of the absorber region. We believe that special variance reduction schemes are needed to account for the lack of neutron communication for loosely coupled systems.

In test problem 5 we have specifically looked into the case where the unperturbed scalar flux differs significantly in shape from the perturbed scalar flux as shown in figure 3.3. The reference system is again chosen as the arithmetic average of the unperturbed and perturbed systems. The purpose of this test problem is to see the effect of fission source convergence between the reference, unperturbed and perturbed systems. Since the scalar flux shapes are significantly different for the unperturbed and perturbed systems, fission sources for the two systems would need to converge to these two different shapes. The fission source of the reference system would converge to a source shape that is in-between the unperturbed and perturbed fission source shapes. From the results of table 3.11 we observe that for this test problem the error in ΔK is approximately 1%. This relatively small error in ΔK implies that the three fission sources for the reference, unperturbed and perturbed systems converged to their respective correct fission sources even though according to figure 3.3 the converged fission source shapes are different for these systems.

In the CSFM method the actual Monte Carlo eigenvalue simulation is done in the reference system. The fission sources of the unperturbed and perturbed systems are correlated to that of the reference system. For every generation the fission neutrons for the reference system are started from the previous generation of fission source distribution of the reference system. No information regarding the fission source

distribution of the unperturbed or the perturbed system is carried onto the next generation. It is possible that there would be an inherent bias in ΔK due to this lack of information for the unperturbed and perturbed fission source distributions. Test problems of this chapter (and of chapter 4) show that this effect is not significant. Even though the ΔK results are accurate, it does not imply that the unperturbed and perturbed Ks converge to the correct values. It is possible that the unperturbed and perturbed Ks have biased values, but due to the positive correlation of these Ks the ΔK results are still accurate. We believe that more study might be necessary to clearly define the limits where the CSFM method would encounter difficulties.

CHAPTER IV

MULTIPLE EIGENVALUE PERTURBATIONS

4.1 Introduction

In this chapter the idea of the CSFM technique is extended to calculate multiple perturbations in the eigenvalue of the Boltzmann transport equation with a single Monte Carlo simulation. The actual Monte Carlo simulation is done in an artificial reference system different from the unperturbed and all perturbed systems. The simulations of the unperturbed and all perturbed systems are correlated to the simulation of the reference system. This allows the determination of multiple ΔKs with a single Monte Carlo simulation, yielding a significant reduction in computational effort.

4.2 Multiple Reactivity Calculation

Let us now consider multiple ΔKs (ΔK_i, i = 1,2,3,...,N) due to multiple small perturbations of the same unperturbed system [Maj95a], where the N perturbed systems are denoted by p_i, i = 1,2,3,...,N :

$$\Delta K_i = K^{p_i} - K^{up} \quad ; i = 1, 2, 3, ..., N. \tag{4.1}$$

4.2.1 Reference System

We have chosen to define the reference system by,

$$\Sigma_x^{ref} = \frac{1}{(N+1)}\left(\Sigma_x^{up} + \sum_{i=1}^{N}\Sigma_x^{p_i}\right) \quad ; i = 1, 2, 3, ..., N, \tag{4.2}$$

where x denotes a different cross section type. It is possible to choose a different reference system; the optimum choice may vary from one problem to another. For all the numerical results shown in this chapter we have used equation (4.2) to determine the reference system. The Monte Carlo particle tracking is done in this reference system. The distance to collision d is sampled from ${\Sigma_t}^{ref} exp(-\Sigma_t^{ref} d)$. The adjusting weight factors for the unperturbed and all perturbed systems are then given respectively by,

$$WF^{up} = \frac{{\Sigma_t}^{up}}{{\Sigma_t}^{ref}} exp({\Sigma_t}^{ref} - {\Sigma_t}^{up})d, \tag{4.3}$$

and

$$WF^{p_i} = \frac{{\Sigma_t}^{p_i}}{{\Sigma_t}^{ref}} exp({\Sigma_t}^{ref} - {\Sigma_t}^{p_i})d \quad ; i = 1, 2, 3, ..., N. \tag{4.4}$$

4.2.2 Forward δ-scatter

As with the single perturbation case, a δ-scatter cross section is added to the total cross sections of the reference, unperturbed and perturbed systems. The δ-scatter cross section for the reference system (δ^{ref}) is chosen, depending upon the problem, such that,

$$\delta^{ref} \geq \mid \Sigma_t^{ref} - max(\Sigma_t^{p_i}) \mid \quad ; if\ max(\Sigma_t^{p_i}) > \Sigma_t^{up}, \tag{4.5}$$

or,

$$\delta^{ref} \geq \mid \Sigma_t^{ref} - \Sigma_t^{up} \mid \quad ; if\ \Sigma_t^{up} > max(\Sigma_t^{p_i}). \tag{4.6}$$

The δ-scatter cross section for the unperturbed system (δ^{up}) and perturbed systems (δ^{p_i}, i = 1,2,3,...,N) are chosen as,

$$\delta^{up} = \Sigma_t^{ref} + \delta^{ref} - \Sigma_t^{up}, \tag{4.7}$$

and,

$$\delta^{p_i} = \Sigma_t^{ref} + \delta^{ref} - \Sigma_t^{p_i}. \tag{4.8}$$

The conditions imposed by equation (4.5) and (4.6) ensure that all the δ-scatter cross sections are nonnegative. In our test problems the above four equations are used to determine forward δ-scatters for the reference, unperturbed and all perturbed systems. The distance to collision d in the reference system is sampled from $({\Sigma_t}^{ref} + \delta^{ref})exp(-(\Sigma_t^{ref} + \delta^{ref})d)$, and the modified biasing factors for the unperturbed and all perturbed systems become,

$$WF^{up} = \frac{({\Sigma_t}^{up} + \delta^{up})exp(-(\Sigma_t^{up} + \delta^{up})d)}{({\Sigma_t}^{ref} + \delta^{ref})exp(-(\Sigma_t^{ref} + \delta^{ref})d)} = 1, \tag{4.9}$$

and,

$$WF^{p_i} = \frac{({\Sigma_t}^{p_i} + \delta^{p_i})exp(-(\Sigma_t^{p_i} + \delta^{p_i})d)}{({\Sigma_t}^{ref} + \delta^{ref})exp(-(\Sigma_t^{ref} + \delta^{ref})d)} = 1. \tag{4.10}$$

This avoids large fluctuations in WF^{up} and WF^{p_i}.

4.2.3 ΔK Calculation

All of the simulation procedures described in section 3.3.4 and 3.3.5 apply to multiple ΔK calculation and hence will not be repeated here. Only the additional computations necessary for multiple ΔK calculation will be described in this section. Instead of a single weight adjusting factor W^p for a single perturbed system, now we have multiple weight adjusting factors, W^{p_i}s (i=1,2,...,N), for the multiple perturbed systems. For the first fission generation, the starting weights of all the perturbed particles are set to unity. For the fission matrix eigenvalue calculation in the reference

system, the first generation is started with an assumed source distribution and W^{ref} = 1, where W^{ref} is the weight of a reference system particle. During the random walk simulation, the weight of each perturbed particle is modified by multiplying them with an appropriate adjusting weight factor WF^{p_i}.

The fission matrix elements for the perturbed fission matrix, A^{p_i}, is scored as,

$$a_{l,m}^{p_i} = a_{l,m}^{p_i} + W^{p_i} \frac{[\nu^{p_i} \Sigma_f^{p_i}]}{[\Sigma_t^{ref} + \delta^{ref}]} \quad ; i = 1, 2, 3, ...N. \tag{4.11}$$

The dominant eigenvalues K^{ref}, K^{up} and K^{p_i} of matrices A^{ref}, A^{up} and A^{p_i} respectively, are determined numerically. Then, multiple ΔKs due to multiple perturbations (p_i, i = 1,2,3,...,N) are calculated as,

$$\Delta K^i = K^{p_i} - K^{up} \quad ; i = 1, 2, ..., N. \tag{4.12}$$

Biasing factors for variance reduction of all the perturbed particles are:

$$p_s^{p_i} = \frac{\Sigma_s^{p_i} + \delta^{p_i}}{\Sigma_t^{ref} + \delta^{ref}} \quad ; i = 1, 2, ...N. \tag{4.13}$$

The weights of all the perturbed particles are reduced as follows at the site of the j^{th} collision,

$$W_j^{p_i} = W_{j-1}^{p_i} p_s^{p_i} \quad ; i = 1, 2, ...N. \tag{4.14}$$

For multigroup problems, the scattering matrix of the reference system is,

$$\Sigma_s^{ref} = \frac{1}{N+1} \left(\Sigma_s^{up} + \sum_{i=1}^{N} \Sigma_s^{p_i} \right) \quad ; i = 1, 2, 3, .., N; \tag{4.15}$$

where,

$$\Sigma_s^{p_i} = \begin{pmatrix} \Sigma_{11}^{p_i} & \Sigma_{12}^{p_i} & ... \\ \Sigma_{21}^{p_i} & \Sigma_{22}^{p_i} & ... \\ & ..... & \end{pmatrix} \quad ; i = 1, 2, 3...N. \tag{4.16}$$

For multigroup problems, survival biasing is applied to the reference, unperturbed and all perturbed particles. Then the unperturbed and all perturbed particles follow the same energy group transfer as the reference particle.

4.3 Numerical Results

We have used the approach described in previous sections of this chapter to solve multiple eigenvalue perturbation problems from a single Monte Carlo simulation. Monte Carlo ΔKs are compared to that of the discrete ordinate code TWODANT [O'De82]. To generate N ΔKs, due to N perturbations, the TWODANT code was run (N+1) times, whereas the Monte Carlo code needed to be run only once. For some of the test problems, we also compare the wall clock time required to compute one ΔK versus N ΔKs using the Monte Carlo approach. For comparison of wall clock timings we made an effort to choose dedicated machines and hence are able to show timing results only for the test problems that were simulated on dedicated machines. This comparison shows how much reduction in computational effort is achieved from the multiple Monte Carlo perturbation approach. Cross sections of the reference systems for all the test problems are chosen as the average of that of the unperturbed and all perturbed systems, i.e., according to equation (4.2). Then the reference systems δ-scatter was chosen according to equation (4.5) or (4.6). δ-scatters for the unperturbed and all perturbed systems are then calculated from equation (4.7) and (4.8). Value of δ^{ref} is given for each case with the perturbation result. The following different test problems, similar to chapter 3 test problems, are studied using the multiple Monte Carlo perturbation approach:

- Test problem 1: homogeneous slab, one energy group
- Test problem 2: heterogeneous slab, one energy group
- Test problem 3: homogeneous X-Y geometry, one energy group
- Test problem 4: heterogeneous X-Y geometry, one energy group

- Test problem 5: homogeneous slab, two energy group
- Test problem 6: homogeneous X-Y geometry two energy group (Millstone assembly cross sections)

Test problem 1: homogeneous slab, one energy group

This case is a one energy group homogeneous slab problem with vacuum boundary conditions on both ends. Dimension and unperturbed cross sections for the slab are shown in table 4.1. Cross section perturbations were done over the entire slab. Calculated results of two ΔKs from a single Monte Carlo simulation are also shown in table 4.1, along with the corresponding TWODANT results. The Monte Carlo runs utilize 40 inactive batches, 100 active batches, and 2000 neutrons per batch.

Homogeneous slab, Unperturbed Σ_t=1.0 cm^{-1},	1 group, cross sections: $\nu\Sigma_f$=0.11 cm^{-1},	T=16 cm Σ_s=0.9 cm^{-1}	
Perturbed cross sections	TWODANT ΔK	Two correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_f$=.001	0.00903	0.0090338±.48E-5	.04
$\Delta\nu\Sigma_f$=-.001	-0.009028	-0.0090338∓.48E-5 (δ^{ref}=0)	.06
$\Delta\Sigma_t$=.1,$\Delta\Sigma_a$=.1	-0.469124	-0.47711∓.35E-3	1.7
$\Delta\Sigma_t$=.0001,$\Delta\Sigma_a$=.0001	-0.000888	-0.00091015∓.81E-6 (δ^{ref}=.06668)	2.5
$\Delta\Sigma_t$=.001,$\Delta\Sigma_a$=.001	-0.008807	-0.0088790 ∓.75E-5	0.8
$\Delta\Sigma_t$=.0001,$\Delta\Sigma_a$=.0001	-0.000888	-0.00089511∓.75E-6 (δ^{ref}=.00068)	0.8
$\Delta\Sigma_t$=.0001,$\Delta\Sigma_a$=.0001	-0.000888	-0.00089633∓.71E-6	0.9
$\Delta\Sigma_t$=-.0001,$\Delta\Sigma_a$=-.0001	0.000889	0.00089795∓.71E-6 (δ^{ref}=.00015)	1.1

Table 4.1: Perturbation Results for Test Problem 1 (two correlated ΔKs).

Test problem 2: heterogeneous slab, one energy group

This is the same problem as test problem 1 of chapter 3. Calculated results of two ΔKs from a single Monte Carlo simulation are shown in table 4.2 along with the corresponding TWODANT ΔKs. The Monte Carlo results utilize 50 inactive batches, 100 active batches, and 6000 neutrons per batch.

Heterogeneous slab, T=6.0cm Σ_t=.717, Σ_a=.32, $\nu\Sigma_f$=.756 Σ_t=2.35, Σ_a=0.05,$\nu\Sigma_f$=0.0	1 group; .8≤X≤1.2 all other X	and 4.8≤X≤5.2	
Perturbation in middle 0.4cm	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\Sigma_t$=.15, $\Delta\Sigma_a$=0.149, $\Delta\Sigma_s$=.002	-0.019339	-0.019024 ∓.80E-4	1.6
$\Delta\Sigma_t$=.05, $\Delta\Sigma_a$=0.049, $\Delta\Sigma_s$=.001	-0.007613	-0.007510 ∓.31E-4 δ^{ref}=.08838	1.3

Table 4.2: Perturbation Results for Test Problem 2 (two correlated ΔKs).

Test problem 3: homogeneous X-Y geometry, one energy group

This is test problem 2a of chapter 3. Calculated results of three ΔKs from a single Monte Carlo simulation are shown in table 4.3 along with the corresponding TWODANT results. The first three Monte Carlo perturbations cases utilize 30 inactive batches, 70 active batches, and 2000 neutrons per batch, while the last case utilize 40 inactive batches, 160 active batches, and 2000 neutrons per batch.

Homogeneous X-Y geometry, 1 group			
Perturbed cross sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_f$=-0.002	-0.016475	-0.016493∓.15E-4	.11
$\Delta\nu\Sigma_f$=0.0008	0.0065902	0.0065973±.63E-5	.11
$\Delta\nu\Sigma_f$=0.011	0.090615	0.090713±.87E-4	.11
		δ^{ref}=0.0	
$\Delta\Sigma_t$=0.0005,$\Delta\Sigma_a$=0.0005	-0.0036524	-0.0037185∓.54E-5	1.8
$\Delta\Sigma_t$=0.001,$\Delta\Sigma_a$=0.001	-0.007276	-0.007406∓.10E-4	1.8
$\Delta\Sigma_t$=0.0001,$\Delta\Sigma_a$=0.0001	-0.0007328	-0.0007461∓.11E-5	1.8
		δ^{ref}=0.00061	
$\Delta\Sigma_a$=-0.00008,$\Delta\Sigma_s$=0.00008	0.00059756	0.00059825±.94E-6	.12
$\Delta\Sigma_a$=-0.0001,$\Delta\Sigma_s$=0.0001	0.0007471	0.0007479±.12E-5	.11
$\Delta\Sigma_a$=0.0001,$\Delta\Sigma_s$=-0.0001	-0.0007459	-0.0007467∓.12E-5	.11
		δ^{ref}=0.0	
$\Delta\Sigma_t$=0.0003,$\Delta\Sigma_s$=0.007	0.026820	0.026802±.38E-4	.07
$\Delta\Sigma_a$= -0.0067,$\Delta\nu\Sigma_f$=-0.003			
$\Delta\Sigma_t$=0.002,$\Delta\Sigma_s$=-0.04	0.005958	0.00582±.11E-3	2.3
$\Delta\Sigma_a$=0.042,$\Delta\nu\Sigma_f$=0.039			
$\Delta\Sigma_t$=0.0001,$\Delta\Sigma_s$=-0.003	0.0095791	0.0095781±.84E-5	.01
$\Delta\Sigma_a$=0.0031,$\Delta\nu\Sigma_f$=0.004			
		δ^{ref}=0.00141	

Table 4.3: Perturbation Results for Test Problem 3 (three correlated ΔKs).

Test problem 4: heterogeneous X-Y geometry, one energy group

This is test problem 2b of chapter 3. Calculated results of three ΔKs from a single Monte Carlo simulation are shown in table 4.4 along with the TWODANT ΔKs. The Monte Carlo cases have 40 inactive batches, 160 active batches, and 2000 neutrons per batch.

Heterogeneous X-Y geometry, 1 group			
Perturbed cross sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_f$=-0.002	-0.009909	-0.009663$\mp$.23E-4	2.5
$\Delta\nu\Sigma_f$=0.0008	0.0039554	0.0039079$\pm$.94E-5	1.2
$\Delta\nu\Sigma_f$=0.011	0.056820	0.05566$\pm$.13E-3	2.0
		δ^{ref}=0.0	
$\Delta\Sigma_t$=0.0025,$\Delta\Sigma_a$=0.0025	-0.011035	-0.010755$\mp$.27E-4	2.5
$\Delta\Sigma_t$=0.001,$\Delta\Sigma_a$=0.001	-0.0044901	-0.004356$\mp$.11E-4	3.0
$\Delta\Sigma_t$=0.002,$\Delta\Sigma_a$=0.002	-0.008872	-0.008639 $\mp$.21E-4	2.6
		δ^{ref}=0.001125	
$\Delta\Sigma_t$=0.001,$\Delta\Sigma_s$=0.007	0.012602	0.012283$\pm$.36E-4	2.5
$\Delta\Sigma_a$= -0.006,$\Delta\nu\Sigma_f$=-0.003			
$\Delta\Sigma_t$=0.0015,$\Delta\Sigma_s$=-0.04	0.005765	0.00572$\pm$.11E-3	0.8
$\Delta\Sigma_a$=0.0415,$\Delta\nu\Sigma_f$=0.039			
$\Delta\Sigma_t$=0.0011,$\Delta\Sigma_s$=-0.003	0.001373	0.001365$\pm$.13E-4	.54
$\Delta\Sigma_a$=0.0041,$\Delta\nu\Sigma_f$=0.004			
		δ^{ref}=0.00061	

Table 4.4: Perturbation Results for Test Problem 4 (three correlated ΔKs).

Test problem 5: homogeneous slab, two energy group

This case is for two energy group homogeneous slab problem with vacuum boundary conditions on both ends. The slab is 10 cm thick and the unperturbed cross sections for the slab are shown in table 4.5. Cross section perturbations were done over the entire slab. Calculated results of two ΔKs from a single Monte Carlo simulation are also shown in table 4.5 along with the corresponding TWODANT results. The Monte Carlo results have 30 inactive batches, 100 active batches, and 2000 neutrons per batch.

Homogeneous slab, 2 group			
Σ_{t1}=1.0,	$\nu\Sigma_{f1}$=.11,	Σ_{s11}=.9,	Σ_{s12}=.07
Σ_{t2}=2.11,	$\nu\Sigma_{f2}$=2.44,	Σ_{s21}=0	Σ_{s22}=.98
Perturbed cross-sections	TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
$\Delta\nu\Sigma_{f1}$=.02	0.1602	0.16007±.30E-3	.08
$\Delta\nu\Sigma_{f2}$=.04	0.3203	0.32015±.60E-3	.05
		δ^{ref1}=0.0,δ^{ref2}=0.0	
$\Delta\nu\Sigma_{f1}$=.02,$\Delta\Sigma_{a1}$ = .001	0.1427	0.14265 ±.24E-3	.04
$\Delta\nu\Sigma_{f2}$=.04	0.3203	0.32124±.55E-3	.3
		δ^{ref1}=0.001,δ^{ref2}=0.0	
$\Delta\nu\Sigma_{f1}$=.005,$\Delta\Sigma_{a1}$ = .001	0.0235	0.023199 ±.33E-4	1.3
$\Delta\nu\Sigma_{f2}$=.04	0.3203	0.32050±.51E-3	.06
		δ^{ref1}=0.005,δ^{ref2}=0.0	

Table 4.5: Perturbation Results for Test Problem 5 (two correlated ΔKs).

Test problem 6: homogeneous X-Y geometry (Millstone assemblies' cross sections), two energy group

This is the same as test problem 3b of chapter 3. The unperturbed cross section is for 2.9 w/o, 0 bp Millstone [NRC85] reactor assembly and the perturbed cross sections are for 2.9 w/o, 20 bp and 2.9 w/o, 24 bp assemblies. The cross sections are given in table 4.6 and the ΔK results are given in table 4.7. The Monte Carlo runs utilize 40 inactive batches, 100 active batches, and 3000 neutrons per batch.

0 bp : Σ_{t1}=.25285,	$\nu\Sigma_{f1}$=.00642,	Σ_{s11}=.22674,	Σ_{s12}=.01678
Σ_{t2}=.85205,	$\nu\Sigma_{f2}$=.12351,	Σ_{s21}=0	Σ_{s22}=.76202
20 bp : Σ_{t1}=.24303,	$\nu\Sigma_{f1}$=.00642,	Σ_{s11}=.21684,	Σ_{s12}=.01602
Σ_{t2}=.79575,	$\nu\Sigma_{f2}$=.12377,	Σ_{s21}=0	Σ_{s22}=.69020
24 bp : Σ_{t1}=.24117,	$\nu\Sigma_{f1}$=.00642,	Σ_{s11}=.21500,	Σ_{s12}=.01580
Σ_{t2}=.78395,	$\nu\Sigma_{f2}$=.12387,	Σ_{s21}=0	Σ_{s22}=.67520

Table 4.6: Two Group Cross Sections (2.9 w/o) of Millstone Assemblies.

Homogeneous	X-Y geometry,	2 group
TWODANT ΔK	Correlated Monte Carlo ΔK	error (%)
-0.165136	-0.159668∓.54E-4	3.3
-0.194535	-0.187647∓.62E-4 δ^{ref1}=0.008,δ^{ref2}= 0.045	3.5

Table 4.7: Perturbation Results for Test Problem 6 (Millstone cross sections, two correlated ΔKs).

4.4 Discussion of Numerical Results

Various eigenvalue perturbation problems have been solved in the previous section using the multiple perturbation CSFM approach. The methodology has worked well for both small and large perturbations, even though the method is specifically meant for problems where the perturbation is small and hence direct Monte Carlo subtraction encounters difficulty. Results of multiple ΔKs, from a single Monte Carlo simulation, have compared well with the corresponding ΔKs computed by the TWODANT code. Comparing the relative errors of ΔKs between the TWODANT code and the multiple perturbation CSFM Monte Carlo approach, we observe that the results always agree within less than 4% for all test cases shown here. Since the relative errors represent comparison of ΔK results between the CSFM method and the TWODANT code, an agreement within 4% can be considered to be quite accurate. Further observation of all the tabulated results in section 4.3 shows that

in general, relative errors of ΔKs are slightly larger when perturbations in Σ_t were made, compared to perturbations in cross sections other than Σ_t. This is due to the fact that a perturbation in Σ_t, in addition to biasing the fission reactions, also forces the CSFM method to bias the weight adjusting factors (WF) for every distance to collision sampled. On the other hand, a perturbation in the $\nu\Sigma_f$ alone requires biasing in the fission reactions only. We have not noticed any general trend in the deterioration of results in going from homogeneous to heterogeneous problems or from one energy group to two energy group problems. This suggests that the two energy group CSFM method can be easily extended to multigroup problems. The reference system for each test problem was chosen using equation (4.2) and the δ-scatter cross sections were chosen using the conditions given in equations (4.5), (4.6), (4.7) and (4.8). These equations provided reasonable choices for both the reference system and δ-scatter cross sections, and can easily be extended to multiple groups.

Since multiple ΔKs due to multiple perturbations are computed from a single Monte Carlo simulation, a significant reduction in computational effort is achieved. We provide two actual wall clock timing results in table 4.8 to show this. These timing results were obtained on a dedicated HP700 series machine. From the timing results of table 4.8 and various other timing results obtained on non-dedicated machines (and hence not shown here), we can conclude that it requires less than 10% extra computational effort for each additional ΔK calculation compared to the first ΔK calculation. Most of this extra computational effort is due to the number of times the matrix iterative algorithm is invoked. However, the computational time spent in the matrix iterative algorithm is relatively modest compared to the time spent in Monte Carlo particle tracking.

Problem Type	# of ΔK calculated	Wall Clock time (sec)
First problem of table 4.3	one	1185.5
	three	1419.4
Second problem of table 4.3	one	1117.1
	three	1359.3

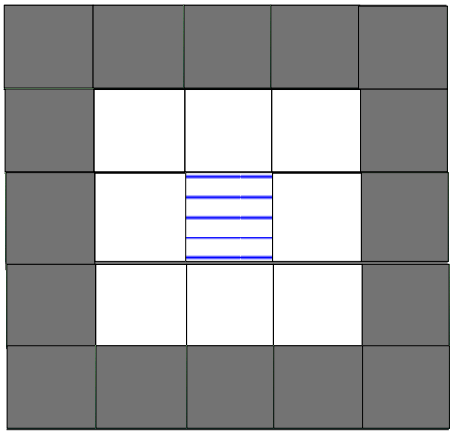
Table 4.8: Timing Results for Multiple ΔK Calculations.

4.5 Multiple K Calculation Using CSFM

We have shown in the previous sections of this chapter that multiple ΔKs can be calculated from a single Monte Carlo simulation with good accuracy using the CSFM approach. In this section we will utilize this to calculate multiple Ks and show the applicability of this method for design calculations. Our purpose is to apply the CSFM method to evaluate eigenvalues of multiple systems that closely resemble each other, i.e., all the different systems are slightly perturbed versions of a single system. This single system can be referred to as the unperturbed system. We assume that the eigenvalue of the unperturbed system is determined with arbitrary accuracy from either a S_N or a Monte Carlo calculation or perhaps an analytical solution if the unperturbed system is simple enough. The CSFM technique can then be applied to compute multiple ΔKs relative to the unperturbed system's eigenvalue. Using the known K of the unperturbed system and the multiple ΔKs computed by the CSFM method, we can evaluate the absolute Ks of the multiple perturbed systems.

The test problem is a 5cm X 5cm square region with one group cross sections and vacuum boundaries. The square region is divided into 25 square cells of dimension

1cm X 1cm. The 16 cells along the boundaries represent fuel cells, and the center cell represents poison. Rest of the 8 cells represent moderator. The objective of this design problem is to observe the effect of varying the scattering ratio of the poison material on the eigenvalue of the system. The geometric configuration and cross sections for the problem are shown in figure 4.1.



Fuel : $\Sigma_t = .717$, $\Sigma_s = .397$, $\nu\Sigma_f = .756$

Mod : $\Sigma_t = 2.35$, $\Sigma_s = 2.3$

Poison : $\Sigma_t = 4.0$, $\Sigma_s = 0.1$

Cross sections for three different scattering ratios :

Poison : $\Sigma_t = 4.0$, $\Sigma_s = 0.7$

Poison : $\Sigma_t = 4.0$, $\Sigma_s = 1.3$

Poison : $\Sigma_t = 4.0$, $\Sigma_s = 1.9$

Figure 4.1: Dimension and Cross Sections for K Calculation Problem.

According to a fine mesh (S_{16}) TWODANT calculation, with inner and outer

iteration convergence criteria of 10^{-12}, the K for the unperturbed system is 1.03006. Next, the CSFM method is used to compute the three ΔKs corresponding to the three different scattering ratios of the poison material at the center cell of the system. Using these three ΔKs and the known K of the unperturbed system, we evaluate the three Ks. These results are shown in table 4.9. The Monte Carlo runs utilize 100 inactive batches, 260 active batches, and 4000 neutrons per batch. We have also calculated the three eigenvalues corresponding to the three different scattering ratios of the poison material using fine mesh S_{16} TWODANT simulations. These TWODANT results are also shown in table 4.9 along with relative errors $\left(\frac{TWODANT(K)-MonteCarlo(K)}{TWODANT(K)}\right)$. The Monte Carlo eigenvalues agree within one tenth of a percent with the fine mesh TWODANT eigenvalues. This example shows application of the CSFM approach for design problems.

Cases	TWODANT K	Monte Carlo K	error (%)
Case 0: poison scattering ratio = $\frac{0.1}{4.0}$	1.03006	1.03006	
Case 1: poison scattering ratio = $\frac{0.7}{4.0}$	1.031031	1.031005$\pm$.13E-4	0.003
Case 2: poison scattering ratio = $\frac{1.3}{4.0}$	1.032291	1.032231$\pm$.27E-4	0.006
Case 3: poison scattering ratio = $\frac{1.9}{4.0}$	1.034017	1.033905$\pm$.41E-4	0.01

Table 4.9: Results of K Calculation Problem.

CHAPTER V

MONTE CARLO PARTICLE TRANSPORT ALGORITHMS ON PARALLEL PROCESSORS

5.1 Introduction

Monte Carlo particle transport is an inherently parallel computational method that has been successfully implemented on diverse computational platforms such as vector processors, shared memory and distributed memory Multiple Instruction, Multiple Data (MIMD) parallel processors, multiple vector processors, and Single Instruction, Multiple Data (SIMD) parallel processors. Vectorizing a Monte Carlo algorithm requires a major change in the algorithm and coding; but the performance gain due to vectorization can be significant and once vectorized it is relatively easy to parallelize across multiple vector processors. Parallelizing a Monte Carlo code is relatively easy, and the continuing decrease in the costs of massively parallel processors (MPPs) makes it attractive to adapt to a MPP. In this chapter we describe the effort to adapt two classes of Monte Carlo particle transport algorithms on different parallel machines. We provide details of the two parallel algorithms, including modifications to the random number generator, and will present wall clock timing results on different parallel architectures. The first algorithm is intended for a fixed source application and simulates photon transport in a high temperature plasma.

The second algorithm is intended for the calculation of eigenvalues and perturbations in eigenvalues for neutron transport problems. The following constraint has been imposed on both parallel algorithms - *the Monte Carlo simulation should yield identical results for runs with the same number of tasks, independent of the number of CPUs which are actually executing these tasks.* This is a generalization of the principle of reproducibility, an essential characteristic of particle transport Monte Carlo codes.

5.2 Parallel Photon Transport Algorithm

The xPHOT [Mar86] series of codes have been used for a number of years to examine alternative algorithms for Monte Carlo particle transport on a variety of computer architectures, including vector supercomputers and parallel processors. These codes simulate the transport of photons in a high temperature plasma, and include realistic opacity data and a general two-dimensional mesh. The most recent version, TPHOT, allows investigation of methods for parallelizing time-dependent Monte Carlo codes.

The basic algorithm that we have employed is the *pool of tasks* algorithm, where the MASTER task partitions the workload into tasks, including the necessary data to perform the simulation. These tasks, denoted the SLAVE tasks, are dispatched to processors as they become available to process a task. The partitioning of workload is done within each time step, for this time-dependent photon transport code, to allow for non-Monte Carlo computation between time steps, such as to change the material properties or modify the mesh. Therefore the overall simulation within a time step is divided into N tasks, where each task has approximately $\frac{1}{N}$ of the total workload.

The partitioning of the particles to balance the workload is done statistically among the tasks. The SLAVE task follows the requisite number of particles, accumulating partial tallies and other statistics as needed by the problem, and reports the results back to the MASTER task, which accumulates the partial tallies into overall global tallies. The MASTER task reads the input file, constructs the necessary arrays to describe the geometry and material properties (i.e., cross sections), and transmits these data to the SLAVE tasks. For shared memory parallel processors, such as the Cray YMP, these data are simply stored in the global COMMON, but for a distributed memory parallel processor, this data is sent in a message to each processor to be stored in its local memory. Figure 5.1 illustrates the basic parallel algorithm used in TPHOT.

The overall workload can be divided into four categories: (1) pre-processing work, (2) serial Monte Carlo, (3) parallel Monte Carlo, and (4) post-processing work. The pre-processing work includes reading the input data and preparing geometry and cross section data arrays, while post-processing includes writing the output results. For our analysis, we define the overall simulation time as the sum of the serial Monte Carlo work performed by the MASTER task before and after the random walks, and the parallel Monte Carlo work performed by the SLAVE tasks for the random walk simulations.

Independent and reproducible sequences of random numbers are generated by the Lehmer [Leh51] tree technique, where the MASTER task generates starting seeds with one generator (the "left" generator) and the SLAVE tasks then generate the actual random sequences for their random walks with a second generator (the "right" generator). In figure 5.2, the black dots represent the seeds produced by the left generator. The white dots represent random numbers generated by each SLAVE's

right generator.

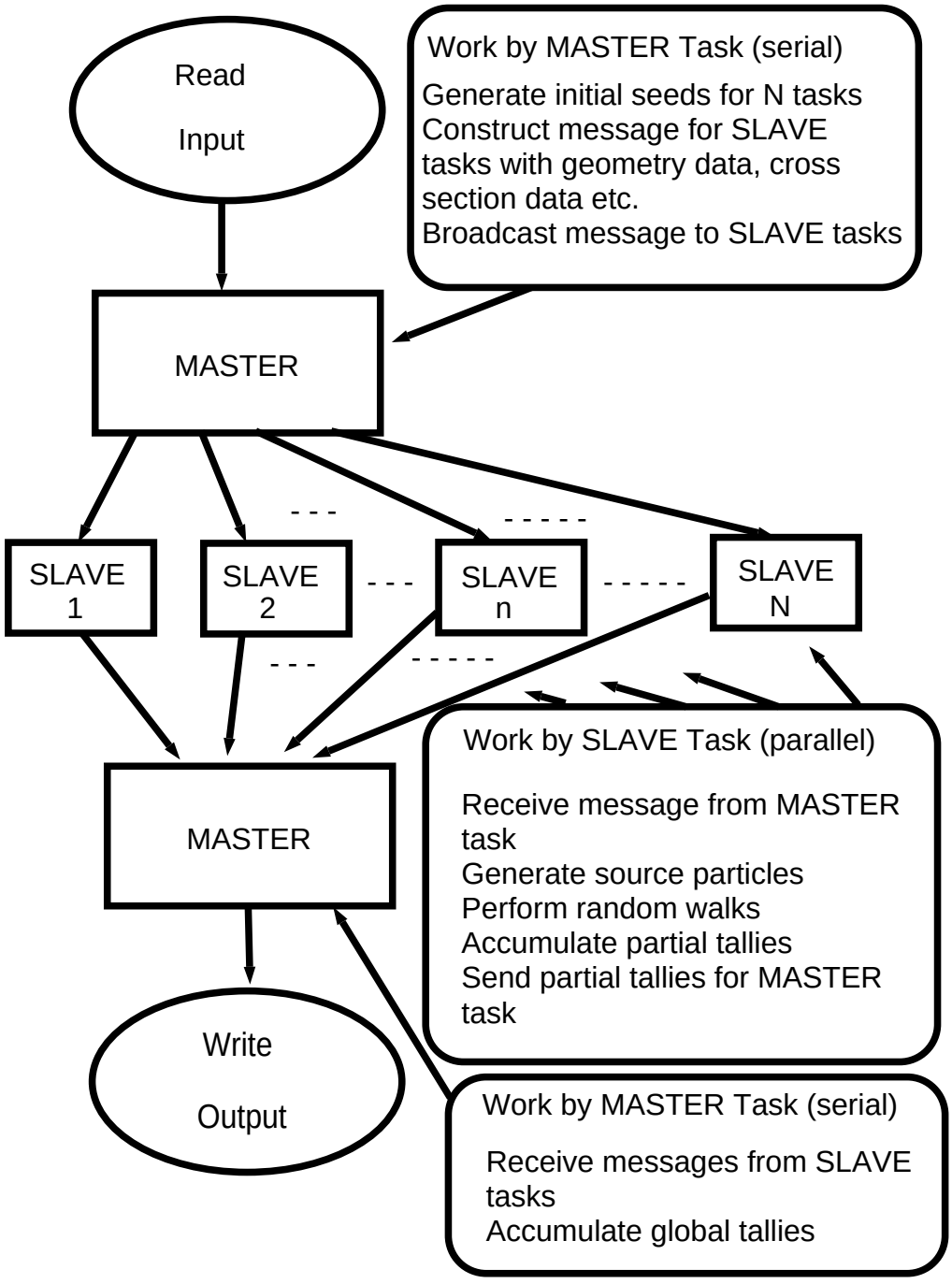


Figure 5.1: Flow Diagram for Fixed Source Parallel Monte Carlo Algorithm.

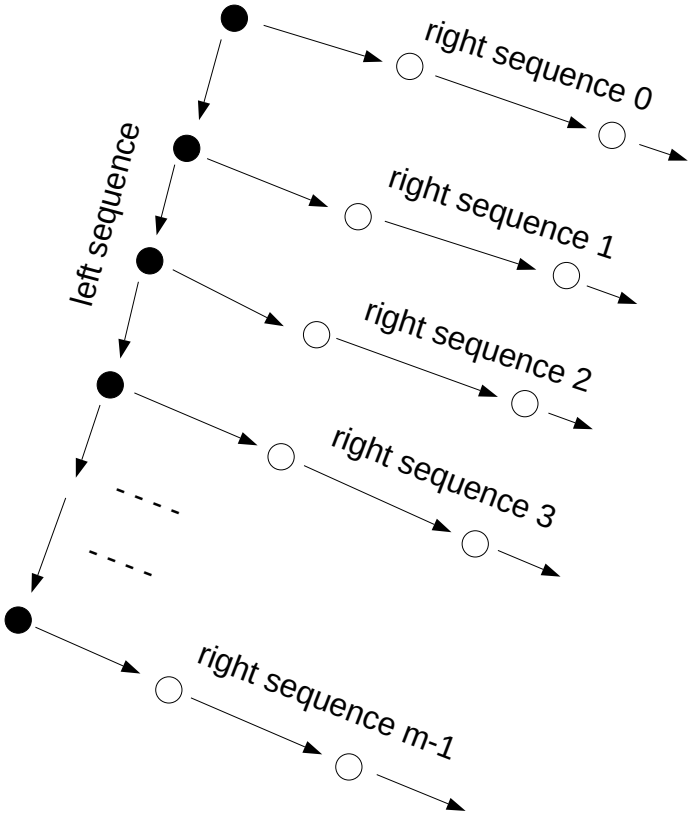


Figure 5.2: Lehmer Tree for Generating Independent Random Number Sequences.

This parallel random number generator was used by researchers [Mar93] for the BBN Butterfly parallel processors. We had difficulty in porting this parallel random number generator to the KSR-1 parallel processor. For the KSR-1 we used the random number generator of the EGS4 [Nel85] code. Each processor ("pthread "in case of the KSR-1) was supplied its own random number seed. These seeds were generated using the *skip ahead* approach, which allowed each processor to generate a sequence of random numbers that did not not overlap any other processor's random number sequence.

5.2.1 Efficiency of Parallel Algorithm

The principal measure of parallelization efficiency is the *speedup*, S_N, defined to be the ratio of the time to execute the computational workload W on a single processor to the time on N processors,

$$S_N = \frac{\tau_1}{\tau_N} \quad , \tag{5.1}$$

where τ_1 is the time to execute the workload on a single processor and τ_N is the time to execute the workload on N processors.

The time τ_N is the sum of three terms,

$$\tau_N = \tau_s + \tau_p + \Delta\tau \quad : \tag{5.2}$$

(1) the time τ_s to execute the serial portion of the workload, (2) the time τ_p to execute the parallel portion, and (3) an additional time $\Delta\tau$ due to the parallelization overhead which is quite general and accounts for any overhead due to implementing the algorithm on the parallel processor, either due to the hardware, the network, the operating system, or the algorithm. We would expect it to be a function of the number of processors, N, as well as the workload, W,

$$\Delta\tau = \Delta\tau(N, W) \quad . \tag{5.3}$$

The *theoretical speedup*, S_N^{th}, is the speedup assuming zero parallelization overhead,

$$S_N^{th} = \frac{\tau_1}{\tau_p + \tau_s} \quad , \tag{5.4}$$

which can be compared to the *true speedup*:

$$S_N = \frac{\tau_1}{\tau_p + \tau_s + \Delta\tau} \quad . \tag{5.5}$$

Note that the theoretical speedup is the best that can be achieved. The ratio of the true speedup to the theoretical speedup is the *parallelization efficiency*,

$$\epsilon_N = \frac{S_N}{S_N^{th}} \quad , \tag{5.6}$$

which is a measure of the efficiency of the parallel processor to execute a given parallel algorithm. Any degradation in performance due to parallelization overhead will result in ϵ_N being less than one.

Now we introduce the *serial fraction* f,

$$f = \frac{W_S}{W} = \frac{W_S}{W_S + W_P} \quad , \tag{5.7}$$

where W is the total computational workload, consisting of the serial workload W_S, and the parallel workload W_p, which can by definition be performed on N processors. We define W in arbitrary units of workload and assume the processor speed is ν (workload unit per second), which leads to the following expression for the theoretical speedup,

$$S_N^{th} = \frac{\tau_1}{\tau_p + \tau_s} = \frac{W/\nu}{\frac{(1-f)W}{N\nu} + \frac{fW}{\nu}} \quad , \tag{5.8}$$

or

$$S_n^{th} = \frac{1}{\frac{(1-f)}{N} + f} \quad . \tag{5.9}$$

For Monte Carlo [Mar93], the total workload for the single processor cases (N=1) can be approximated very well (as will be seen by our timing results) by a simple linear function of the number of histories, N_h, to be simulated:

$$W = a' + b'N_h \quad . \tag{5.10}$$

Now we divide equation (5.10) by ν, which is a constant for a specific processor and define new constants a and b to obtain the single processor execution time:

$$\tau_1 = a + bN_h \quad . \tag{5.11}$$

Since all histories are independent, they can be tracked in parallel. Further, it is reasonable to associate the constant a with the serial time, τ_s. The serial fraction is given by,

$$f = \frac{\tau_s}{\tau_1} = \frac{a}{\tau_1} \quad . \tag{5.12}$$

Therefore, one can find the constants a and b by observing at least two different workloads (i.e. different N_h) for the N=1 (single processor) case.

5.2.2 Parallelization on the CRAY YMP

The TPHOT code has been implemented on the Cray YMP with eight CPUs by researchers at Lawrence Livermore National Laboratory and yields correct results on our test problems. This code was macrotasked on the Cray YMP, taking advantage of its shared memory. The code yields identical results for a given number of tasks, regardless of the number of Cray CPUs which are actually executing the simulation, including a single processor which executes all tasks in sequence.

5.2.3 Parallelization on the BBN Butterfly

The message-passing version of TPHOT was implemented on Lawrence Livermore National Laboratory's 128 processor BBN Butterfly TC2000 using the Livermore Message-Passing (LMPS), a library of message-passing routines. Each processor of the BBN has 16 MBytes of memory that can be "shared" by all nodes via a "butterfly switch". Under LMPS, however, each node's memory belongs to only itself from the perspective of the application program. The code yielded identical results for the test problem run with 8 tasks on both the BBN and the Cray. Many different runs were made on the BBN, varying the number of processors from 1 to 116 and the number of particles (i.e., the workload W) from 2400 to 24,000,000.

Table 5.1 gives the simulation times for the Butterfly as a function of the number

Number of processors	Workload (W)									
	0.01		0.1		1.0		10.0		100.0	
	time (sec)	S_N	time (sec)	S_N	time (sec)	S_N	time (sec)	S_N	time (sec)	S_N
1	17	-	144	-	1407	-	-	-	-	-
4	6	2.83	38	3.79	357	3.94	-	-	-	-
8	5	3.40	22	6.55	181	7.77	1769	7.95	-	-
9	5	3.40	20	7.20	161	8.74	1595	8.82	-	-
10	5	3.40	18	8.00	145	9.70	1416	9.94	-	-
16	7	2.43	13	11.08	94	14.97	888	15.84	-	-
32	15	1.13	15	9.60	54	26.06	450	31.27	-	-
64	-	-	31	4.65	53	26.55	251	56.06	2364	59.52
80	-	-	-	-	-	-	223	63.09	1813	77.61
100	-	-	-	-	-	-	215	65.44	1493	94.24
116	-	-	-	-	-	-	224	62.81	1366	103.0

Table 5.1: Observed TPHOT Execution Times and Speedups for BBN.

of processors N and the workload W. We have arbitrarily assigned W=1.0 to the case with approximately 240,000 particles. Blanks appear in the table for two reasons: (1) large workloads are prohibitively expensive on few processors, and (2) small workloads on a large number of processors yield chaotic timings.

The speedups for each case in table 5.1 are computed using equation (5.1), using the N=1 case for each workload as the reference serial case (for τ_1). This is not quite correct, because this will not be the optimal serial code. This is probably not a large

effect, but it will tend to make the speedups appear better than they should be.

Workload (W)	# of histories (N_h)	model single processor execution time (τ_1)(sec)	observed single processor execution time (τ_1)(sec)	serial fraction (f)
0.01	2347	17.2	17	0.19
0.10	23843	143.8	144	0.023
1.00	238232	1407	1407	0.0024
10.0	2382320	14070	-	0.00024
100.0	23823200	140700	-	0.000024

Table 5.2: Parameters of BBN Linear Model.

5.2.4 Determination of Serial Fraction for BBN

Using the observed times in table 5.1 for the three cases W=0.01,0.10 and 1.0, a simple linear fit yielded the following values for the constants a and b in equation (5.11):

$$a = 3.34 \quad sec, \tag{5.13}$$

$$b = 0.00589 \quad sec/history. \tag{5.14}$$

To show the validity of the linear model in equation (5.11), table 5.2 compares the predicted values of τ_1 using equation (5.11) and the observed values from table 5.1. For the cases W=10.0 and 100.0, the serial execution times were too large to attempt, and were estimated using the linear model. Also tabulated are the predicted values of the serial fraction for each workload W. Speedup in table 5.1 for these cases use the prediction for τ_1 from table 5.2.

5.2.5 Parallelization Overhead for BBN

The theoretical speedup for the BBN as a function of N and W can be computed using the calculated values for the serial fraction, f, given in table 5.2. When this is done, the results compare quite poorly with the observed speedups tabulated in table 5.1. This indicates that the parallelization overhead cannot be ignored, and one needs to include a non-zero $\Delta\tau$ in the equation for speedup i.e., equation (5.5). Let us start with equation (5.5), and assume that $\Delta\tau$ can be modeled as follows:

$$\Delta\tau \approx \alpha \frac{N-1}{\nu} \quad , \tag{5.15}$$

where α is a constant. This model assumes that the increase in time due to parallelization is proportional to the increase in the number of processors and inversely proportional to the processor speed ν. Substituting equation (5.15) into equation (5.5) and using the results from equations (5.7) to (5.9), we find the following expression for speedup:

$$S_N = \frac{1}{\frac{(1-f)}{N} + f + \frac{\alpha(N-1)}{W}} \quad . \tag{5.16}$$

This expression for speedup is used to find values of α that would force the predicted speedups to match the observed speedups given in table 5.1. These "required" values of α show a trend in α somewhat independent of W but nearly linear with N, the number of processors. Therefore, α was modeled as follows:

$$\alpha \approx 0.0006 \frac{N}{116} \quad . \tag{5.17}$$

This expression is linear with N and the constant 0.0006 appears due to the fact that the required α was approximately 0.0006 for N=116, somewhat independent of W. Of course, this implies that the overhead term $\Delta\tau$ is not linear in N but quadratic:

$$\Delta\tau \approx \frac{0.0006}{.116\nu} N(N-1) \quad . \tag{5.18}$$

This $O(N^2)$ dependence may be due to memory contention resulting from the BBN implementation of the message passing system, LMPS, which relies on the shared memory features of the BBN.

This expression for α in equation (5.17) was then used in equation (5.16) to predict the speedups for the five different workloads and the number of processors varying from 1 to 116. The results are shown in figure 5.3. The plots include the observed speedups from table 5.1 for comparison. It is clear that this simple model, which uses two empirical parameters (f and α), predicts the behavior of the BBN very well over a wide range of numbers of processors and workloads.

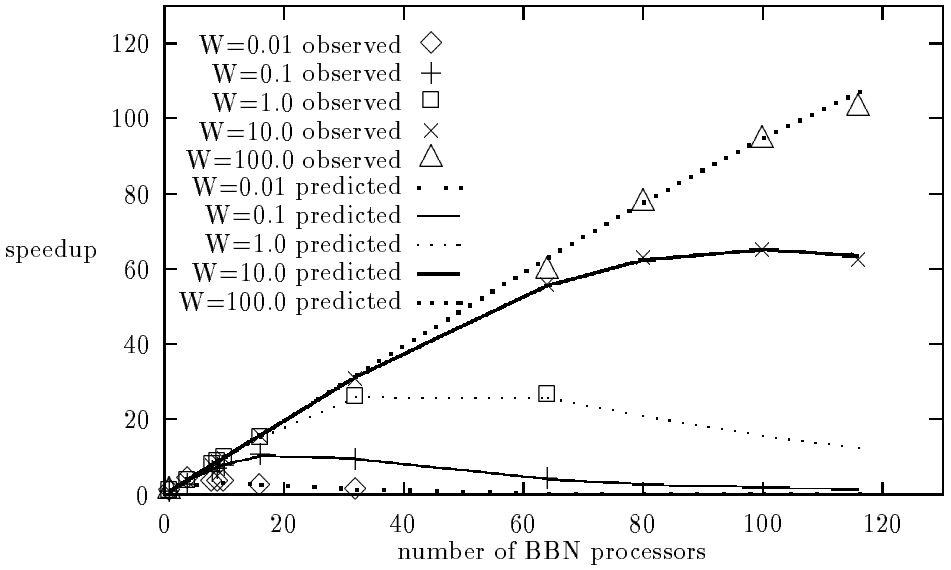


Figure 5.3: Observed and Predicted Speedups for BBN.

This model explains the behavior of the performance curves. As the workload, W, increases, linear speedups result even for the largest number of processors because the serial fraction, f, is inversely proportional to W, and equation (5.16) shows that the speedup becomes linear in N as f decreases and W increases. For small workloads,

f increases and S_N goes to zero, except for a small numbers of processors where the effect of the parallelization overhead vanishes, resulting again in a linear behavior of speedup S_N.

5.2.6 Parallelization on the Kendall Square KSR-1

The University of Michigan operated a Kendall Square Research KSR-1 parallel processor with 32 processors until its decommissioning in 1995. Although the memory is physically distributed, as with the BBN Butterfly, it is logically shared due to the unique architecture of the system. The message-passing version of the TPHOT code was implemented on the KSR-1, utilizing "pthreads" which are not like macrotasking on the Cray YMP. Table 5.3 tabulates the simulation times for our implementation of TPHOT on the KSR-1. Since one "pthread" corresponds to one processor for our timing runs, the table gives the run times in terms of "processors". In almost all cases, the Monte Carlo results are identical between the BBN and KSR-1 runs, with some exceptions that are thought to be due to the fact that decisions made on floating point arithmetic are sensitive to the order in which the sums are accumulated. In addition to the observed times, table 5.3 includes the corresponding speedups computed using equation (5.1) and using the single processor case for each workload as the reference serial run.

Using the same approach as with the BBN, the constants a and b needed in equation (5.11) for the KSR-1 were obtained using the single processor runs for W=0.01, 0.10 and 1.0:

$$a = 1.16 \quad sec, \tag{5.19}$$

$$b = 0.001 \quad sec/history. \tag{5.20}$$

These values were then used to predict the serial execution time τ_1 as a function

of workload W. These are given in table 5.4, along with the corresponding serial function, f. As with the BBN, the workload W=10.0 was not simulated for N=1; only the predicted values are shown. For the KSR-1, the value of α was found to be approximately 0.002, somewhat independent of workload W. The observed and predicted speedup plots for the KSR-1 are shown in figure 5.4. The speedup curves of the KSR-1 show similar characteristics as that of the BBN Butterfly parallel processor, i.e., as the workload increases, almost linear speedup is observed.

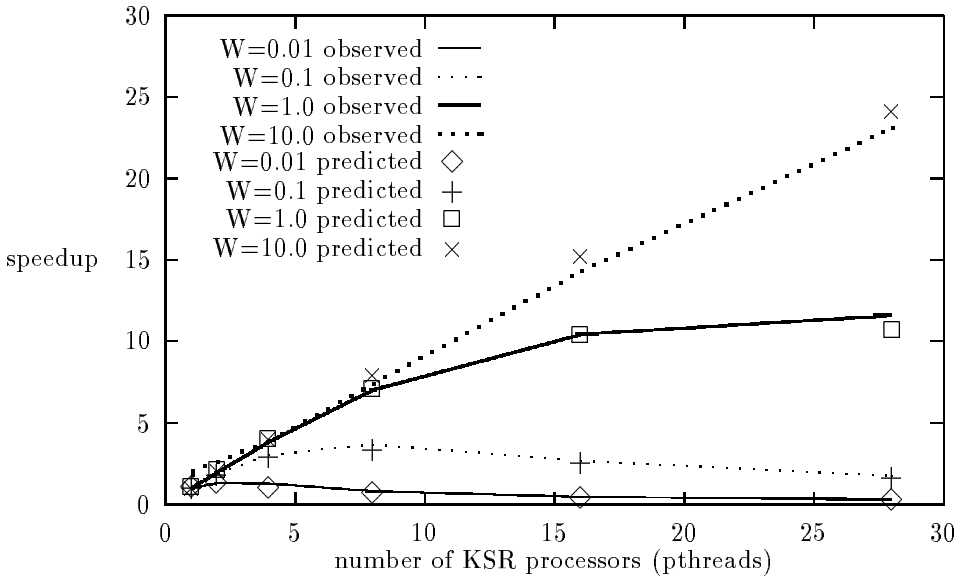


Figure 5.4: Observed and Predicted Speedups for KSR-1.

Number of processors	Workload (W)							
	0.01		0.1		1.0		10.0	
	time (sec)	S_N	time (sec)	S_N	time (sec)	S_N	time (sec)	S_N
1	3.52	1	25.9	1	248.2	1	-	-
2	2.7	1.28	14.0	1.85	129	1.93	-	-
4	2.9	1.25	8.6	3.02	65	3.82	640	3.9
8	4.6	0.77	7.1	3.66	35.5	6.99	335	7.4
16	8.6	0.42	9.7	2.67	24	10.4	173	14.3
28	13.9	0.26	15.2	1.71	21.4	11.6	107	23.1

Table 5.3: Observed TPHOT Execution Times and Speedups for KSR-1.

Workload (W)	# of histories (N_h)	model single processor execution time (τ_1)(sec)	observed single processor execution time (τ_1)(sec)	serial fraction (f)
0.01	2347	3.62	3.51	0.33
0.10	23843	25.8	25.9	0.045
1.00	238336	248.2	248.2	0.0047
10.0	2383360	2472	-	0.00047
100.0	23833600	24706	-	0.000047

Table 5.4: Parameters of KSR-1 Linear Model.

5.3 Parallel Monte Carlo Eigenvalue and Perturbation

In this section we present the issues involved in parallelizing Monte Carlo eigenvalue (using source iteration and fission matrix) and perturbations (using correlated sampling applied to the fission matrix approach) calculation algorithms [Maj95b]. Monte Carlo eigenvalue and perturbation algorithms differ from fixed source algorithms in the sense that both require an iteration procedure to determine the source distribution and the eigenvalue. At the end of each iteration, all the processors (master and slaves) need to be synchronized and the slave processors need to exchange information with the master processor. These synchronization and communication requirements increase parallelization overhead compared to fixed source Monte Carlo algorithms. The basic parallel algorithm used for eigenvalue and reactivity type simulation is shown in figure 5.5. We present in this section observed speedup performances from the IBM-SP2 parallel computer and fit predicted theoretical curves to these observed speedup results.

5.3.1 Parallel Algorithm

The eigenvalue and perturbation parallel algorithms are also based on a master-slave approach, as described in section 5.2. The master processor divides the total number of particles, for each fission generation, equally among all the available processors or slaves. Each slave processor simulates random walk for particle histories simultaneously with other slave processors. This simultaneous simulation of particle histories among all the slave processors is possible because particle histories are independent within each fission generation. Each slave processor also stores the sites and the number of next generation fission neutrons produced. This information is used by each slave processor for the next fission generation. At the end of each fission

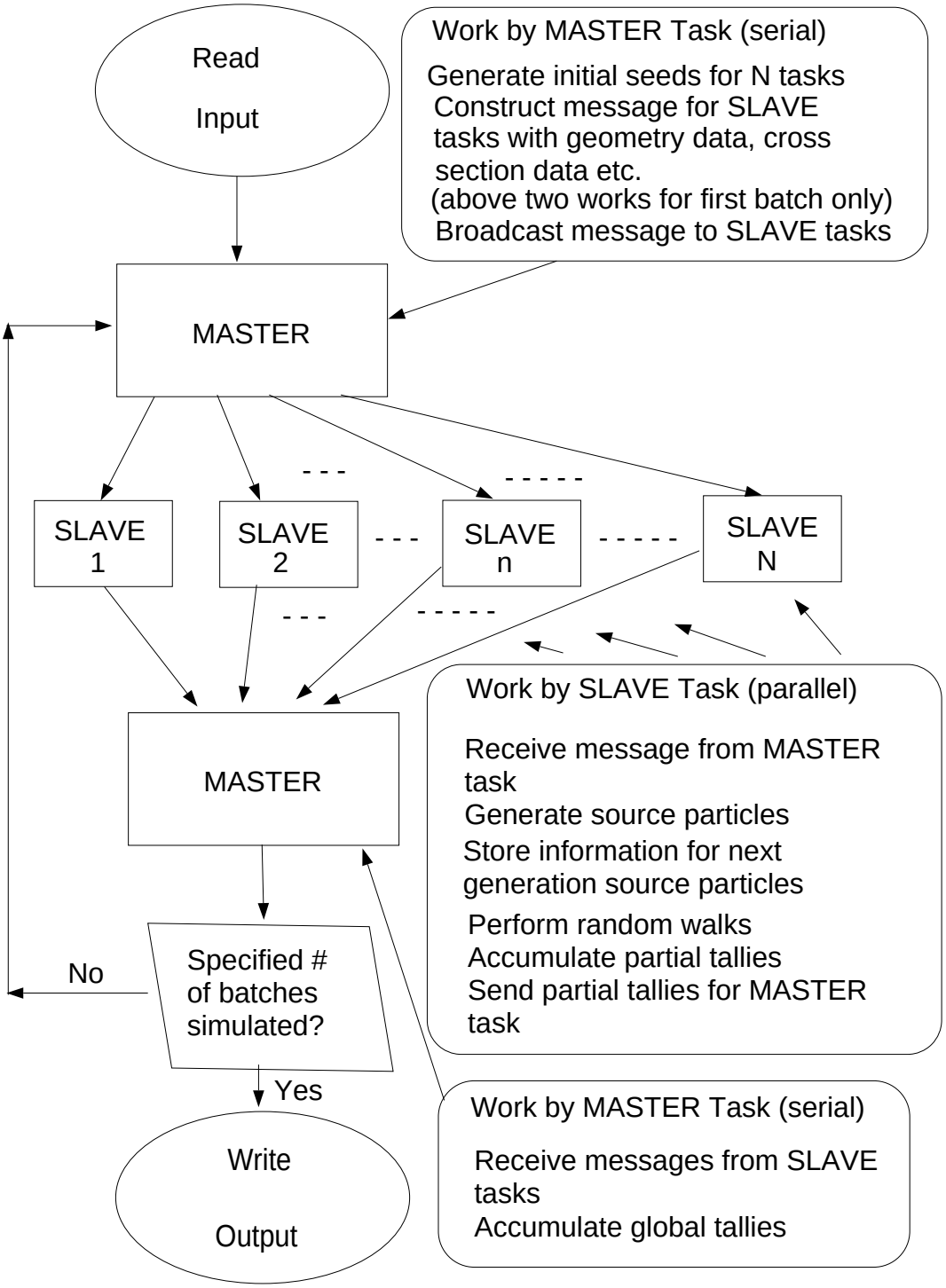


Figure 5.5: Flow Diagram for Eigenvalue and ΔK Parallel Monte Carlo Algorithms.

generation, tally results from each slave processor are collected in the master processor and the eigenvalue is computed. This iteration procedure, over fission gen-

erations, is repeated several times. This parallel algorithm also obeys the principle of reproducibility. Independent and reproducible sequences of random numbers are generated using the EGS4 [Nel85] random number generator. Each processor is given its own random number seed. These seeds are generated using the *skip ahead* approach, which allows each processor to generate a sequence of random numbers not overlapping with any other processor's random number sequence.

5.3.2 Theoretical Speedup Curves

Similar to section 5.2.1, an expression for speedup can be derived for the eigenvalue and perturbation parallel algorithms. For the master-slave algorithm, serial communication takes place, because the master process can only receive one message at a time. We expect the parallelization overhead and synchronization time to be proportional to (N-1). Hence, we express $\Delta\tau$ as;

$$\Delta\tau = \alpha(N-1) \quad . \tag{5.21}$$

The time for a parallel simulation on N processors can be written as;

$$\tau_N = a + \frac{bN_h}{N} + \alpha(N-1) \quad . \tag{5.22}$$

In the above equation, after evaluating constants a and b from two serial execution time, all the terms are known except α. The parameter τ_N is the actual measured time on N processors of IBM-SP2. Hence, α can be determined from equation (5.22). The observed speedup is expressed as;

$$S_N = \frac{\tau_1}{\tau_N} = \frac{\tau_1}{a + b\frac{N_h}{N} + \alpha(N-1)} \quad . \tag{5.23}$$

The predicted speedup follows the curve given by;

$$S_N(predicted) = \frac{N}{1+\beta N} \quad , \tag{5.24}$$

where,

$$\beta = \frac{\alpha}{\tau_N} \quad . \tag{5.25}$$

Here β represents the fraction of total computation time spent in parallelization overhead, communication, synchronization, etc. between two processors [Mat94].

5.3.3 IBM-SP2 Parallel Computer

The IBM-SP2 operated at the University of Michigan is a collection of 32 Power-2 chips processors grouped into a number of frames [IBM94]. Each Power-2 chip processor has 256 MBytes of memory. The group of processors has an additional high performance switch (HP-2). The high performance switch can be linked in either statically (when the parallel program is compiled) or dynamically (when the program is invoked). The IBM-SP2 has a latency of 40 micro seconds and bandwidth of 40 MBytes/sec. The IBM Message Passing Library (MPL) routines were used to run the parallel tasks.

5.3.4 Results

We have measured parallel speedups for three different algorithms. The first and second algorithms calculate eigenvalues using the source iteration technique and the fission matrix approach, respectively. The third algorithm calculates ΔK using the correlated sampling technique applied to the fission matrix approach of eigenvalue calculation (two ΔKs are calculated from a single Monte Carlo simulation). The amount of communication (measured within the master processor) between the master and the slave processors is largest for the perturbation algorithm and smallest for the source iteration eigenvalue algorithm. The number of communications required per iteration for the source iteration algorithm is three, for the fission matrix algorithm is four, and for the reactivity algorithm is eight. Table 5.5 shows the constants

a and b for the three algorithms. In tables 5.6, 5.7, 5.8 and 5.9 we show wall-clock timing results for different cases (i.e., number of particles/batch (p/b) and number of batches (b)) on the IBM-SP2 for the three different algorithms. Figures 5.6 and 5.7 show observed and predicted speedup results for the source iteration eigenvalue calculation. Figure 5.8 shows the percentage of total computation time spent in communication (message passing) in the master processor for the source iteration algorithm. Figures 5.9, 5.10, and 5.11 show similar plots for fission matrix eigenvalue calculations and figures 5.12, 5.13 and 5.14 show plots for correlated sampling perturbation calculations. We find that our predicted speedup results match the observed results reasonably well. The observed results were taken on the dedicated IBM-SP2 parallel computer. From the speedup plots, we see that we obtain speedups close to 9 for 10 processors for all the three algorithms. Among these three algorithms, the perturbation algorithm shows largest fraction of total computation time spent in communication.

Algorithm type	a (sec)	b (sec/history)
Source Iteration Eigenvalue	0.5194	4.5692E-4
Fission Matrix Eigenvalue	0.02	4.434E-4
Correlated Sampling Reactivity	0.7431	6.5815E-4

Table 5.5: Constants a and b of single processor execution time.

# of processors	Wall-clock timings in seconds		
	Source Iteration	Fission Matrix	Correlated Sampling
1	365.84	357.52	527.59
2	212.23	183.96	279.25
4	105.24	100.11	158.92
8	58.72	59.51	97.61
10	50.73	56.39	81.55

Table 5.6: Wall-clock Timing Results on IBM-SP2 for 8000 particle/batch, 100 batch case.

# of processors	Wall-clock timings in seconds		
	Source Iteration	Fission Matrix	Correlated Sampling
1	731.73	710.94	1053.57
2	377.43	369.10	590.04
4	200.64	190.38	280.29
8	107.58	107.40	166.28
10	98.18	89.30	151.37

Table 5.7: Wall-clock Timing Results on IBM-SP2 for 16000 particle/batch, 100 batch case.

# of processors	Wall-clock timings in seconds		
	Source Iteration	Fission Matrix	Correlated Sampling
1	364.36	354.04	526.03
2	187.65	195.72	277.08
4	98.82	98.59	141.53
8	53.78	50.44	78.39
10	45.47	44.10	70.20

Table 5.8: Wall-clock Timing Results on IBM-SP2 for 16000 particle/batch, 50 batch case.

# of processors	Wall-clock timings in seconds		
	Source Iteration	Fission Matrix	Correlated Sampling
1	729.08	709.55	1050.06
2	372.50	361.28	550.08
4	192.92	191.80	284.25
8	104.75	97.13	158.12
10	84.29	82.02	123.57

Table 5.9: Wall-clock Timing Results on IBM-SP2 for 32000 particle/batch, 50 batch case.

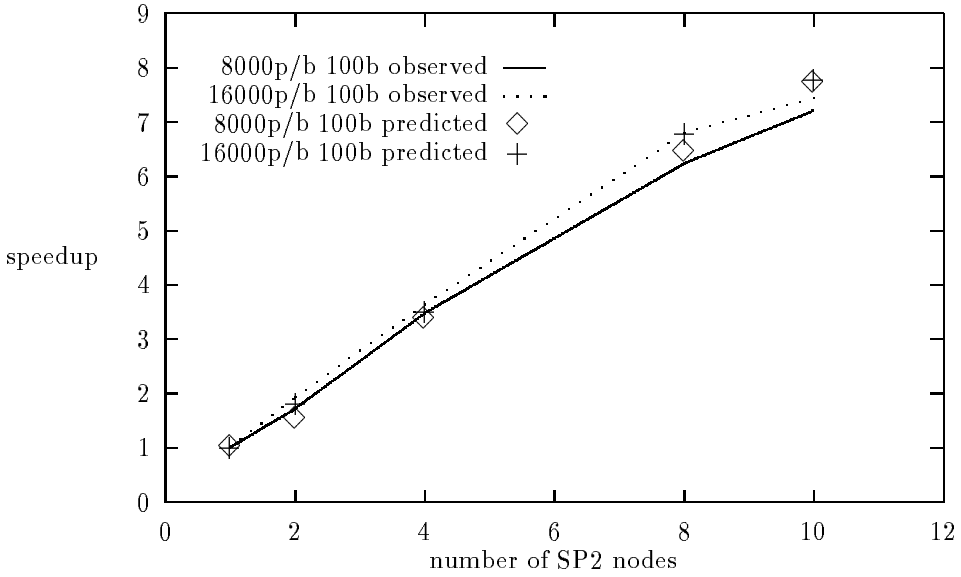


Figure 5.6: Source Iteration Speedup Plots.

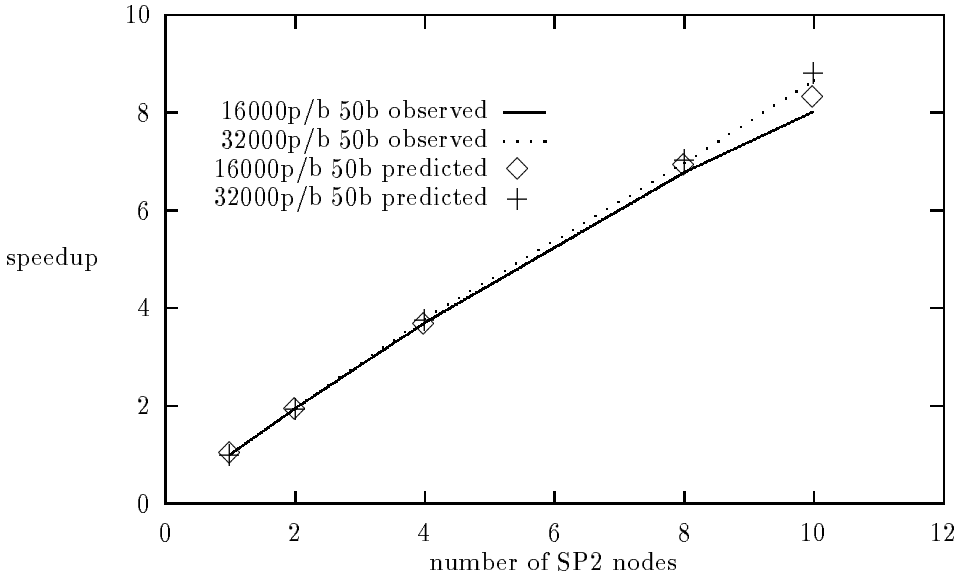


Figure 5.7: Source Iteration Speedup Plots.

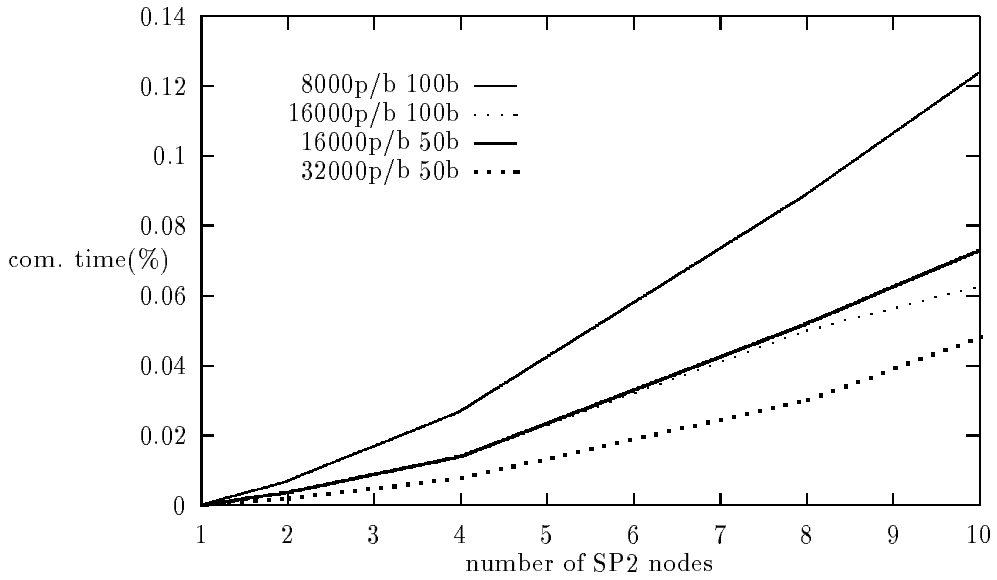


Figure 5.8: % of Total Time Spent in Communication in a Processor for Source Iteration.

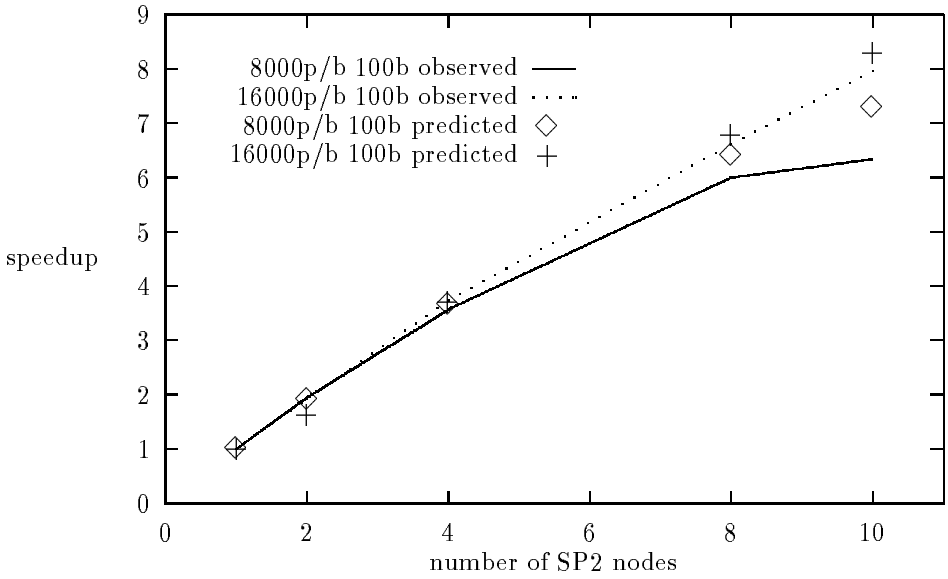


Figure 5.9: Fission Matrix Speedup Plots.

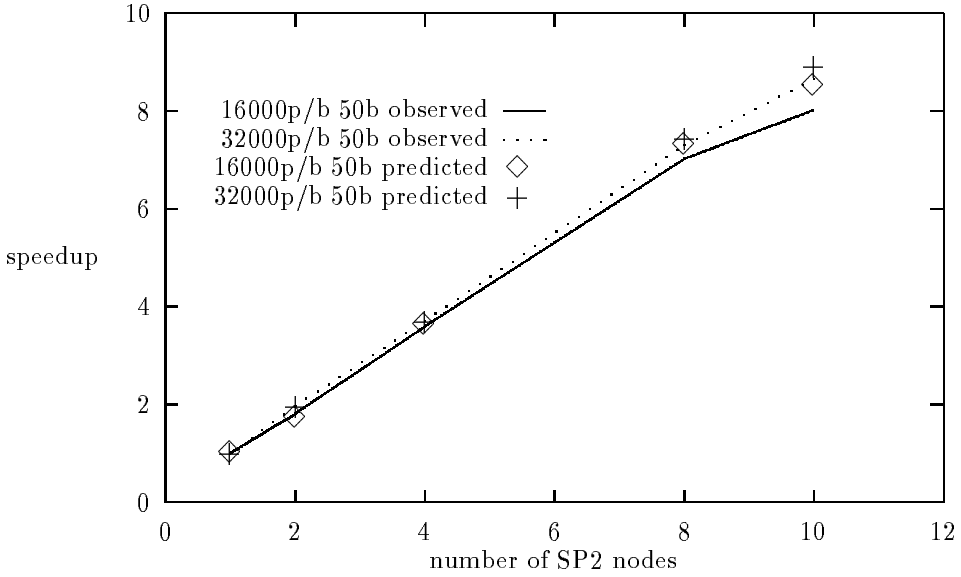


Figure 5.10: Fission Matrix Speedup Plots.

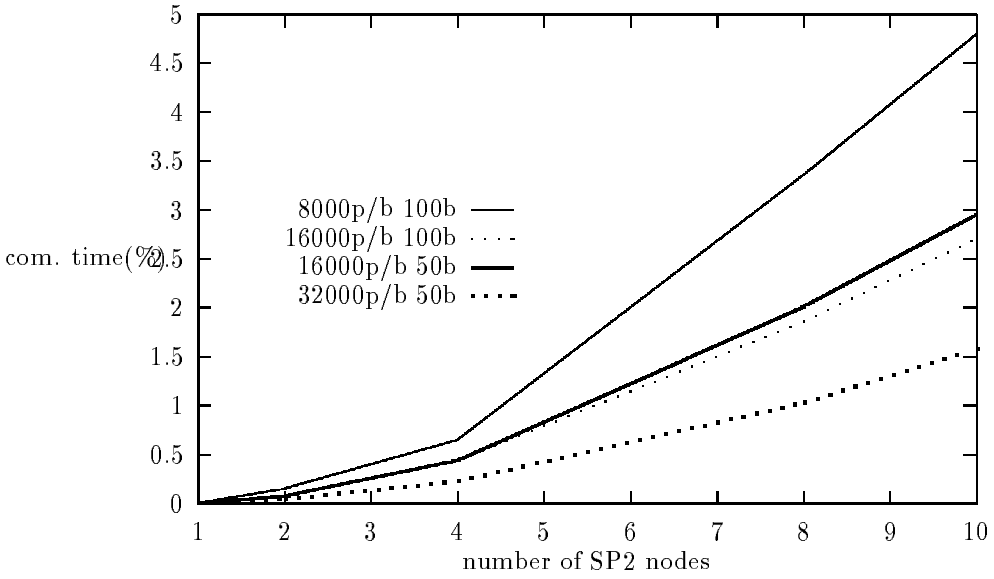


Figure 5.11: % of Total Time Spent in Communication in a Processor for Fission Matrix.

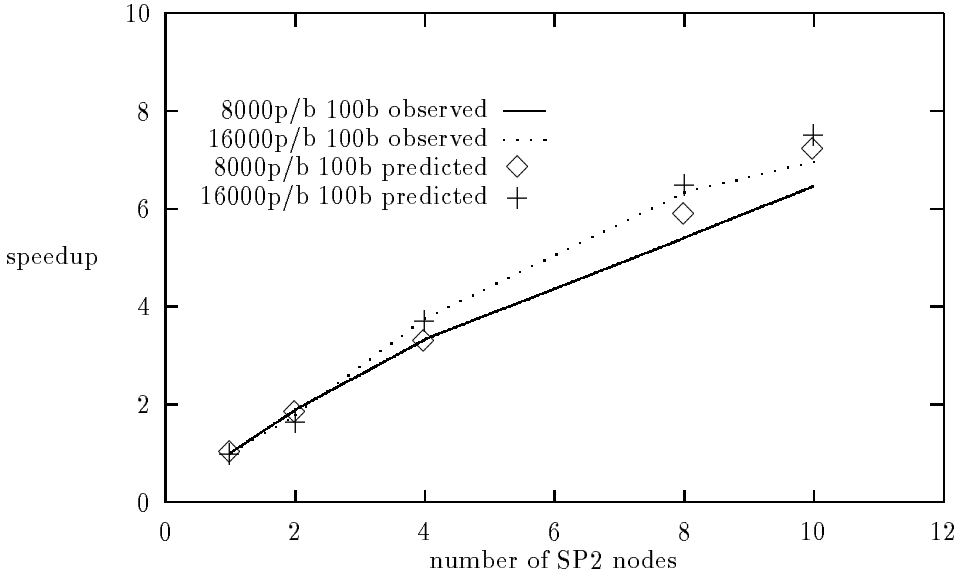


Figure 5.12: Correlated Sampling Speedup Plots.

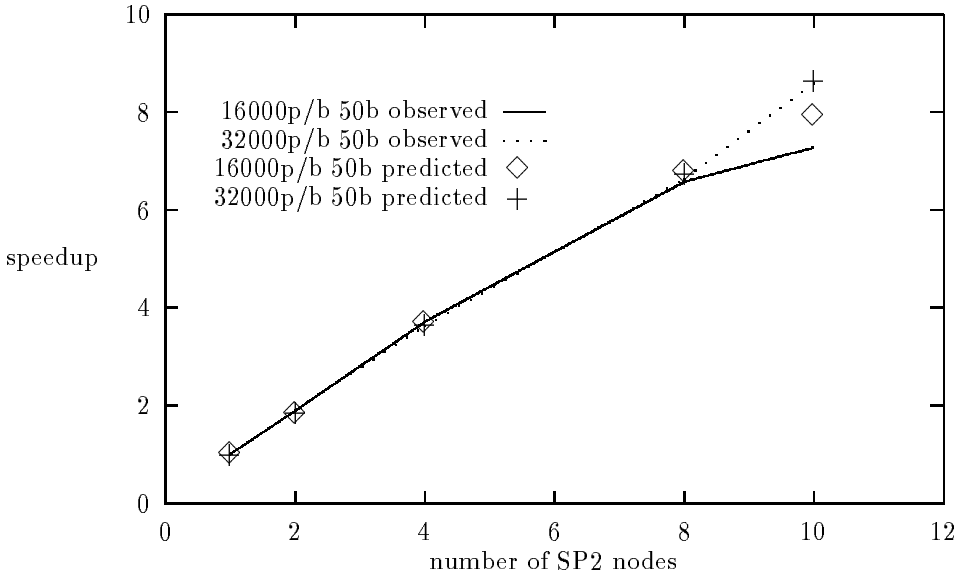


Figure 5.13: Correlated Sampling Speedup Plots.

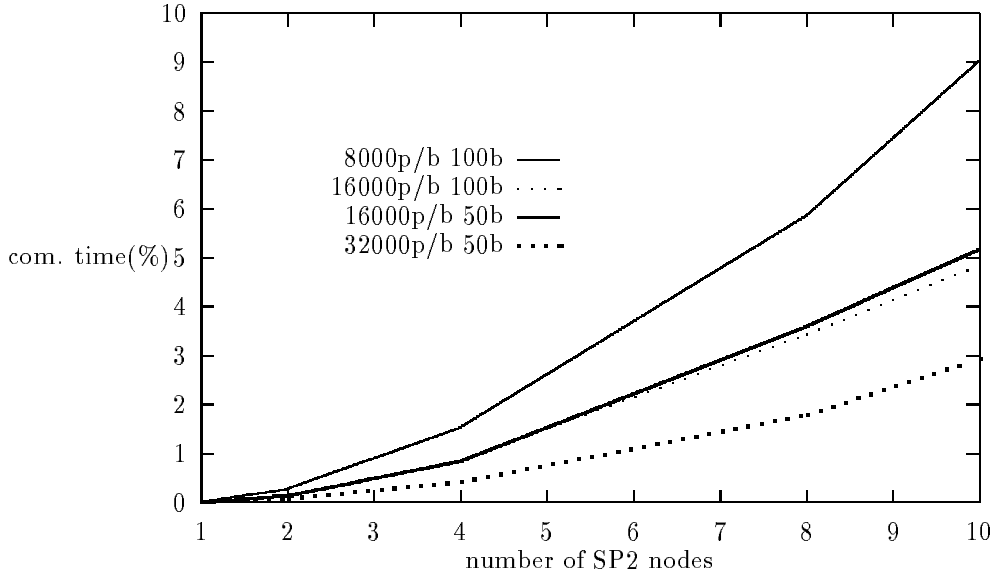


Figure 5.14: % of Total Time Spent in Communication in a Processor for Correlated Sampling.

5.3.5 $\Delta\tau$ for the IBM-SP2

The values of $\Delta\tau$s for the IBM-SP2 computer was calculated from equation (5.22) where τ_N is the actual wall clock timing on N processors. In tables 5.10, 5.11, 5.12 and 5.13 we show the $\Delta\tau$s corresponding to results of tables 5.6, 5.7, 5.8 and 5.9 respectively.

# of processors	Source Iteration	Fission Matrix	Correlated Sampling
2	29.2	6.58	15.0
4	10.0	11.4	26.4
8	12.5	15.1	30.8
10	13.7	20.9	27.9

Table 5.10: IBM-SP2 $\Delta\tau$s for 8000p/b, 100 batch case.

# of processors	Source Iteration	Fission Matrix	Correlated Sampling
2	11.4	14.4	62.8
4	17.3	13.0	16.3
8	15.7	18.7	33.9
10	24.6	18.3	45.0

Table 5.11: IBM-SP2 $\Delta\tau$s for 16000p/b, 100 batch case.

# of processors	Source Iteration	Fission Matrix	Correlated Sampling
2	4.4	18.3	13.1
4	6.9	9.9	9.3
8	7.7	6.1	11.9
10	8.4	8.6	17.1

Table 5.12: IBM-SP2 $\Delta\tau$s for 16000p/b, 50 batch case.

# of processors	Source Iteration	Fission Matrix	Correlated Sampling
2	6.4	6.5	22.8
4	9.6	14.4	20.3
8	12.9	8.4	25.7
10	10.6	11.0	17.5

Table 5.13: IBM-SP2 $\Delta\tau$s for 32000p/b, 50 batch case.

Even though it is difficult to predict an empirical formula that exactly determines $\Delta\tau$s, we notice that for most of the cases $\Delta\tau$ is about a constant. If $\Delta\tau$ is assumed to be a constant then α is proportional to $\frac{1}{N}$. Hence speedup models for IBM-SP2 can be predicted reasonably well with two parameters, the serial time constant a and the proportionality constant for $\frac{1}{N}$.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this chapter we discuss the original aspects of this research work. Different theories and issues that lead to the development of the numerical method for multiple eigenvalue perturbation calculations are summarized. While developing the multiple perturbation technique, we investigated two fission matrix algorithms for eigenvalue problems. The significance of parallel computers for Monte Carlo particle transport algorithms is discussed. Finally, we provide some suggestions for further research topics. Some of these research topics would give more insights into the method developed here, while others would extend and apply the ideas developed in this work to different related problems of interest.

6.1 Conclusions

The main objective of this research work was to develop a computational method that would calculate multiple perturbation effects in the eigenvalue of the Boltzmann transport equation for neutrons from a single Monte Carlo simulation. Even though Monte Carlo methods can efficiently estimate the eigenvalue of the transport equation, calculation of small perturbation effects encounters difficulties. It has been shown in this research work by numerical examples, and by other researchers, that

subtracting two independent Monte Carlo simulations for estimating perturbation effects in eigenvalue is not efficient and sometimes provides incorrect estimates. This is specifically true for cases in which the difference in eigenvalues is of the same order as the uncertainty of the eigenvalues. Hence, to estimate eigenvalue perturbation effects using Monte Carlo techniques, the unperturbed and perturbed simulations need to be positively correlated. Special Monte Carlo perturbation techniques such as the correlated sampling, derivative operator sampling, or the importance function approach must be employed to obtain an efficient estimate of eigenvalue perturbations. Our work has dealt with the correlated sampling technique.

In chapter II we discussed two Monte Carlo approaches to estimate the eigenvalue of a system, the source iteration method and the fission matrix approach. In the source iteration technique, the eigenvalue of a system is calculated as the ratio of source neutrons of two successive neutron generations. The fission matrix algorithm solves the homogeneous neutron transport equation which holds for every generation. This equation is discretized to generate a matrix equation. Monte Carlo particle tracking is done to estimate contributions of neutrons to each of the matrix elements of the fission matrix, and then the largest eigenvalue of the fission matrix is determined numerically. We have investigated two variations of the fission matrix algorithm. In the cycle fission matrix algorithm the fission matrix is generated from contributions of all the neutrons of a particular batch. In contrast, in the cumulative fission matrix algorithm the fission matrix is formed from contributions of neutrons from all the batches up to and including the last batch. We have tested these two algorithms for problems representing tightly coupled and loosely coupled systems. We have observed from our computational experiments that for tightly coupled systems, both the cycle fission matrix and the cumulative fission matrix algorithm perform

well and yield comparable performance. But for loosely coupled systems, the cumulative fission matrix algorithm performs better than the cycle fission matrix algorithm. For loosely coupled systems there is less neutron communication between the different parts of the system, and hence a fission matrix formed with the neutrons of a particular batch provides poor sampling. One characteristic of the cumulative fission matrix algorithm is that it does not provide any uncertainty of the results, since the eigenvalues of different batches are not statistically independent.

The correlated sampling technique was applied to both the source iteration and the cycle fission matrix approach. It was shown, by numerical experiments, that the correlated sampling technique, when applied to the source iteration method, fails to efficiently calculate perturbation effects in the eigenvalue of the transport equation. It is not possible to calculate perturbations in eigenvalues by propagating perturbed weights from one generation to another. Any useful information about the eigenvalue perturbation is lost due to fluctuations in the perturbed weights from one generation to another. However the calculation of the fission matrix constitutes an initial value problem and correlated sampling or derivative operator sampling can be applied directly. The correlated sampling technique, when applied to the fission matrix method, can accurately and efficiently estimate small perturbation effects in the eigenvalue. Next, we introduced the idea of performing the actual Monte Carlo simulation in an artificial reference system different from both the unperturbed and perturbed systems. The choice of a proper reference system is an important issue. The important characteristic of the reference system is that a transition that may occur in any of the unperturbed or perturbed systems could occur in the reference system. Even after satisfying this characteristic there could be many choices for the reference system. In appendix A we have performed computational experiments to

address this issue and have shown that varying a reference system does impact the outcome of the simulation and hence this issue should be addressed from theoretical standpoint. For all our simulations, we have taken the reference system to be an average of the unperturbed and all perturbed systems. We have also introduced the concept of adding a forward δ-scatter cross section to the total cross section of all the systems to reduce fluctuations in the adjusting weight factors. All these theories and numerical examples to support them are given in chapter III.

Next, in chapter IV, the ideas of correlated sampling technique, fission matrix approach, and an artificial reference system were combined to develop the multiple perturbation CSFM Monte Carlo technique. In this technique the Monte Carlo simulation is done in an artificial reference system and multiple fission matrices for the multiple perturbed systems and for the unperturbed system are formed by correlating them to the reference system's fission matrix. The correlated sampling technique allows one to form multiple fission matrices from a single Monte Carlo simulation. At the end of the simulation, the dominant eigenvalue of each of the multiple fission matrices of the perturbed systems are evaluated numerically, along with the dominant eigenvalue of the unperturbed fission matrix. We have tested this Monte Carlo technique for various test problems and compared the results to that of the TWODANT code. Satisfactory comparison of results between the multiple perturbation Monte Carlo method and the TWODANT code validates the multiple perturbation method. This method allows significant savings in computational efforts as discussed in chapter IV. This method can be applied to problems in which it is desired to calculate multiple perturbations in the eigenvalue due to small variations in cross sections. Some practical examples of such problems are perturbations in eigenvalue due to changes in soluble boron concentrations, different number of absorber rods in assem-

blies, and different assembly loading patterns for global core calculations. We have also shown that the CSFM method can be used as a tool for design applications when the eigenvalue of the unperturbed system is known beforehand to a good accuracy by a separate calculation. Even though the CSFM method is meant for small perturbation problems, it has performed well for some large perturbation cases.

We have also implemented parallel Monte Carlo algorithms for particle transport on the KSR-1, BBN Butterfly and IBM-SP2 parallel computers. Both fixed source and eigenvalue type neutron transport algorithms were implemented on these machines. For many applications, Monte Carlo algorithms may be slow compared to deterministic methods, whereas Monte Carlo algorithms are inherently parallel and hence easy to parallelize compared to deterministic methods. We have observed close to linear speedups for some of the example Monte Carlo problems. These speedups are observed for both fixed source and eigenvalue type Monte Carlo algorithms. Theoretical models of speedups for parallel particle transport algorithms were developed and compare well with observed speedup results for all three parallel machines. As the price of MPPs decreases, the inherent parallelism of Monte Carlo will make it a significant computational tool of choice.

6.2 Future Work

Finally, we would like to suggest some future research directions based on this work that would either provide more insight into the multiple perturbation approach or make it applicable to other problems of interest. In appendix A we perform a computational study on the choice of the reference system. Results of this study suggest that an optimized choice of a reference system for a given problem is desirable. An optimized reference system could be determined by requiring minimum variance

in the result. This requires further theoretical study and we will leave this as a future research topic.

The multiple perturbation method developed here is for an isotropic scattering medium. A possible next step would be to extend this approach to anisotropic scattering. Another possibility is to extend this approach to continuous energy Monte Carlo.

Calculation of multiple perturbations in eigenvalue involves solution of fixed source Monte Carlo problems over a number of fission generations. We were able to calculate perturbations in the eigenvalue because this method was able to successfully solve fixed source problem for each generation. This suggests that this method could be applied for solving perturbation problems in reaction rates. This would lead to solution of perturbation problems in shielding calculations where different shielding materials are tested.

In this research work we applied the fission matrix approach of eigenvalue calculation to the correlated sampling perturbation technique. We have mentioned that another Monte Carlo perturbation technique is the derivative operator sampling approach. Another extension to this work might be to combine the fission matrix and derivative operator sampling methods to calculate multiple eigenvalue perturbations.

APPENDICES

APPENDIX A

CHOICE OF A REFERENCE SYSTEM

The multiple perturbation method developed in this dissertation requires that the Monte Carlo simulations be performed in a reference system different from the unperturbed and all perturbed systems. We have chosen the reference system cross sections to be the arithmetic average of the cross sections of the unperturbed and all perturbed systems. It may be possible to choose another reference system that is more appropriate for a given problem. An optimized reference system for a given problem would be the one that produces the most accurate ΔK along with the minimum variance. Our choice of reference system is based on intuition rather than any theoretical study. We believe that it might be possible to study the choice of a reference system as an optimization problem. We suggested this optimization problem as a future research topic in chapter VI. In this appendix we perform numerical experiments to investigate the effect of the choice of a reference system on the result of ΔK. We will perform these numerical experiments on a few problems that have been used as example problems in this dissertation.

For the numerical experiment, we choose a problem in which there is only one perturbed system corresponding to an unperturbed system. It is understood that similar observations can be made for problems with multiple perturbed systems. The

reference system's cross sections will be varied according to the following formula,

$$\Sigma_x^{ref} = \alpha\Sigma_x^{up} + (1-\alpha)\Sigma_x^{p} \quad , \quad 0 \leq \alpha \leq 1, \tag{A.1}$$

where Σ_x refers to some cross section. We would look at the ΔK results as a function of α.

The first example is for a one energy group homogeneous slab problem with a thickness of 16 cm and vacuum boundary conditions on both ends. Cross section perturbation was done over the entire slab. Unperturbed cross sections of the slab are as follows,

$$\Sigma_t^{up} = 1.0, \Sigma_s^{up} = 0.9, \nu\Sigma_f^{up} = 0.11, \Sigma_a^{up} = 0.1.$$

All the cross sections have units of cm^{-1}. Perturbation was done in the fission cross section of the slab, and the perturbed cross sections are as follows,

$$\Sigma_t^{p} = 1.0, \Sigma_s^{p} = 0.9, \nu\Sigma_f^{p} = 0.111, \Sigma_a^{p} = 0.1.$$

Table A.1 shows the ΔK results as a function of α. The Monte Carlo results were generated from 30 inactive batches, 70 active batches, and 2000 neutrons per batch.

It appears from table A.1 that for this problem, when the reference system is chosen as the unperturbed system, the most accurate result for ΔK is produced. It should be noted that the errors are not linearly dependent on α. This analysis assumes the TWODANT result as exact.

Next, we perturb the absorption cross section for the same homogeneous one energy group slab problem as in table A.1. The perturbed cross sections are as follows,

$$\Sigma_t^{p} = 1.001 \quad , \Sigma_s^{p} = 0.9 \quad , \nu\Sigma_f^{p} = 0.11 \quad , \Sigma_a^{p} = 0.101.$$

α	TWODANT ΔK	Monte Carlo ΔK	% error
1.0	0.00903	0.0090308$\pm$.58E-5 $\delta^{ref} = 0.0$	0.009
0.75	0.00903	0.0090344$\pm$.63E-5 $\delta^{ref} = 0.0$	0.05
0.50	0.00903	0.0090496$\pm$.56E-5 $\delta^{ref} = 0.0$	0.22
0.25	0.00903	0.009021$\pm$.53E-5 $\delta^{ref} = 0.0$	0.1
0.0	0.00903	0.0090405$\pm$.54E-5 $\delta^{ref} = 0.0$	0.12

Table A.1: Perturbation(in $\nu\Sigma_f$) Results as a Function of Reference System.

Perturbation results as a function of α are shown in table A.2. The Monte Carlo results were generated using 30 inactive batches, 70 active batches and 2000 neutrons per batch.

From table A.2 we observe that the most accurate result is achieved when α equals 0.25. We observe for this case also that the errors are not a linear function of α. Again these conclusions are made based on the fact that the TWODANT result is the correct one.

The above two numerical experiments suggest that even though the results are a function of the chosen reference system, it is difficult to determine the relation. For complicated systems (heterogeneous, multigroup, etc.), it will be more difficult to

α	TWODANT ΔK	Monte Carlo ΔK	% error
1.0	-0.008807	-0.0088914$\mp$.85E-5 δ^{ref} =0.001	0.96
0.75	-0.008807	-0.0088895$\mp$.88E-5 δ^{ref} = 0.00075	0.94
0.50	-0.008807	-0.0089069$\mp$.83E-5 δ^{ref} = 0.0005	1.13
0.25	-0.008807	-0.008879$\mp$.84E-5 δ^{ref} = 0.00025	0.82
0.0	-0.008807	-0.0088832$\mp$.94E-5 δ^{ref} = 0.0	0.87

Table A.2: Perturbation(in Σ_a) Results as a Function of Reference System.

determine this relationship. This is an optimization problem, and a theoretical investigation is required to gain more insight into it.

APPENDIX B

TWODANT ACCURACY

In this appendix we perform accuracy test for the TWODANT code. For a given problem we vary the mesh size as well as the quadrature sets to observe the effect of these variations on TWODANT calculated eigenvalue results. These results guide us to determine up to how many digits, after the decimal point, the TWODANT calculated ΔKs are accurate.

The test problem is a homogeneous, one energy group, slab, with thickness of 16 cm and vacuum boundary conditions on both ends. Tables B.1 and B.2 show the different eigenvalue results due to different mesh sizes and quadrature sets.

In most of our TWODANT calculations, for chapter 2, 3 and 4 results, we have used mesh sizes and quadrature sets which are equivalent to that of the first row of table B.1. This implies that the TWODANT ΔKs are accurate only up to five digits after the decimal point. We observe that even though the inner and outer iteration convergence criteria are set to 10^{-12} the TWODANT eigenvalues are not accurate up to that many digits after the decimal point.

Mesh Size (mfp)	quadrature sets	convergence criteria	eigenvalue
0.125	32	10^{-12}	0.993070
0.0625	32	10^{-12}	0.993075
0.03125	32	10^{-12}	0.993077
0.015625	32	10^{-12}	0.993077

Table B.1: Eigenvalue Results for Different Mesh Sizes.

Mesh Size (mfp)	quadrature sets	convergence criterion	eigenvalue
0.125	16	10^{-12}	0.993056
0.0625	16	10^{-12}	0.993061
0.03125	16	10^{-12}	0.993063
0.015625	16	10^{-12}	0.993063

Table B.2: Eigenvalue Results for Different Quadrature Sets.

BIBLIOGRAPHY

BIBLIOGRAPHY

[Ave58] R. Avery, "Theory of Coupled Reactors," *Proc. United Nations 2nd Int. Conf. Peaceful Uses of Atomic Energy*, Geneva, Switzerland, **12**, 182 (1958).

[Bel70] G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Rienhold, New York (1970).

[Blo83] R. N. Blomquist, U. Feldman and E. M. Gelbard, "Monte Carlo Small-Sample Perturbation Calculations," *Proc. Topl. Mtg. Advances in Reactor Computations*, March 28 - 31, 1983, Salt Lake City, Utah, Vol. 1, p. 124 (1983).

[Bob84] F. W. Bobrowicz, J. E. Lynch, K. J. Fisher and J. E. Tabor, "Vectorized Monte Carlo Photon Transport," *Parallel Computing*, **1**, 295 (1984).

[Bow72] A. H. Bowker and G. J. Liberman, *Engineering Statistics*, Second Edition, Prentice-Hall Inc., New Jersey (1972).

[Bow83] H. Bowsher, E. M. Gelbard and P. Gemmel and G. Pack, "Magnitude of Bias in Monte Carlo Eigenvalue Calculations," *Trans. Am. Nucl. Soc.*, **45**, 324 (1983).

[Bra70] R. W. Brandon and J. C. Robinson, "A Variational Approach for the Determination of Neutron Flux Spectra from Detector Activation," *Nucl. Sci. Eng.*, **39**, 151 (1970).

[Bri88] J. F. Briesmeister, Ed., "MCNP-A General Monte Carlo Code N-Particle Transport Code," LA-12625-M, Los Alamos National Laboratory (1988).

[Bri86] R. J. Brissenden and A. R. Garlick, "Biases in the Estimations of K^{eff} and Its Errors by Monte Carlo Methods," *Ann. Nucl. Energy*, **13**, 2, 63 (1986).

[Bro81] F. B. Brown, W. R. Martin and D. A. Calahan, "A Discrete Sampling Method for Vectorized Monte Carlo Calculations," *Trans. Am. Nucl. Soc.*, **38**, 354 (1981).

[Bro85] F. B. Brown and W. R. Martin, "Monte Carlo Methods for Radiation Transport Analysis on Vector Computers," *Prog. Nucl. Energy.*, **14**, 269 (1985).

[Bro86] F. B. Brown, "Vectorization of Three-Dimensional General-Geometry Monte Carlo," *Trans. Am. Nucl. Soc.*, **53**, 283 (1986).

[Car75] L. L. Carter and E. D. Cashwell, *Particle Transport Simulation with the Monte Carlo Method*, ERDA Critical Review Series, U.S. Energy Research and Development Administration, Technical Information Center-26607, Oak Ridge, TN (1975).

[Cas67] K. M. Case and P. F. Zweifel, *Linear Transport Theory*, Addison - Wesley, Reading, Massachusetts (1967).

[Cha85] Y. Chauvet, "Vectorization and Multitasking with a Monte Carlo Code for Neutron Transport Problems," *Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics, Proc.*, Cadarache Castle, France, April 1985, R. Alcouffe, R. Dautray, A. Forster, G. Ledanois and B. Mercier, Eds., p. 234, Springer-Verlag, New York (1985).

[O'De82] R. D. O'Dell, F. W. Brinkley and D. Marr, "User's Manual for TWODANT: A code Package for Two-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," LA-9814-M, Los Alamos National Lab., (1982).

[Dud79] J. J. Duderstadt and W. R. Martin, *Transport Theory*, John Wiley & Sons, New York (1979).

[Eno90] M. Enosh, D. Shalaitin and Y. Yeiven, "The Bias in Monte Carlo Eigenvalue Calculations," *Prog. Nucl. Energy*, **24**, 1-3, 259 (1990).

[Gal95] F. Z. Gallmeier, "A New MCNP Option: KCORR - The Use of the Correlated Sampling Method to Study Reactivity Effects Due to Changes of a Reactor Arrangement," *Nucl. Sci. Eng.*, **120**, 102 (1995).

[Gas75] R. C. Gast and N. R. Candelore, "Monte Carlo Eigenfunctions Strategies and Uncertainties," *Proc. NEACRP Mtg. Monte Carlo Study Group*, ANL-75-2, NEA-CRP-L-118, Argonne National Lab. (1975).

[Gel74] E. M. Gelbard and R. E. Prael, "Monte Carlo Work at Argonne National Laboratory," ANL-75-2(NEACRP-L-118), Argonne National Laboratory, 202 (1974).

[Gel81] E. M. Gelbard, "Unfinished Monte Carlo Business," *Proc. ANS/ENS Int. Meeting on Adv. in Mathematical Methods of Nuclear Engineering Problems*, Munich, Germany, April 27-29, 1981, European Nuclear Society (1981).

[Gel90a] E. M. Gelbard and R. E. Prael, "Computations of Standard Deviations in Eigenvalue Calculations," *Prog. Nucl. Energy*, **24**, 237 (1990).

[Gel90b] E. M. Gelbard, "Present Status of Future Prospects of Neutronics Monte Carlo," *Int. Conf. on monte Carlo Methods for Neutron and Photon Transport Calculations*, Budapest, Hungary, Sept. 25-28, (1990).

[Gel91] E. M. Gelbard, "Monte Carlo Eigenvalue Biases: Generalization Beyond the Absorption Estimate," *Trans. Am. Nucl. Soc.*, **64**, 302 (1991).

[Gel94] E. M. Gelbard and A. G. Gu, "Biases in Monte Carlo Eigenvalue Calculations," *Nucl. Sci. Eng.*, **117**, 1 (1994).

[Goa59] W. Goad and R. Johnston, "A Monte Carlo Method for Criticality Problems," *Nucl. Sci. Eng.*, **5**, 371 (1959).

[Hal80] M. C. G. Hall, "Monte Carlo Perturbation Theory in Neutron Transport Calculations," *Proc. Seminar Workshop*, ORNL/RSIC-44 Report, Oak Ridge National Laboratory, 47 (1980).

[Hal82] M. C. G. Hall, "Cross-section adjustment with Monte Carlo Sensitivities. Application to the Winfrith Iron Benchmark," *Nucl. Sci. Eng.*, **81**, 423 (1982).

[Ham64] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, John Wiley & Sons, New York (1964).

[Hof72a] T. J. Hoffman and L. M. Petrie, "Monte Carlo Reactivity Calculations Using a Perturbation Source," *Trans. Am. Nucl. Soc.*, **15**, 912 (1972).

[Hof72b] T. J. Hoffman, J. C. Robinson and P. N. Stevens, "The Adjoint Difference Method and Its Application to Deep-Penetration Radiation Transport," *Nucl. Sci. Eng.*, **48**, 179 (1972).

[Hof78] T. J. Hoffman, L. M. Petrie and N. F. Landers, "A Monte Carlo Perturbation Source Method for Reactivity Calculations," *Nucl. Sci. Eng.*, **66**, 60 (1978).

[IBM94] IBM AIX Parallel Environment Operation and Use, Number SH26-7228, (1994).

[Jon87] D.B. Jones, "ARMP-02 Documentation, Part II, Chapter 6 – CPM-2 Computer Code Manual," EPRI NP-4574-CCM, Part II, Ch. 6, Volumes 1,2, and 3, Electric Power Research Institute (1987).

[Kal86] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods, Vol.I: Basics*, John Wiley & Sons, New York (1986).

[Kap58] E. L. Kaplan, "Monte Carlo Methods of Equilibrium Solutions in Neutron Multiplication," UCRL-5275-T, University of California (1959).

[Kap74] H. G. Kaper, A. J. Lindeman and G. K. Leaf, "Benchmark Values for the Slab and Sphere Criticality Problem in One-Group Neutron Transport Theory," *Nucl. Sci. Eng.*, **54**, 94 (1974).

[Lar95] E. W. Larsen and R. P. Rulko, Private Communications (1995).

[Leh51] D. H. Lehmer, "Mathematical Methods in Large-Scale Computing Units," *Ann. Comp. Lab. Harvard Univ.*, **26**, 141 (1951).

[Lew93] E. E. Lewis and W. F. Miller, Jr. , *Computational Methods of Neutron Transport*, American Nuclear Society, Inc., La Grange Park, Illinois, (1993).

[Lie68] J. Lieberoth, "A Monte Carlo Technique to Solve the Static Eigenvalue Problem of the Boltzmann Transport Equation," *Nukleonik*, **11**, 5, 203 (1968).

[Lux89] I. Lux, "Note on Efficient Correlated Monte-Carlo Games," *Ann. Nucl. Energy*, **16**, 1, 39 (1989).

[Lux91] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Florida (1991).

[Mac73] D. B. MacMillan, "Monte Carlo Confidence Limits for Iterated-Source Calculations," *Nucl. Sci. Eng.*, **50**, 73 (1973).

[Maj94] A. Majumdar and W. R. Martin, "Calculations of Reactivity Perturbations Using Correlated Sampling Monte Carlo," *Trans. Am. Nucl. Soc.*, **71**, 203 (1994).

[Maj95a] A. Majumdar and W. R. Martin, "Multiple Reactivity Calculation Using Single Correlated Sampling Monte Carlo Simulation," *Proc. Int. Conf. on Mathematics and Computations, Reactor Physics and Environmental Analyses*, Portland, Oregon, April 39 - May 4, 1995, Vol. 1, p. 85 (1995).

[Maj95b] A. Majumdar and W. R. Martin, "Development of a Multiple Perturbation Monte Carlo Methods for Criticality Problems and Implementation on Parallel Processors," *Joint Nuclear Conferences X ENFIR/III ENAN*, Auguas de Lindola, Brazil, August 7 - 11 (1995).

[Mar91] W. R. Martin, "Monte Carlo Methods on Advanced Computer Architectures," *Advances in Nuclear Science and Technology*, **22**, 105 (1991).

[Mar86] W. R. Martin, P. F. Nowak and J. A. Rathkopf, "Monte Carlo Photon Transport on a Vector Supercomputer," *IBM J. Res. and Dev.*, **30**, 193 (1986).

[Mar87a] W. R. Martin and F. B. Brown, "Status of Vectorized Monte Carlo for Particle Transport Analysis," *International Journal of Supercomputer Applications*, **1**(2), 11 (1987).

[Mar87b] W. R. Martin, T. C. Wan, T. S. Abdel-Rahman and T.N. Mudge, "Monte Carlo Photon Transport on Shared Memory and Distributed Memory Parallel Processors," *International Journal of Supercomputer Applications*, **1**(3), 57 (1987).

[Mar93] W. R. Martin, A. Majumdar, J. A. Rathkopf and M. Litvin, "Experiences with Different Parallel Programming Paradigms for Monte Carlo Particle Transport Leads to a Portable Toolkit for Parallel Monte Carlo," *Proc. Int. Joint Conf. Mathematical Methods and Supercomputing in Nuclear Applications*, Karlsruhe, Germany, April 19-23, 1993, Vol. II, p. 418 (1993).

[Mat72] W. Matthes "Calculation of Reactivity Perturbations with the Monte Carlo Method," *Nucl. Sci. Eng.*, **47**, 234 (1972).

[Mat94] S. Matsuura, F. B. Brown and R. N. Blomquist, "Parallel Monte Carlo Eigenvalue Calculations," *Trans. Am. Nucl. Soc.*, **71**, 199 (1994).

[Men68] M. R. Mendelson, "Monte Carlo Criticality Calculations for Thermal Reactors," *Nucl. Sci. Eng.*, **32**, 319 (1968).

[Mih67] J. T. Mihalczo, "Multiplication of Uranium Metal by One Velocity Monte Carlo Calculations," *Nucl. Sci. Eng.*, **27**, 557 (1967).

[Mik67] G. A. Mikhailov, "Calculation of System-parameter Derivatives of Functionals of the Solutions to the Transport Equation," Zh. Vychisl. Mat. Mat. Fiz., **7**, 915 (1967) (in Russian).

[Mil67] L. B. Miller, *Monte Carlo Analysis of Reactivity Coefficients in Fast Reactors, General Theory and Applications*, ANL-7307 (TID 4500) Report, Argonne National Laboratory (1967).

[Moo76] J. G. Moore, "The Solution of Criticality Problems by Monte Carlo Methods," *Adv. Nucl. Sci. Tech.*, **9**, 73 (1976).

[Nak77] S. Nakamura, *Computational Methods in Engineering and Science*, Wiley, New York (1977).

[Nel85] W. R. Nelson, H. Hirayama and D. W. O. Rogers, "The EGS4 Code System," Standford Linear Accelerator Center Report SLAC-265 (Stanford Calif) (1985).

[NRC85] Nuclear Regulatory Commission, "Millstone Nuclear Power Station Unit 3 Final Safety Analysis Report," U.S. Nuclear Regulatory Commission Docket Number 50-423, Amendment 13 (1985).

[Rie84] H. Rief, "Generalized Monte Carlo Perturbation Algorithms for Correlated Sampling and A Second-order Taylor Series Approach," *Ann. Nucl. Energy*,**11**, 9, 455 (1984).

[Rie86] H. Rief, E. M. Gelbard, R. W. Schaefer and K. S. Smith, "Review of Monte Carlo Techniques for Analyzing Reactor Perturbations," *Nucl. Sci. Eng.*, **92**, 289 (1986).

[Rie88] H. Rief, "Monte Carlo Uncertainty Analysis," *Uncertainty Analysis*, edited by Y. Ronen, CRC Press, Inc., 188 (1988).

[Rub81] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New York (1981).

[Sei91] E. Seifert, "Calculation of Reactivity Effects by the Monte Carlo Perturbation Code OMEGA/P," *Proceeding of the International Conference on Nuclear Criticality Safety*, Oxford, September, 1991, p. II-11 (1991).

[Spa69] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*, Addison-Wesley, Reading, Massachusetts (1969).

[Tak70] H. Takahashi, "Monte Carlo Method for Geometrical Perturbation and its Application to the Pulsed Fast Reactor," *Nucl. Sci. Eng.*, **41**, 259 (1970).

[Urb95a] T. J. Urbatsch, R. A. Forster, R. E. Prael and R. J. Beckman, "Understanding the Three-Combined k_{eff} confidence Intervals in MCNP," *Proc. of the Fifth Int. Conf. on Nuclear Criticality Safety*, Albuquerque, New Mexico, September 17-21, (1995).

[Urb95b] T. J. Urbatsch, "Iterative Acceleration Methods for Monte Carlo and Deterministic Criticality Calculations," Ph.D. Dissertation, The University of Michigan, Ann Arbor, Michigan (1995).

[Uss55] L. N. Ussachoff, "Equations for the Importance of Neutrons," *Proc. U.N. Int. Conf. Peaceful Uses At. Energy*, **5**, 1656 (1955).

[Var62] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J. (1962).

[Wal94] S. Walters, K. Butterfield and D. Dudziak, "Analysis of Void Formation in SHEBA II Using S_N and Monte Carlo Codes," *Trans. Am. Nucl. Soc.*, **70**, 184 (1994).

[Wac66] E. L. Wachspress, *Iterative Solution of Elliptic Systems*, Prentice-Hall, Englewood Cliffs, N.J. (1966).

[Whi71] G. E. Whitesides, "A Difficulty in Computing the K-effective of the World," *Trans. Am. Nucl. Soc.*, **14**, 680 (1971).

[Whi66] G. E. Whitesides, G. W. Morrison and E. C. Crume, "Few-Group Monte Carlo Criticality Calculations," *Trans. Am. Nucl. Soc.*, **9**, 133 (1966).

[Zol83] V. G. Zolotukhin and L. V. Maiorov, "An Estimate of the Systematic Errors in the Calculations of Criticality by the Monte Carlo Method," Atomnaya Energia, **55**, 3, 173 (1983).

ABSTRACT

DEVELOPMENT OF A MULTIPLE PERTURBATION MONTE CARLO METHOD FOR EIGENVALUE PROBLEMS AND IMPLEMENTATION ON PARALLEL PROCESSORS

by

Amitava Majumdar

Chairperson: William R. Martin

We have developed a Monte Carlo method that calculates multiple perturbation effects in the eigenvalue (K) of the Boltzmann transport equation for neutrons from a single Monte Carlo simulation. Two Monte Carlo techniques, source iteration and fission matrix approaches, have been described. We have shown that subtracting two independent Monte Carlo simulations for eigenvalue perturbation calculation encounters difficulties. It is necessary to utilize some type of Monte Carlo perturbation technique. We have shown that the combination of the correlated sampling and source iteration methods encounters difficulties in calculating eigenvalue perturbations. When the correlated sampling approach is combined with the fission matrix approach, it can successfully evaluate eigenvalue perturbations. We have implemented the idea of performing Monte Carlo simulation in an artificial reference

system. Utilizing the fission matrix approach, correlated sampling, and an artificial reference system, we have developed the multiple perturbation technique. The actual simulation is done in an artificial reference system and all the perturbed and unperturbed systems' fission matrices are correlated to that reference system. At the end of the simulation, the dominant eigenvalue of the unperturbed and all perturbed fission matrices are evaluated numerically. This provides us with multiple ΔKs from a single Monte Carlo simulation. We have tested this method for different test problems and the results compared well with that of the TWODANT S_N transport code. This method allowed significant savings in computational effort.

We have implemented fixed source and eigenvalue algorithms for neutron transport on three different parallel machines, the BBN Butterfly, KSR-1, and IBM-SP2. We have addressed the issue of parallel random number generators and showed how the fixed source and eigenvalue parallel algorithms differ. Theoretical models for speedups have been developed and have compared well with the observed speedups. Close to linear speedups were observed for many of the test problems.