1

A FULLY COUPLED MONTE CARLO/DISCRETE ORDINATES SOLUTION TO THE NEUTRON TRANSPORT EQUATION

by

Randal Scott Baker

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF NUCLEAR AND ENERGY ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN NUCLEAR ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1990

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Randal Scott Baker

entitled A FULLY COUPLED MONTE CARLO/DISCRETE ORDINATES SOLUTION TO THE NEUTRON TRANSPORT EQUATION

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of DOCTOR OF PHILOSOPHY.

William L. Filippone	10/31/90 Date
Barry D. Ganapol	10/31/90 Date
Raymond E. Alcouffe	10/31/90 Date
George Lamb	10/31/90 Date
Bruce Wood	10/31/90 Date

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertati to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director William L. Filippone	10/31/90 Date

STATEMENT BY THE AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: 

ACKNOWLEDGMENTS

The author would like to express his thanks to the Los Alamos National Laboratory in general, and specifically to Bob Little and Brad Clark, for providing the support that made this work possible. I also deeply appreciate the assistance of Ray Alcouffe in modifying TWODANT, and his patience in helping me learn some of its inner workings. Finally, I must also thank Bill Filippone for his efforts above and beyond the duties of dissertation director.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The neutron transport equation is solved by a hybrid method that iteratively couples regions where deterministic (S_N) and stochastic (Monte Carlo) methods are applied. Unlike previous hybrid methods, the Monte Carlo and S_N regions are fully coupled in the sense that no assumption is made about geometrical separation or decoupling. The hybrid method provides a new means of solving problems involving both optically thick and optically thin regions that neither Monte Carlo nor S_N is well suited for by themselves.

The fully coupled Monte Carlo/S_N technique consists of defining spatial and/or energy regions of a problem in which either a Monte Carlo calculation or an S_N calculation is to be performed. The Monte Carlo region may comprise the entire spatial region (with vacuum boundary conditions) for selected energy groups, or may consist of a rectangular area that is either completely or partially embedded in an arbitrary S_N region. The Monte Carlo and S_N regions are then connected through the common angular boundary fluxes, which are determined iteratively using the response matrix technique, and volumetric sources.

The hybrid method has been implemented in the S_N code TWODANT by adding special-purpose Monte Carlo subroutines to calculate the response matrices and volumetric sources, and linkage subroutines to carry out the interface flux

iterations. The common angular boundary fluxes are included in the S_N code as interior boundary sources, leaving the logic for the solution of the transport flux unchanged, while, with minor modifications, the diffusion synthetic accelerator remains effective in accelerating the S_N calculations. The special-purpose Monte Carlo routines used are essentially analog, with few variance reduction techniques employed. However, the routines have been succesfully vectorized, with approximately a factor of five increase in speed over the non-vectorized version.

The hybrid method is capable of solving forward, inhomogeneous source problems in $X - Y$ and $R - Z$ geometries. This capability includes multigroup problems involving upscatter and fission in non-highly multiplying ($k_{eff} \leq .8$) systems. The hybrid method has been applied to several simple test problems with good results.

CHAPTER 1.
INTRODUCTION

Knowledge about the distribution of neutrons in a medium is essential in many of today's applications in nuclear science. Most obviously, the distribution of neutrons in a reactor directly relates to the energy released by fission, thus determining power levels and safety margins. Many other applications also exist, however, such as oil well logging, in which the distribution of neutrons emitted from a source placed in a well can be correlated to the possible presence or absence of hydrocarbons in the surrounding soil. Both these, as well as other areas, require an accurate means of evaluating the neutron distribution.

The distribution of neutrons in a medium is best described by the Boltzmann transport equation, which can be derived from the requirements of particle conservation across an element of phase space.[1] If the distribution $N(\mathbf{r}, E, \mathbf{\Omega})\, d\mathbf{r}\, dE\, d\mathbf{\Omega}$ represents the number of neutrons in volume $d\mathbf{r}$ about location $\mathbf{r}$, with energy in dE about E, and moving in solid angle $d\mathbf{\Omega}$ about direction $\mathbf{\Omega}$, then the total track length per second of the particles in the phase space element $d\mathbf{r}\, dE\, d\mathbf{\Omega}$ is $v(E)\, N(\mathbf{r}, E, \mathbf{\Omega})\, d\mathbf{r}\, dE\, d\mathbf{\Omega}$, where $v(E)$ is the neutron speed. In transport theory, the quantity $v(E)\, N(\mathbf{r}, E, \mathbf{\Omega})$ is commonly defined as the angular neutron flux

$\varphi(\mathbf{r}, E, \mathbf{\Omega})$, and the linear, steady state Boltzmann transport equation is written in terms of the angular flux as

$$\begin{aligned} &\mathbf{\Omega} \cdot \nabla \varphi(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_T(\mathbf{r}, E)\, \varphi(\mathbf{r}, E, \mathbf{\Omega}) = \\ &\int_0^\infty dE' \int_{4\pi} d\mathbf{\Omega}'\, \Sigma_S(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega}' \cdot \mathbf{\Omega})\, \varphi(\mathbf{r}, E', \mathbf{\Omega}') \\ &+ \chi(E) \int_0^\infty dE'\, \nu(E')\, \Sigma_F(\mathbf{r}, E')\, \varphi(\mathbf{r}, E') + S(\mathbf{r}, E, \mathbf{\Omega}). \end{aligned} \tag{1-1}$$

The total macroscopic cross section $\Sigma_T(\mathbf{r}, E)$ is the number of neutron interactions of all types per unit track length, the differential macroscopic scattering cross section $\Sigma_S(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega}' \cdot \mathbf{\Omega})\, dE\, d\mathbf{\Omega}$ represents the number of particles per unit track length scattered from energy E' and direction $\mathbf{\Omega}'$ to dE about E and $d\mathbf{\Omega}$ about $\mathbf{\Omega}$, and the fission macroscopic cross section $\Sigma_F(\mathbf{r}, E)$ is the number of fission interactions occuring per unit track length. The average number of neutrons emmitted per fission is defined as $\nu(E)$, while the fission spectrum $\chi(E)\, dE$ represents the probability of a fission particle appearing in energy range dE about E. The scalar flux $\varphi(\mathbf{r}, E)$ is defined as the integral $\int_{4\pi} d\mathbf{\Omega}'\, \varphi(\mathbf{r}, E, \mathbf{\Omega}')$, and the inhomogeneous source $S(\mathbf{r}, E, \mathbf{\Omega})\, d\mathbf{r}\, dE\, d\mathbf{\Omega}$ is the number of neutrons emitted in volume $d\mathbf{r}$ about location $\mathbf{r}$, with energy in dE about E, and moving in solid angle $d\mathbf{\Omega}$ about direction $\mathbf{\Omega}$.

The first term on the left-hand side of Eq. (1-1) is interpreted as the rate of change of particles in the phase space due to streaming, while the second term represents the loss of particles from the phase space due to all types of collisions. The first term on the right-hand side is the rate of gain of particles in the phase space due to scattering from all phase space elements at $\mathbf{r}$, while the second term is the gain due to fissions.

Several physical assumptions are inherent in Eq. (1-1). Scattering is assumed to be dependent only upon the scattering angle $\mathbf{\Omega} \cdot \mathbf{\Omega}'$, which is valid for isotropic media. The fission spectrum $\chi(E)$ is assumed independent of the fission energy E', and is also assumed to be identical for both prompt and delayed fission neutrons. The angular distribution of fission neutrons is assumed to be isotropic, which is reasonable at energies under consideration here ($\leq$ 20 MeV). To preserve linearity, macroscopic cross sections are assumed to be independent of the neutron flux.

Equation (1-1) is an integro-differential in six variables (three spatial, two angular, and one in energy), and a general solution has not been found. While analytic solutions have been obtained for simplified cases,[1] realistic problems require a numerical solution by one of several methods, which can be divided into two general classes – deterministic, of which the most popular method is discrete ordinates (S_N), and probabilistic (Monte Carlo).

While both S_N and Monte Carlo are well suited for specific types of problems, neither is efficient for all types. The S_N method is difficult to apply to problems with complicated geometries and encounters numerical difficulties in low scattering regions. While, in theory, Monte Carlo can be used to solve any problem, the computational time required for an accurate solution can be enormous, especially for problems with highly scattering regions.

Previous efforts at developing hybrid techniques that enable the best features of discrete ordinates and Monte Carlo to be exploited in a single problem have been limited. One of the first approaches was to employ the results of a one-dimensional adjoint S_N calculation as a biasing function in a three-dimensional Monte Carlo

shielding problem.[2] Note, however, that the Monte Carlo and S_N methods were not directly linked; rather, the S_N results provided additional information to the Monte Carlo, which performed the actual calculation. In the code DOMINO (Discrete Ordinates Monte Carlo Interface Operation),[3,4] the boundary fluxes resulting from a discrete ordinates code were used as input to the Monte Carlo code MORSE, either directly as a source, or indirectly as an importance function. However, the S_N and Monte Carlo regions were assumed to be decoupled, in that the solution to the S_N problem was not affected by the Monte Carlo region. Thus, even though the methods were coupled, the problem geometry was not. Similarly, Monte Carlo calculations have been used to provide a surface source (forward or adjoint) for S_N calculations,[5-7] again, with a decoupled geometry.

Another means of coupling Monte Carlo and S_N is the first collision source method,[8] in which source particles from a localized source are tracked to their first collision via Monte Carlo, then tallied to form a source for the S_N. This removes the difficulties encountered in running S_N with singular sources, such as a point source, while the time involved in the Monte Carlo calculation is minimal, since particles are only followed until their first collision. However, S_N is still used to compute the final fluxes throughout the entire problem geometry, so the first collision source method only partially alleviates any problems due to the presence of low scattering regions.

In Ref. 9, Monte Carlo methods are implemented within the S_N method itself. This is accomplished by choosing the S_N quadrature directions randomly, which are then used with standard S_N techniques. The use of random angular

quadrature directions helps mitigate ray effect problems, but does not eliminate errors due to spatial differencing (i.e., problems due to optically thin regions).

In contrast, the fully coupled S_N/Monte Carlo technique described here consists of defining spatial regions in which either a Monte Carlo calculation or an S_N calculation is to be done. The regions are then connected through the common boundary fluxes, which are determined iteratively using the response matrix technique. This technique is completely consistent in that the solution obtained is not dependent upon any assumptions of geometric separation or decoupling.

As an example of a configuration suitable for analysis by the fully coupled hybrid technique, consider a geometrically complicated neutron source located in a low scattering material, such as air, encased in a large cylinder of highly scattering material, such as steel. Due to the complex source geometry and streaming in air, the S_N technique by itself would be unreliable. On the other hand, Monte Carlo alone would be inefficient because of the significant multiple scattering in the steel. Even a decoupled linking technique that used a Monte Carlo calculation in the source and air regions, followed by an S_N analysis in the steel, would be inadequate if particles were likely to reenter the air region after having entered the steel region. By using the Monte Carlo technique for the air and source regions, the S_N technique for the steel region, and the response matrix method to determine the interface currents between the air and steel regions, we are able to apply both methods where they are most efficient and still obtain a fully coupled solution.

The hybrid technique, as originally developed by Alcouffe and Filippone,[10,11] was first implemented for monoenergetic transport in simple $X-Y$ geometry problems, thus demonstrating the feasibility of the method for one-group problems in two dimensions. (Although three dimensional S_N codes are currently under development, past limits in computational resources have restricted development of practical discrete ordinates codes to two dimensions.) Although application of the hybrid method to realistic problems would require the inclusion of multiple energy groups, a "brute force" extension of the method was not possible due to physical limits on data storage. In addition, numerous other features and capabilities were needed in order to handle the complexities posed by realistic problems. Although the hybrid method showed promise, it was unclear if it could be extended to more complicated problems.

The goal of this dissertation is to determine the feasibility of the application of the hybrid method to problems of real world complexities by:

a. Extending the hybrid method to include multiple energy groups.

b. Generalizing the treatment of neutron scattering to allow anisotropic scattering.

c. Including the capability for treating fission and upscatter.

d. Allowing the treatment of more complicated geometries, including cylindrical.

e. Enhancing the efficiency of the Monte Carlo portion of the hybrid method through vectorization.

f. Increasing the efficiency of the S_N portion of the hybrid method through improved acceleration techniques.

g. Developing a method for estimating the variance of quantities calculated by stochastic techniques in the hybrid method.

Chapters 2 and 3 of this dissertation will provide a brief review of discrete ordinates and Monte Carlo techniques, respectively, while Chapter 4 will detail the theory behind the response matrix hybrid technique. Chapter 5 will describe the actual implementation of the hybrid method, i.e., how the Monte Carlo and S_N are coupled together. Chapter 6 describes the specifics of the Monte Carlo techniques used in the hybrid method, while Chapter 7 covers the vectorization of the hybrid Monte Carlo method. Chapter 8 describes how multigroup cross sections are obtained for the Monte Carlo from the S_N, and explores some of the related effects of using such derived cross sections. The use of statistics and variance estimators in the hybrid method is explained in Chapter 9, while the use of diffusion synthetic accleration in the S_N with the hybrid method is covered in Chapter 10. Chapter 11 presents a series of benchmark comparisons between pure S_N and the hybrid method. Finally, conclusions and recommendations about the hybrid method are presented in Chapter 12.

CHAPTER 2.
DISCRETE ORDINATES

The discrete ordinates (S_N) method is a deterministic means of solving the transport equation through discretization of the independent variables. The energy domain E is subdivided into G groups (i.e., the multigroup method is used), and the spatial domain $\mathbf{r}$ is partitioned into "rectangular" cells. A set of discrete angular variables (ordinates) is then selected and the transport equation evaluated along these directions. Because of the nature of the linear Boltzmann integro-differential equation, its solution is most conveniently developed by a process of von Neumann or source iteration.

Many discrete ordinates codes currently exist today, some of which have been constructed in a different manner. The following review of discrete ordinates is tailored to TWODANT,[12] the S_N code within which the hybrid method has been implemented, and is based upon Refs. 13 and 14.

2.1 Coordinate Systems

A two dimensional $X - Y$ geometry consists of a block with finite x and y dimensions, and an infinite length in the z direction. In the coordinate system

shown at Fig. 2-1. μ is the direction cosine along $\hat{\mathbf{e}}_x$, η the direction cosine along $\hat{\mathbf{e}}_y$, and ξ the direction cosine along $\hat{\mathbf{e}}_z$. The phase space thus consists of five dimensions – x, y, E, μ, and η, since we have the constraint $\mu^2 + \eta^2 + \xi^2 = 1$.

A two dimensional $R-Z$ geometry consists of a finite cylinder with radius R, height H, and azimuthal symmetry. In the coordinate system shown at Fig. 2-2. μ is the direction cosine alone $\hat{\mathbf{e}}_r$, η the direction cosine along $\hat{\mathbf{e}}_z$, and ξ the direction cosine along $\hat{\mathbf{e}}_\theta$. Again, the phase space has five dimensions – r, z, E, μ, and η.

2.2 The Multigroup Method

The energy groups are represented by the group index g, where increasing g corresponds to decreasing energy. Let $0 < E_1 < E_2 < \ldots < E_{G-1} < E_G$, where E_G is the maximum energy of interest in the system, then the definition of the group flux is

$$\phi_g(\mathbf{r}, \mathbf{\Omega}) \equiv \int_{E_{g-1}}^{E_g} dE'\, \phi(\mathbf{r}, E', \mathbf{\Omega}). \qquad (2-1)$$

Similarly, the group source is

$$S_g(\mathbf{r}, \mathbf{\Omega}) \equiv \int_{E_{g-1}}^{E_g} dE'\, S(\mathbf{r}, E', \mathbf{\Omega}). \qquad (2-2)$$

The group total and fission cross sections are defined so that

$$\Sigma_{T,g}(\mathbf{r}, \mathbf{\Omega}) \equiv \frac{1}{\phi_g(\mathbf{r}, \mathbf{\Omega})} \int_{E_{g-1}}^{E_g} dE'\, \phi(\mathbf{r}, E', \mathbf{\Omega})\, \Sigma_T(\mathbf{r}, E'), \qquad (2-3a)$$

$$\nu\Sigma_{F,g}(\mathbf{r}, \mathbf{\Omega}) \equiv \frac{1}{\phi_g(\mathbf{r}, \mathbf{\Omega})} \int_{E_{g-1}}^{E_g} dE'\, \phi(\mathbf{r}, E', \mathbf{\Omega})\, \nu(E')\, \Sigma_F(\mathbf{r}, E'), \qquad (2-3b)$$

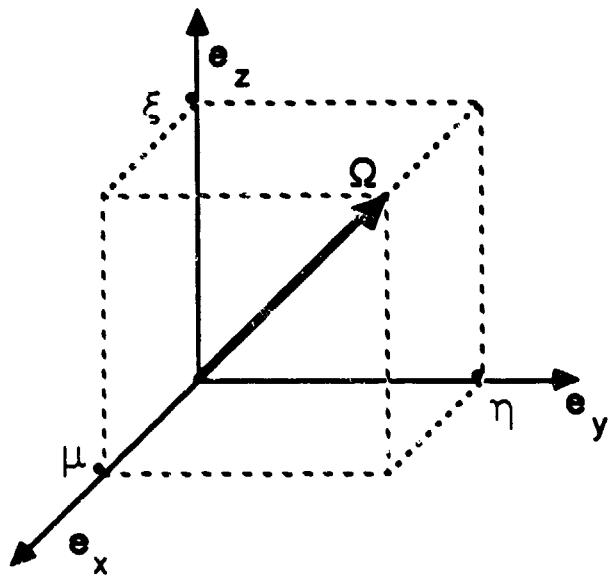

Figure 2-1 Cartesian Coordinate System

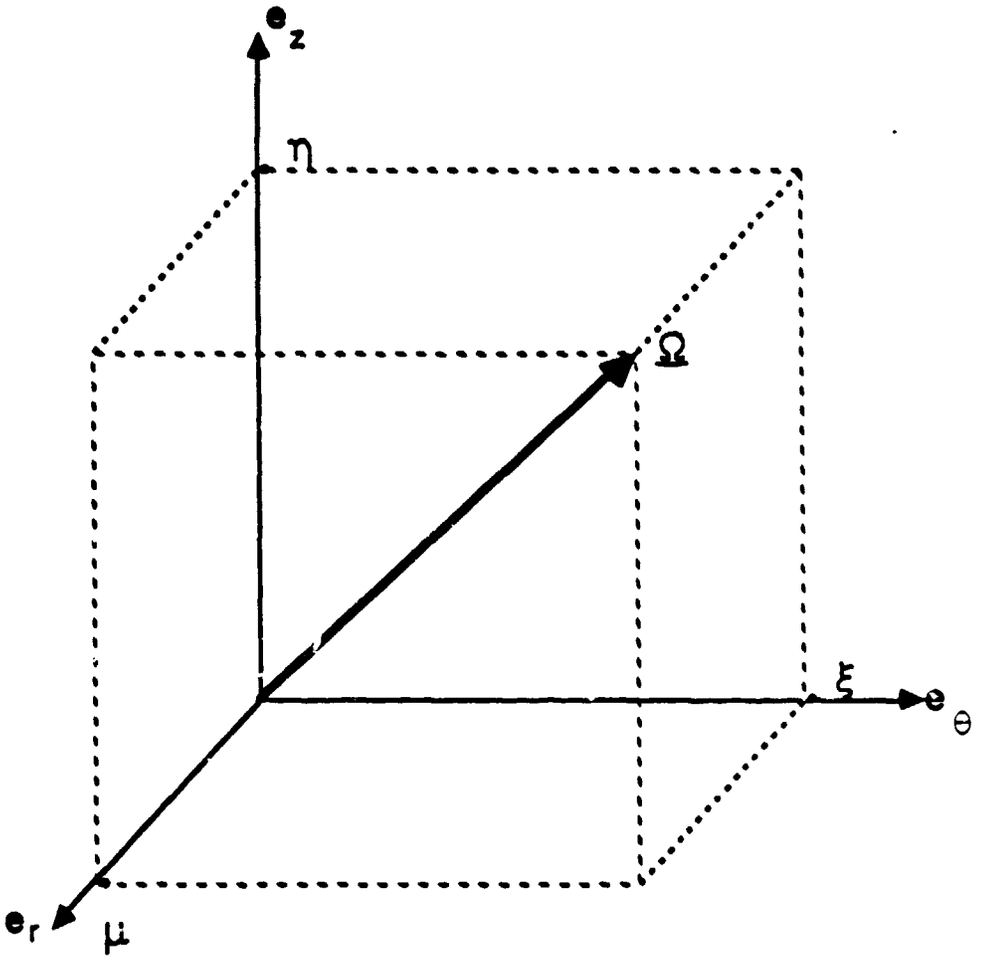

Figure 2-2 Cylindrical Coordinate System

and the group-to-group transfer cross section is

$$\Sigma_{g'-g}(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) \equiv \frac{1}{\phi_{g'}(\mathbf{r}, \mathbf{\Omega})} \int_{E_{g-1}}^{E_g} dE \int_{E_{g'-1}}^{E_{g'}} dE' \, \phi(\mathbf{r}, E', \mathbf{\Omega}) \Sigma_S(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega}' \cdot \mathbf{\Omega}). \tag{2-3c}$$

Since the angular flux $\phi(\mathbf{r}, E, \mathbf{\Omega})$ is not known a priori, multigroup cross sections are generally calculated by assuming that the angular flux is separable into functions $f(\mathbf{r}, \mathbf{\Omega}) g(E)$, where $f(\mathbf{r}, \mathbf{\Omega})$ need not be determined, and the spectral weighting function $g(E)$ is estimated. Then, the group macroscopic cross sections are functions of $\mathbf{r}$ only.

With the definitions at Eqns. (2-3), integration of Eq. (1-1) over the energy group g results in

$$\mathbf{\Omega} \cdot \nabla \phi_g(\mathbf{r}, \mathbf{\Omega}) + \Sigma_{T,g}(\mathbf{r}) \phi_g(\mathbf{r}, \mathbf{\Omega}) = \sum_{g'=1}^{g} \int_{4\pi} d\mathbf{\Omega}' \, \Sigma_{g'-g}(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) \phi_{g'}(\mathbf{r}, \mathbf{\Omega}')$$

$$+ \sum_{g'=g+1}^{G} \int_{4\pi} d\mathbf{\Omega}' \, \Sigma_{g'-g}(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) \phi_{g'}(\mathbf{r}, \mathbf{\Omega}') + \chi_g \sum_{g'=1}^{G} \nu \Sigma_{F,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + S_g(\mathbf{r}, \mathbf{\Omega}). \tag{2-4}$$

where χ_g is defined as $\int_{E_{g-1}}^{E_g} dE' \, \chi(E')$ and $\phi_g(\mathbf{r})$ as $\int_{E_{g-1}}^{E_g} dE' \, \phi(\mathbf{r}, E')$.

Note that the first term on the left-hand side is due to neutrons scattering from higher energy groups to lower energy groups (plus within group scatter), and is referred to as "downscatter", while the second term is due to neutrons scattering from lower energy groups to higher energy groups, and is referred to as "upscatter". For problems without upscatter or fission, the multigroup equation may be solved directly by solving for the flux in group one, calculating its contribution to the second and lower groups, solving for the second group's flux, etc., until all groups have been solved. With upscatter or fission, an iterative procedure is most conveniently

used. Letting p be the iteration index, and using the updated scatter source where available, Eq. (2-4) becomes

$$\mathbf{\Omega}\cdot\nabla\,o_g^{p+1}(\mathbf{r},\mathbf{\Omega})+\Sigma_{T,g}(\mathbf{r})\,o_g^{p+1}(\mathbf{r},\mathbf{\Omega})=\sum_{g'=1}^{g}\int_{4\pi}d\mathbf{\Omega}'\,\Sigma_{g'-g}(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega})\,o_{g'}^{p+1}(\mathbf{r},\mathbf{\Omega}')$$

$$+\sum_{g'=g+1}^{G}\int_{4\pi}d\mathbf{\Omega}'\,\Sigma_{g'-g}(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega})\,o_{g'}^{p}(\mathbf{r},\mathbf{\Omega}')+\chi_g\sum_{g'=1}^{G}\nu\Sigma_{F,g'}(\mathbf{r})\,o_{g'}^{p}(\mathbf{r})+S_g(\mathbf{r},\mathbf{\Omega}). \tag{2-5}$$

This iteration procedure is referred to as the "outer" iteration in discrete ordinates codes.

2.3 The Scattering Cross Section

Before we proceed with the discretization of the angular variables, the scattering cross section must be further specified. Letting $\mu_0=\mathbf{\Omega}'\cdot\mathbf{\Omega}$, and expanding the group-to-group transfer cross section in Legendre polynomials, we obtain

$$\Sigma_{g'-g}(\mathbf{r},\mu_0)=\sum_{l=0}^{L}\frac{2l+1}{4\pi}\Sigma_{l,g'-g}(\mathbf{r})\,P_l(\mu_0). \tag{2-6}$$

where $P_l(\mu_0)$ is the lth Legendre polynomial and the infinite series representation has been truncated after L terms. Truncation at $L=0$ corresponds to isotropic scattering, while truncation at $L=1$ is referred to as linearly anisotropic scattering. Since the Legendre polynomials are orthogonal, the coefficients of the expansion are

$$\Sigma_{l,g'-g}(\mathbf{r})=2\pi\int_{-1}^{+1}d\mu_0\,\Sigma_{g'-g}(\mathbf{r},\mu_0)\,P_l(\mu_0). \tag{2-7}$$

The lth Legendre polynomial can be expressed as

$$P_l(\mu_0) = \frac{4\pi}{2l+1} \sum_{m=-l}^{m=+l} Y_{lm}(\mathbf{\Omega}) Y_{lm}^*(\mathbf{\Omega}'), \qquad (2-8)$$

by using the addition theorem for spherical harmonics, with $Y_{l,m}(\mathbf{\Omega})$ representing the lth, mth spherical harmonic, and $Y_{l,m}^*(\mathbf{\Omega})$ its complex conjugate. If we now expand the group angular flux in terms of the spherical harmonics, i.e.,

$$\begin{aligned} o_g(\mathbf{r}, \mathbf{\Omega}) &= \sum_{l=0}^{L} \sum_{m=-l}^{m=+l} Y_{lm}(\mathbf{\Omega}) \tilde{o}_{lmg}(\mathbf{r}) \\ \tilde{o}_{lmg}(\mathbf{r}) &= \int_{4\pi} d\mathbf{\Omega} Y_{lm}^*(\mathbf{\Omega}) o_g(\mathbf{r}, \mathbf{\Omega}). \end{aligned} \qquad (2-9)$$

The scattering source (including both upscatter and downscatter) then becomes

$$\sum_{g'=1}^{G} \int_{4\pi} d\mathbf{\Omega}' \Sigma_{g'-g}(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) o_{g'}(\mathbf{r}, \mathbf{\Omega}') = \sum_{g'=1}^{G} \sum_{l=0}^{L} \Sigma_{l,g'-g}(\mathbf{r}) \sum_{m=-l}^{m=+l} Y_{lm}(\mathbf{\Omega}) \tilde{o}_{lmg'}(\mathbf{r}). \qquad (2-10)$$

Note that because of symmetry considerations for two dimensional geometries, the moments $\tilde{o}_{lmg'}(\mathbf{r})$ are zero for $m < 0$.[13]

2.4 Discretization of the Angular Variable

In discrete ordinates, the unit directional sphere is represented by a set of discrete directions $\mathbf{\Omega}_n$ with associated weights w_n (see Fig. 2-3). Although there is no unique quadrature set of directions, all such sets are chosen so as to preserve physical symmetries and properties of the transport equation.[15] The method of discretization is independent of the actual quadrature set used; however, it is somewhat dependent upon the specified geometry.

Beginning with Cartesian geometry, the streaming operator is $\mathbf{\Omega} \cdot \nabla = \mu \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y}$. Note that there is no angular coupling in the streaming operator in $X - Y$ geometry, i.e., as a particle travels through space its direction cosines μ, η, and ξ remain constant until it suffers a collision. Along the quadrature direction $\mathbf{\Omega}_n$ with direction cosines μ_n and η_n, therefore, the transport equation is

$$\left[\mu_n \frac{\partial}{\partial x} + \eta_n \frac{\partial}{\partial y}\right] \phi_{ng}(\mathbf{r}) + \Sigma_{T,g}(\mathbf{r})\, \phi_{ng}(\mathbf{r}) = \sum_{g'=1}^{G} \sum_{l=0}^{L} \Sigma_{l,g'-g}(\mathbf{r}) \sum_{m=0}^{m=l} Y_{lmn}\, \tilde{\phi}_{lmg'}(\mathbf{r}) + \chi_g \sum_{g'=1}^{G} \nu\Sigma_{F,g'}(\mathbf{r})\, \tilde{\phi}_{00g'}(\mathbf{r}) + S_{ng}(\mathbf{r}). \tag{2-11}$$

where $Y_{lmn} = Y_{lm}(\mathbf{\Omega}_n)$, $S_{ng}(\mathbf{r}) = S_g(\mathbf{r}, \mathbf{\Omega}_n)$, and the scalar flux $\phi_{g'}(\mathbf{r})$ has been rewritten as the zeroth moment $\tilde{\phi}_{00g'}(\mathbf{r})$ for clarity.

In cylindrical geometry, however, the discretization is not as straightforward, since the streaming operator becomes $\frac{\mu}{r}\frac{\partial(r\bullet)}{\partial r} - \frac{1}{r}\frac{\partial(\xi\bullet)}{\partial\theta} + \eta\frac{\partial(\bullet)}{\partial z}$. Although the first and third terms are evaluated at $\mathbf{\Omega}_n$ as before, the second term represents an angular redistribution where the direction cosines μ and ξ change as the particle streams through the cylinder (see Fig. 2-4). This is reflected in the discretization process by introducing coupling coefficients[13] such that

$$\left.\frac{\partial[\xi\phi_g(\mathbf{r}, \mathbf{\Omega})]}{\partial\theta}\right|_{\mathbf{\Omega}_n} = \frac{1}{w_n}[\alpha_{n+1/2}\, \phi_{n+1/2,g}(\mathbf{r}) - \alpha_{n-1/2}\, \phi_{n-1/2,g}(\mathbf{r})]. \tag{2-12}$$

The coupling coefficients are evaluated by requiring Eq. (2-12) to obey the conservation form of the neutron balance equation, and the streaming term go to zero when the angular flux is uniform and isotropic, with the results[14]

$$\alpha_{1/2} = 0; \qquad \alpha_{n+1/2} = \alpha_{n-1/2} - w_n \mu_n. \tag{2-13}$$

The diamond difference relationship in angle is used to evaluate the additional angular fluxes, i.e.,

$$o_{n,g}(\mathbf{r}) = \frac{1}{2}[o_{n+1/2,g}(\mathbf{r}) + o_{n-1/2,g}(\mathbf{r})]. \qquad (2-14)$$

The starting angular flux can be obtained through use of the "step-start" procedure, where $o_{i,j,3/2} = o_{ij,1}$ is used for each starting interval in angle.[13]

2.5 Spatial Discretization

Discretization of the spatial variable is performed by partitioning the spatial dimensions into intervals that form rectangular mesh cells, where the standard convention for denoting the mesh cells is shown at Fig. 2-5. The dimensions of the ith, jth mesh cell are $\Delta x_i = x_{i+1/2} - x_{i-1/2}$, $\Delta y_j = y_{j+1/2} - y_{j-1/2}$ in $X-Y$ geometry, and $\Delta r_i = r_{i+1/2} - r_{i-1/2}$, $\Delta z_j = z_{j+1/2} - z_{j-1/2}$ in $R-Z$ geometry. Cross sections are assumed to be constant within a cell, e.g., $\Sigma_{ijg} = \Sigma_g(\mathbf{r}_{ij})$, $x_{i-1/2} < x < x_{i+1/2}$, $y_{j-1/2} < y < y_{j+1/2}$ in Cartesian coordinates.

In $X-Y$ geometry, we integrate the transport equation over the ith, jth mesh cell for $d\mathbf{r} = dx\,dy$. With the definitions

$$o_{ijng} \equiv \frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} dx\,dy\, o_{ng}(\mathbf{r}), \qquad (2-15a)$$

$$\tilde{o}_{ijlmg} \equiv \frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} dx\,dy\, \tilde{o}_{lmg}(\mathbf{r}), \qquad (2-15b)$$

and

$$S_{ijng} \equiv \frac{1}{\Delta x_i \Delta y_j} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} dx\,dy\, S_{ng}(\mathbf{r}), \qquad (2-15c)$$

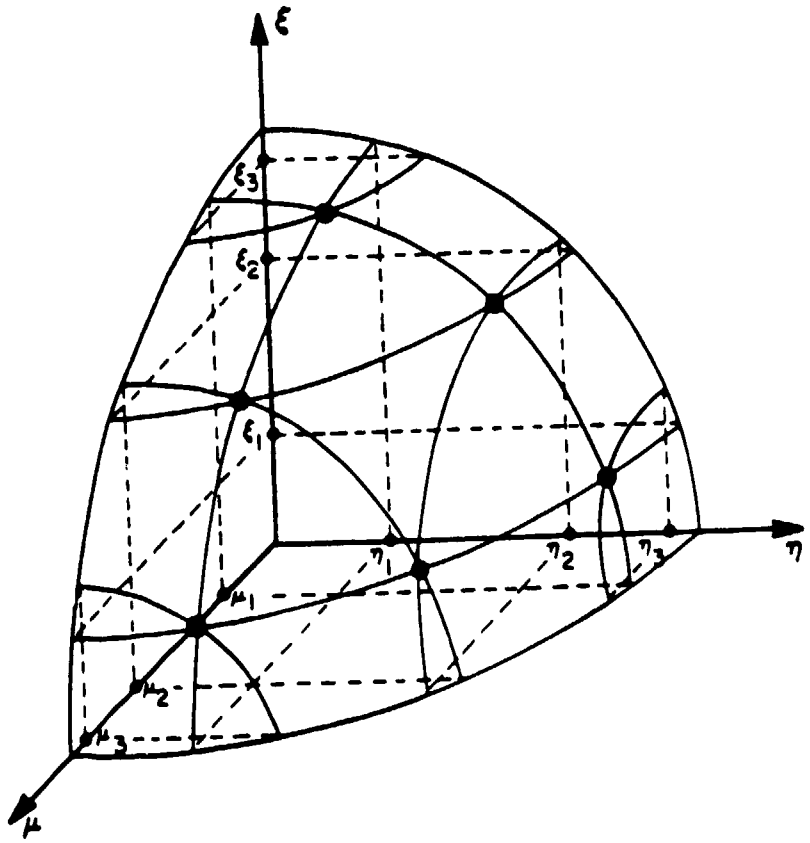

Figure 2-3 Sample S_6 Quadrature Set[15]

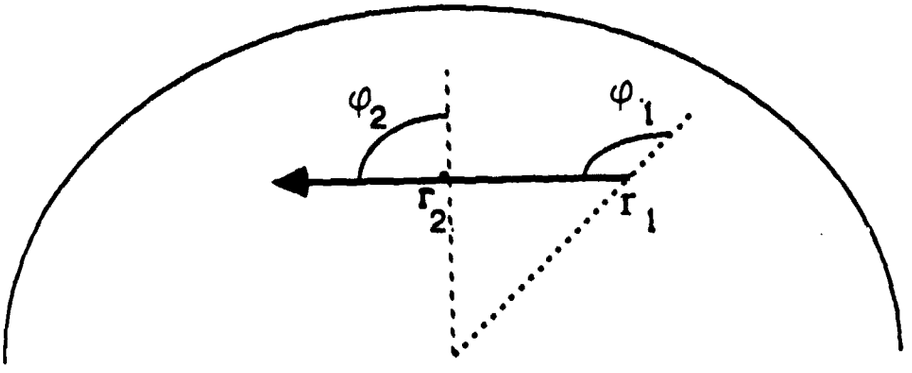

Figure 2–4 Angular Redistribution

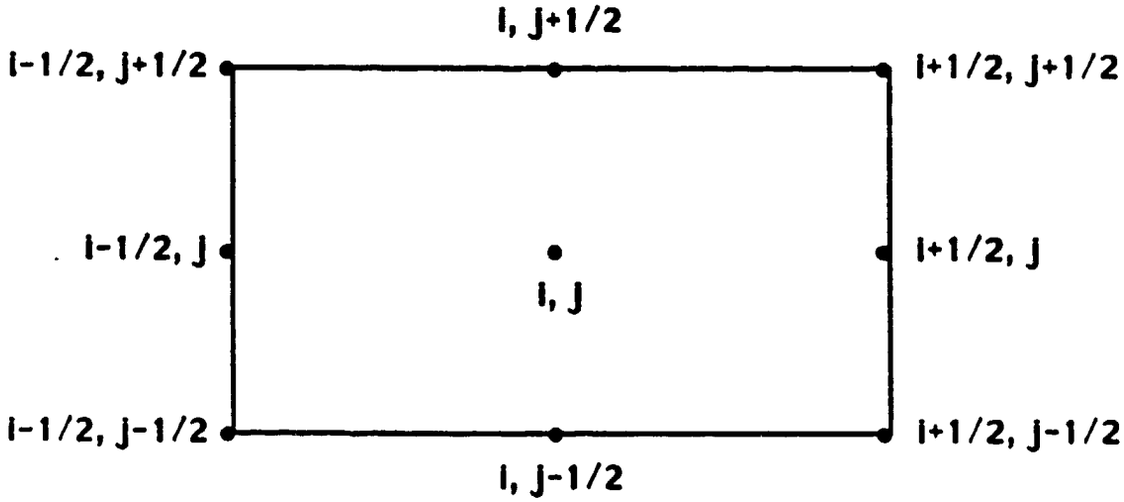

Figure 2-5 Sample Mesh Cell

the result is

$$\frac{\mu_n}{\Delta x_i}(\phi_{i+1/2,jng} - \phi_{i-1/2,jng}) + \frac{\eta_n}{\Delta y_j}(\phi_{i,j+1/2,ng} - \phi_{i,j-1/2,ng}) + \Sigma_{T,ijg}\,\phi_{ijng} = \sum_{g'=1}^{G}\sum_{l=0}^{L}\Sigma_{ijl,g'-g}\sum_{m=0}^{m=l}Y_{lmn}\,\tilde{\phi}_{ijlmg'} + \chi_g\sum_{g'=1}^{G}\nu\Sigma_{F,ijg'}\,\tilde{\phi}_{ij00g'} + S_{ijng}. \tag{2-16}$$

The volume element in $R-Z$ geometry is $d\mathbf{r} = 2\pi r\,dr\,dz$. Let the area of the ith face of a mesh cell be defined as $A_{i\pm1/2,j} = 2\pi r_{i\pm1/2}\,\Delta z_j$, and the volume of the ith, jth cell as $V_{ij} = \pi(r^2_{i+1/2} - r^2_{i-1/2})\Delta z_j$. Let the cell-averaged flux be defined as

$$\phi_{ijng} \equiv \frac{1}{V_{ij}}\int_{r_{i-1/2}}^{r_{i+1/2}}\int_{z_{j-1/2}}^{z_{j+1/2}} 2\pi r\,dr\,dz\,\phi_{ng}(\mathbf{r}). \tag{2-17}$$

with similar definitions for the moments and the fixed source. Upon intergrating the transport equation over $d\mathbf{r}$ for the ith, jth cell, we obtain[13]

$$\mu_n[A_{i+1/2,j}\,\phi_{i+1/2,jng} - A_{i-1/2,j}\,\phi_{i-1/2,jng}] - \frac{V_{ij}}{w_n}\left(\frac{1}{r}\right)_i[\alpha_{n+1/2}\,\phi_{ij,n+1/2,g} - \alpha_{n-1/2}\,\phi_{ij,n-1/2,g}] + \eta_n\,\pi[r^2_{i+1/2} - r^2_{i-1/2}][\phi_{i,j+1/2\,ng} - \phi_{i,j-1/2,ng}] + \Sigma_{T,ijg}\,\phi_{ijng}\,V_{ij} = \sum_{g'=1}^{G}\sum_{l=0}^{L}\Sigma_{ijl,g'-g}\sum_{m=0}^{m=l}Y_{lmn}\,\tilde{\phi}_{ijlmg'}\,V_{ij} + \chi_g\sum_{g'=1}^{G}\nu\Sigma_{F,ijg'}\,\tilde{\phi}_{ij00g'}\,V_{ij} + S_{ijng}\,V_{ij}. \tag{2-18}$$

where the approximation

$$\int_{r_{i-1/2}}^{r_{i+1/2}}\int_{z_{j-1/2}}^{z_{j+1/2}} 2\pi\,dr\,dz\,\phi_{ng}(\mathbf{r}) = \left(\frac{1}{r}\right)_i\phi_{ijng}\,V_{ij} \tag{2-19}$$

has been used. The quantity $(1/r)_i$ is determined by considering the case of the uniform, isotropic angular flux, from which

$$\left(\frac{1}{r}\right)_i = \frac{A_{i+1/2,j} - A_{i-1/2,j}}{V_{ij}}. \tag{2-20}$$

2.6 Solution of the Discretized Transport Equation

The discretized transport equation in $X-Y$ geometry [Eq. (2-16)] constitutes one equation with five unknowns - ϕ_{ijng}, $\varphi_{i\pm1/2,jng}$, and $\phi_{i,j\pm1/2,ng}$ for fixed i, j, n, and g. While two of the unknown cell edge fluxes are determined from either boundary conditions or the previous cell's results, auxiliary equations are needed to determine the remaining unknowns. One standard approach is that of diamond differencing, where the auxiliary equations

$$\phi_{ijng} = \frac{1}{2}[\phi_{i+1/2,jng} + \phi_{i-1/2,jng}] \qquad (2-21a)$$

and

$$\phi_{ijng} = \frac{1}{2}[\phi_{i,j+1/2,ng} + \phi_{i,j-1/2,ng}] \qquad (2-21b)$$

are used. With these, the cell-centered fluxes can be uniquely determined.

The discretized transport equation in $R-Z$ geometry [Eq. (2-18)] has two additional unknowns due to the angular redistribution term - $\phi_{ij,n\pm1/2}$. These are determined thorugh the diamond difference in angle relationship of Eq. (2-14), in conjuction with Eq. (2-13) and the step-start procedure.

Two of the most common boundary conditions used in discrete ordinates codes are those of vacuum boundaries and symmetry boundaries. For vacuum boundaries, the inconiing angular flux on the cell edge is fixed at zero, while for symmtery boundaries the value of the incoming flux is set equal to the value of the outgoing flux. A typical example of a symmetry boundary is that of the centerline in a cylinder.

The solution is obtained by an exact inversion of the discretized transport operator, then assuming that the source is known. The process is an iterative one, as can most clearly be seen by writing the transport equation as

$$[\Omega \cdot \nabla \phi_g^{k+1/2}(\mathbf{r})]_n + \Sigma_{T,g}(\mathbf{r})\, \phi_{ng}^{k+1/2}(\mathbf{r}) = \sum_{l=0}^{L} \Sigma_{l,g\to g}(\mathbf{r}) \sum_{m=0}^{m=+l} Y_{lmn}\, \tilde{\phi}_{lmg}^{k}(\mathbf{r}) + Q_{ng}(\mathbf{r}), \tag{2-22}$$

with

$$\tilde{\phi}_{lmg}^{k+1}(\mathbf{r}) = \sum_{n=1}^{N} w_n\, Y_{lmn}^{*}\, \phi_{ng}^{k+1/2}(\mathbf{r}), \tag{2-23}$$

where k is the iteration index and $Q_{ng}(\mathbf{r})$ represents the source due to upscatter and downscatter, fission, and the fixed source. This iteration process is called the "inner" iteration, and is continued until some convergence criteria has been met.

The cell-edge fluxes calculated using the diamond difference method are not guaranteed to be positive, and as such can be non-physical (i.e., negative). One common means of remedying this is through use of a negative flux fixup where the negative flux is set to zero.[13] The cell center flux is then recalculated, along with the remaining cell-edge flux, in order to preserve particle balance. This ensures a positive solution, although it also introduces nonlinearites into the computational process. However, negative flux fixup is usually not significant unless the mesh sizing exceeds approximately two mean free paths, or has highly asymmteric dimensions.

2.7 Limitations of Discrete Ordinates

One important problem inherent in the use of discrete ordinates is the occurrence of ray effects in highly absorping or vacuous mediums. (Note that a highly

downscattering group in a multigroup problem is equivalent to a high absorber). In these types of media, particles tend to concentrate along the discrete directions of travel, causing non-physical maxima and minima in the angular flux. Even the use of large quadrature sets often only partially alleviates the problem while greatly increasing computational time.

Another inherent problem is the difficulty in mocking up complicated geometries, since the geometry must be defined in terms of a finite mesh restricted to a two-dimensional slab or cylinder. This often forces the user to simplify the problem when analyzing such devices as a tool used for oil well logging. Similarly, it may be difficult to model geometries composed of objects with greatly varying sizes, e.g., a point source in a slab, or an atmospheric transport problem, where the size of the atmospheric layer dwarves the size of the ground layer.

It can also be difficult to select an appropriate mesh size. While a small mesh size ensures accuracy and provides detailed spatial information, it also greatly increases the computational time and memory requirements. Conversely, a large mesh size, while reducing the cost of the calculation, can result in a serious loss of accuracy.

While S_N provides an efficient and accurate means for solving many transport problems, the limitations described above prevent it from being the method of choice for all problems. More importantly, from our point of view, there exists a class of problems where S_N is an appropriate solution method for part of the problem geometry, and another method (i.e., probabilistic) is better suited for the

remainder. It is for these types of problems that the hybrid method subsequently described is intended for.

CHAPTER 3.
THE MONTE CARLO METHOD

The Monte Carlo method is a stochastic means of solving the transport equation through simulation of a finite number of neutron histories.[16,17] While the neutron distribution in the medium of interest typically consists of anywhere from 10^{10} to 10^{20} particles, we can obtain estimates of the distribution by randomly sampling a few thousand to a few million particles. As we sample more particles, we obtain better estimates, at a corresponding increase in computational time.

A neutron history consists of following a neutron from its creation by either a fixed source or fission until it is lost to the system by capture or leakage. Possible starting locations, directions, and energies are expressed in terms of probability distribution functions, which are sampled by drawing random numbers from a pseudo random number generator in order to determine the exact coordinates in phase space. Another random number is then drawn to determine the track length, which is compared with the calculated distance to the next boundary.

If the sampled track length is greater than the distance to the boundary, and the boundary is an exterior boundary, than the particle has leaked from the system and its history is terminated. If the boundary is an internal boundary, such as one separating regions with differing material compositions, the distance to the next boundary is calculated and compared with the remaining track length as above.

When the sampled track length is less than the distance to the next boundary, the particle undergoes a collision. The type of collision (e.g., radiative capture, fission, elastic scattering, etc.) is determined by random sampling. If the result is a capture-type reaction, the particle history is terminated. Otherwise, the neutron is continued on in a new direction and energy determined both by sampling from cross section data and from physical laws governing particle interactions (e.g., conservation of momentum). The track length is again randomly sampled, and the tracking process continued as above.

In analog Monte Carlo, the probability distributions are formulated so as to correspond to the analogous physical behavior of a neutron. This can lead to gross inefficiencies, as in the classic example of a shielding problem consisting of an absorbing slab with a source on one side and a detector on the other. If the slab is optically thick, the probability of an individual neutron reaching the detector is small and a large number of histories must be run in order to obtain an accurate estimate of the flux at the detector. Fortunately, numerous biasing procedures have been developed to enhance the efficiency of Monte Carlo methods in this and other problem areas.[16-18] In general, a biasing procedure modifies the appropriate probability distribution to increase the chance of a score, while the particle's importance (weight) is adjusted so as to maintain a "fair game". Introduction of these procedures is referred to as "non-analog" Monte Carlo.

While several sophisticated, general-purpose Monte Carlo codes currently exist, it was considered more advantageous to develop our own code for implementation with the hybrid method. Existing Monte Carlo codes are designed to accept

input from, and return output to, a user, while the hybrid method requires interaction with a discrete ordinates code (TWODANT), posing some unique problems as described in subsequent chapters. In addition, the relatively simple geometries and scoring techniques considered in the hybrid method enable the use of a semi-analog code whose efficiency can be improved through the use of vectorization (reference Chapter 8). Therefore, the remainder of this chapter is limited to a review of probability functions and error estimates basic to all Monte Carlo methods, while details of the actual method implemented will be described in later chapters.

3.1 Sampling Probability Functions

A probability density function (PDF) $p(x)$ is defined such that $p(x)\,dx$ is the probability that a random event will take on the value x' between x and $x + dx$.[16] Probability density functions are non-negative functions normalized such that the integral (for a continuous function) or sum (for a discrete function) over the applicable range is one. The cumulative distribution function (CDF) $P(x)$ is defined as the probability that a random number x' is less than or equal to x. Therefore, the CDF is a non-negative, non-decreasing function with a range of zero to one. If $f(x)$ is the function that we wish to sample over a range of $a < x < b$, then the PDF is $p(x) = f(x) / \int_a^b f(x')\,dx'$ and the CDF is $P(x) = \int_a^x p(x')\,dx'$.

Let $g(\zeta)$ represent a uniform distribution between zero and one such that $g(\zeta) = 1$. Since both $g(\zeta)$ and $p(x)$ are distributions, they satisfy $g(\zeta)|d\zeta| = p(x)|dx|$, or, since $g(\zeta) = 1$,

$$p(x)|dx| = |d\zeta|. \qquad (3-1)$$

By integrating from a to x, we obtain

$$P(x) = \zeta. \tag{3-2}$$

where ζ is a random number uniformly distributed between zero and one. Given a random number ζ, the value of the distribution we wish to sample is thus

$$x = P^{-1}(\zeta). \tag{3-3}$$

This inversion may be performed either analytically, if simple, or by rejection techniques.[16]

As an example, consider the distribution $f(x) = x^2$ which we wish to uniformly sample over the range $0 < x < 1$. The PDF for this distribution is $p(x) = 3x^2$, while the CDF is $P(x) = x^3$. Thus, from Eq. (3-3), we could sample $f(x)$ directly by $x = \sqrt[3]{\zeta}$. If we wished to avoid calculating the cube root, we could instead use a rejection technique as follows. Choose ζ_1 and ζ_2 uniformly distributed between zero and one. If $\zeta_2 \leq \zeta_1^2$, then let $x = \zeta_1$. If not, then choose another pair of random numbers and test again. Note that the efficiency of a rejection technique is proportional to the area under the integral of the sampled function over the range sampled, which in this case is only 1/3, so out of every three random number pairs selected, only one would meet the rejection criteria. Thus, the use of a direct inversion as opposed to rejection techniques must be evaluated on a case-to-case basis.

3.2 Error Estimates

Since the information provided by the Monte Carlo method is obtained by stochastic processes, it has an associated uncertainty with it. Some estimate of this

associated uncertainty must be provided if the user is to have any confidence in his data.

Let $p(x)$ represent the probability density function (PDF) (i.e., neutron distribution function) we are sampling, where x represents some property of the PDF. Then the expected value of x is equal to the true mean, or

$$\bar{x} = E(x) = \int_a^b x\, p(x)\, dx. \qquad (3-4)$$

while the sample mean, with N trials x_n, is

$$\hat{x} = \frac{1}{N} \sum_{n=1}^{N} x_n. \qquad (3-5)$$

Note that the expectation value of the sample mean is equal to the true mean, so the sample mean is said to be an unbiased estimator of the true mean.[17]

The variance of $p(x)$ is defined as

$$\sigma^2 = E[(x - \bar{x})^2] = \int_a^b (x - \bar{x})^2 p(x)\, dx. \qquad (3-6)$$

where σ is the standard deviation. It can be then be shown that the standard deviation of the sample distribution is related to the standard deviation of the true distribution by[17]

$$\hat{\sigma} = \frac{\sigma}{\sqrt{N}}. \qquad (3-6)$$

The standard deviation of the sample about the true mean thus decreases as the square root of the number of histories run, and is directly proportional to the true

standard deviation. However, neither $p(x)$ or σ is known, so Eq. (3-6) cannot be directly evaluated. Instead, let us define the sample variance S^2 as

$$S^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \hat{x})^2 = \frac{N}{N-1}(\widehat{x^2} - \hat{x}^2),$$
$$\widehat{x^2} = \frac{1}{N}\sum_{n=1}^{N} x_n^2. \qquad (3-7)$$

It can then be shown[17] that the expectation value of the sample variance $E[S^2]$ is equal to the true variance σ^2, so Eq. (3-6) becomes

$$\hat{\sigma} \approx \frac{S}{\sqrt{N}}. \qquad (3-8)$$

The approximation sign is used in Eq. (3-8) because it is the expected value of the sample variance which is equal to the true variance, not the sample variance itself. This approximation holds as long as N is large enough to adequately sample the problem and provide a good estimate of the true variance and mean.

Since the PDF we are sampling is highly unlikely to be a normal distribution, we cannot associate the standard deviation with the typical confidence limits used with a normal distribution. However, if we run K batches of N histories per batch, then the distribution of means $p_N(\hat{x})$ resulting from those batches will assume a normal distribution if N is large enough. This is formally stated by the central limit theorem.[18]

$$p_N(\hat{x}) = \sqrt{N/2\pi}\frac{1}{\sigma}\exp\left[\frac{-N(\hat{x}-\bar{x})^2}{2\sigma^2}\right], \qquad N \to \infty. \qquad (3-9)$$

From Eqs. (3-6) through (3-9), we can then state that the probability of $\bar{x} - \hat{\sigma} \leq \hat{x} \leq \bar{x} + \hat{\sigma}$ is 68.3%, while the probabilities for two and three standard deviations are 95.4% and 99.7%, respectively, where

$$\hat{\sigma}^2 = \frac{1}{K(K-1)} \sum_{k=1}^{K} (\hat{x}_k - \hat{\bar{x}})^2. \qquad (3-10)$$

The unanswered question from the discussion above is how big must N be in order to be big enough? Since the necessary value of N is obviously problem dependent, no fixed value can be given. However, the concept of the figure of merit (FOM) can be used as a guide.[18] From Eq. (3-8), the variance $\hat{\sigma}^2$ behaves as $1/N$, while the CPU time T used in a Monte Carlo calculation is directly proportional to the number of histories run. If we define the FOM as $1/(\hat{\sigma}^2 T)$, then the FOM should remain approximately constant for a given problem, regardless of the number of trials run. However, if the FOM continues to fluctuate widely after the first few trials, then the value of N is probably too low to adequately sample the problem.

The figure of merit also provides a means of comparing the relative efficiency between two different Monte Carlo methods. For a given problem, the method with the largest FOM is considered to be the best overall, since it combines measures of both how accurate and how expensive a method is.

3.3 Advantages and Disadvantages of the Monte Carlo Method

The Monte Carlo method has some significant advantages over deterministic methods. Since the Monte Carlo method is continuous in energy, space, and angle,

it avoids the discretization errors inherent in the deterministic method. Thus, the Monte Carlo method is well-suited for mocking up complicated geomteries, and has no ray effect problems in low scattering regions.

Although the Monte Carlo method has no discretization errors, it is subject to stochastic uncertainties. Since the standard deviation only decreases as the square root of the number of histories run, this can be very expensive to reduce. If only integral information is desired about the problem, such as the total leakage from a cylinder, then oftentimes a low number of histories will suffice for an answer within the desired error bounds. If, however, detailed spatial information about the flux distribution in the cylinder is desired, then the required number of histories for the same error bounds is much greater. This is due to the fact that while every neutron which passes through the cylinder provides information to the integral leakage tally, only those neutrons which pass through a given spatial subsection of the cylinder will provide information about the flux in that area. In contrast, deterministic methods provide information for each and every cell in the problem, and although deterministic computational times do increase as the number of cells is increased, it is usually much less significant than the computational increases required with Monte Carlo for similarly detailed spatial information.

Also, although the Monte Carlo method is clearly better suited than deterministic methods for problems with low scattering regions, the converse is true in problems containing highly scattering regions. Consider again a cylinder composed of graphite with a point source in the center for which we wish to determine the integral leakage. If the cylinder is optically thick, than a neutron will have to undergo

many scattering collisions before it escapes from the surface, and the computational time per history will be large.

Thus, neither deterministic methods nor probabilistic methods provide a panacea for all possible computational problems involved in solving the transport equation. It is then natural to ask if the two methods could be combined to solve problems that neither method alone is well suited for. Our answer to that question will be the subject of the remainder of this dissertation.

CHAPTER 4.
THEORY OF THE RESPONSE MATRIX HYBRID METHOD

Consider a Monte Carlo region embedded in a S_N region (Fig. 4-1), where Ψ^{in} and Ψ^{out} represent the boundary fluxes entering and leaving the Monte Carlo region, and $\mathbf{n}$ is the outward directed normal. The elements of Ψ^{out} are the incoming angular fluxes to the S_N region, where each element corresponds to a unique combination of spatial mesh cell, energy group, and quadrature direction Ω_n, with $\Omega_n \cdot \mathbf{n} > 0$. The interface angular fluxes for $\Omega_n \cdot \mathbf{n} < 0$ are the elements of Ψ^{in}. The outgoing flux from the Monte Carlo region is related to the incoming flux from the S_N region by

$$\Psi^{out} = \underline{\mathbf{R}}\,\Psi^{in} + \mathbf{S}^{out}, \qquad (4-1)$$

where $\mathbf{S}^{out}$ is the exiting flux from the Monte Carlo region under vacuum boundary conditions ($\Psi^{in} = 0$). The element $r_{kk'}$ of the response matrix $\underline{\mathbf{R}}$ represents the angular flux leaving the Monte Carlo region in S_N state k due to a unit incident angular flux in S_N state k'. Because Ψ^{in} is generally not known, Eq. (4-1) is solved iteratively by

$$\Psi^{out(r+1)} = \underline{\mathbf{R}}\,\Psi^{in(r)} + \mathbf{S}^{out}, \qquad (4-2)$$

where, for example, we can set

$$\Psi^{out(1)} = \mathbf{S}^{out}, \qquad (4-3)$$

$\Psi^{in(r)}$ is obtained using an S_N solver with the prescribed boundary flux $\Psi^{out(r)}$, and the response matrix $\underline{\mathbf{R}}$ has been precalculated by the Monte Carlo method. The Monte Carlo method is also used to sample the fixed source in the Monte Carlo region, determining $\mathbf{S}^{out}$. For subcritical problems, Eq. (4-2) is expected to be unconditionally convergent on a physical basis. Note also that the above equations may be extended to partially embedded Monte Carlo regions by replacing the appropriate elements of Ψ^{in} with the specified S_N boundary conditions.

As explained in Ref. 10, it is more advantageous to perform a separate S_N calculation with each iteration r, as opposed to precalculating a response matrix for the S_N region. In order to precalculate a S_N response matrix, a separate S_N calculation would be required for each state in the S_N/Monte Carlo interface, while experience has shown that the number of iterations required for Eq. (4-2) to converge is much less than the total number of states in the interface. Since the time required per S_N calculation is approximately the same, regardless of whether the calculation is done for a single state or all states in the interface, it is clearly more efficient to perform separate S_N calculations at every iteration r. This also reduces storage requirements.

Conversely, the exact opposite situation occurs in the case of the Monte Carlo response matrix. There, the calculation time required for the response matrix is approximately equivalent to the calculation time of the entire boundary value problem at an iteration r. Thus, it is more advantageous to precalculate and store the Monte Carlo response matrix.

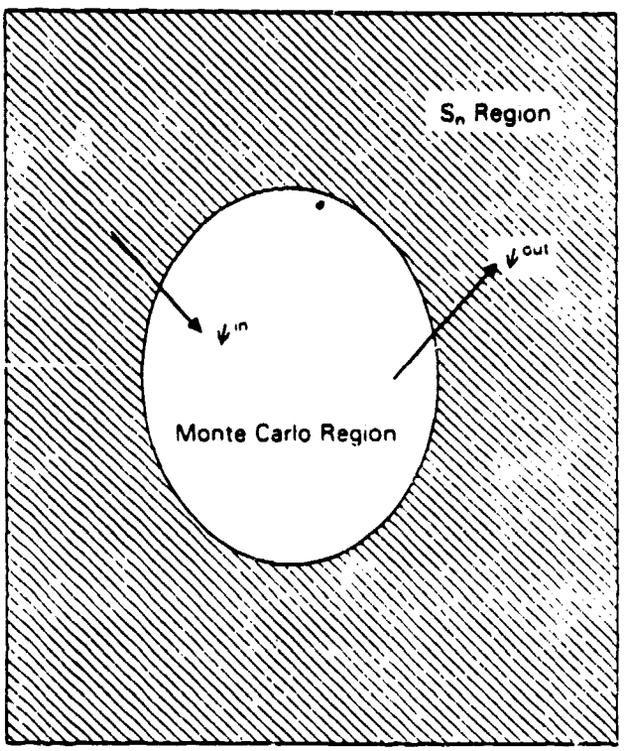

Figure 4-1 Boundary Fluxes at an S_N/Monte Carlo Interface

A straightforward evaluation of Eq. (4-2) would require a response matrix $\underline{\mathbf{R}}$ of size $K_G \times K_G$ with K_G-dimensional vectors $\boldsymbol{\Psi}^{out}$, $\boldsymbol{\Psi}^{in}$, and $\mathbf{S}^{out}$, where

$$K_G = K \times G, \tag{4-4}$$

and

$$K = IL \times M/2, \tag{4-5}$$

with G defined as the number of energy groups, IL as the number of spatial mesh cells along the S_N/Monte Carlo interface, and M as the total number of directions in the S_N quadrature set. The factor of two appears in Eq. (4-5) because, in two dimensions, we require only two out of four directional quadrants since we are concerned only with outgoing directions.

For typical S_N mesh grids, the memory requirements of Eq. (4-2) are too large for practical implementation. As an example, with a total of 60 spatial cells along the S_N/Monte Carlo interface, an S_6 quadrature set ($M = 24$), and 30 energy groups, over 465 million words of memory would be needed for storage of the response matrix $\underline{\mathbf{R}}$, exceeding the capacities of most of today's machines. However, significant reductions in storage requirements are possible. Let the K-dimensional subvectors $\boldsymbol{\Psi}_g^{out}$, $\boldsymbol{\Psi}_g^{in}$, and $\mathbf{S}_g^{out}$ denote the group g portions of $\boldsymbol{\Psi}^{out}$, $\boldsymbol{\Psi}^{in}$, and $\mathbf{S}^{out}$. Equation (4-2) may then be replaced by the equations

$$\boldsymbol{\Psi}_g^{out(r+1)} = \underline{\mathbf{R}}_{gg}\, \boldsymbol{\Psi}_g^{in(r)} + \mathbf{Q}_g^{out}, \qquad g = 1, \ldots, G, \tag{4-6}$$

and

$$\mathbf{Q}_g^{out} = \sum_{g' \neq g} \underline{\mathbf{R}}_{gg'}\, \boldsymbol{\Psi}_{g'}^{in} + \mathbf{S}_g^{out}. \tag{4-7}$$

where $\underline{\mathbf{R}}_{gg'}$ is the $K \times K$ submatrix of $\underline{\mathbf{R}}$ representing transfer from group g' to group g.

For problems without upscatter or fission, Eq. (4-7) reduces to

$$\mathbf{Q}_g^{out} = \sum_{g'<g} \underline{\mathbf{R}}_{gg'} \, \mathbf{\Psi}_{g'}^{in} + \mathbf{S}_g^{out}. \qquad (4-8)$$

Equations (4-6) and (4-8) can be implemented as follows:

1. Using Monte Carlo, calculate the $\mathbf{S}_g^{out}$ by sampling the fixed source in the Monte Carlo region.

2. For $g = 1, \ldots, G$,

   a. determine $\underline{\mathbf{R}}_{gg}$ using Monte Carlo,

   b. using $\underline{\mathbf{R}}_{gg}$ and $\mathbf{Q}_g^{out}$, determine $\mathbf{\Psi}_g^{out}$ and $\mathbf{\Psi}_g^{in}$ from Eq. (4-6), with one S_N calculation per iteration step r,

   c. calculate directly (without first determining the $\underline{\mathbf{R}}_{gg'}$) the contributions to $\mathbf{Q}_{g'}^{out}$, $g' > g$, by sampling $\mathbf{\Psi}_g^{in}$,

   d. discard $\underline{\mathbf{R}}_{gg}$ to make room for $\underline{\mathbf{R}}_{g+1,g+1}$.

With this algorithm, only the submatrix $\underline{\mathbf{R}}_{gg}$ is used for calculations at any given time, reducing the storage requirements by a factor of G^2. However, this method also introduces a penalty in efficiency, since it is more efficient to calculate the entire response matrix at one time, versus calculating the $\underline{\mathbf{R}}_{gg}$ and $\mathbf{Q}_g^{out}$ separately. In a calculation of $\underline{\mathbf{R}}$, all particles eventually score by leaving the Monte

Carlo region. In calculating $\underline{\mathbf{R}}_{gg}$, however, particles that down scatter are not scored. Instead, the downscatter contribution to $\mathbf{Q}_{g'}^{out}$ is determined in a separate Monte Carlo calculation by sampling Ψ_g^{in}. In this calculation, particles escaping from the Monte Carlo region in group g are not scored. The inefficiency in the calculation of $\underline{\mathbf{R}}_{gg}$ is easily removed by replacing explicit downscatter with an appropriate reduction in particle weight (reference Section 6.4), while the inefficiency in the computation of the $\mathbf{Q}_{g'}^{out}$ can be minimized by using forced collisions (reference Chapter 9). Note also that the S_N calculation in step 2.b is itself an iterative process, involving inner iterations as per Eq. (2-22).

With upscatter and/or fission, additional iterations over energy are required (i.e., an outer iteration). Accordingly, we include an energy iteration index p on $\Psi^{in(p,r)}$, $\Psi^{out(p,r)}$, and $\mathbf{Q}^{out(p)}$, and iterate over step 2 above until convergence occurs. In effect, we are determining Eqs. (4-6) and (4-7) by the iteration process

$$\Psi_g^{out(p+1,r+1)} = \underline{\mathbf{R}}_{gg}\, \Psi_g^{in(p+1,r)} + \mathbf{Q}_g^{out(p)}, \qquad (4-9)$$

and

$$\mathbf{Q}_g^{out(p)} = \sum_{g'<g} \underline{\mathbf{R}}_{gg'}\, \Psi_{g'}^{in(p+1,\infty)} + \sum_{g'>g} \underline{\mathbf{R}}_{gg'}\, \Psi_{g'}^{in(p,\infty)} + \mathbf{S}_g^{out}, \qquad (4-10)$$

where $\Psi_g^{out(p,\infty)}$ and $\Psi_g^{in,(p+1,\infty)}$ represent the converged fluxes from Eq. (4-9). Since the $\underline{\mathbf{R}}_{gg}$ are independent of p, step 2.d is eliminated so that the response matrices can be saved and reused at each value of p. The storage requirements for the $\underline{\mathbf{R}}_{gg}$, $g = 1, 2, \ldots, G$ of GK^2 words is still a factor of G less than those needed for the full matrix $\underline{\mathbf{R}}$. Since only one submatrix is needed at a time, the rest are stored on a mass storage device, reducing the core memory requirements to K^2

words, the same as required for the pure downscatter case. Again, we emphasize that the $\underline{\mathbf{R}}_{gg'}$ are not actually calculated. Instead, the contributions to $\mathbf{Q}_g^{out(p)}$ are determined by sampling the $\Psi_g^{in(p,\infty)}$ directly.

In calculating the $\underline{\mathbf{R}}_{gg}$ with upscatter and/or fission, we again replace explicit outscatter (up and down) and fission with appropriate modifications in particle weight to improve efficiency. With pure downscatter, however, once a particle scattered out of a group there was no possibility of it scattering back into that group and scoring. This is not the case with upscatter and fission. Thus, we are no longer calculating the true group-to-group response matrix $\underline{\mathbf{R}}_{gg}$, but a group-within-group response matrix which we denote by $\tilde{\underline{\mathbf{R}}}_{gg}$. We correct for this when sampling the $\Psi_g^{in(p,\infty)}$ for contributions to $\mathbf{Q}_{g'}^{out(p)}$, $g' \neq g$, by scoring in $\mathbf{Q}_g^{out(p)}$ particles that exit the Monte Carlo region in group g after having entered, at some point, a group g'. Thus, Eqs. (4-9) and (4-10) are replaced with

$$\Psi_g^{out(p+1,r+1)} = \tilde{\underline{\mathbf{R}}}_{gg}\, \Psi_g^{in(p+1,r)} + \mathbf{Q}_g^{out(p)}. \tag{4-11}$$

and

$$\begin{aligned} \mathbf{Q}_g^{out(p)} = &\sum_{g'<g} \underline{\mathbf{R}}_{gg'}\, \Psi_{g'}^{in(p+1,\infty)} + \sum_{g'>g} \underline{\mathbf{R}}_{gg'}\, \Psi_{g'}^{in(p,\infty)} + \\ &(\underline{\mathbf{R}}_{gg} - \tilde{\underline{\mathbf{R}}}_{gg})\Psi_g^{in(p,\infty)} + \mathbf{S}_g^{out}. \end{aligned} \tag{4-12}$$

Note that the outer iteration process for the Monte Carlo region differs from the outer iteration process for the S_N region (reference Section 2.2) in that both the updated fission source and scatter source, where available, are used in Eq. (4-12), whereas S_N uses only the updated scattering source. However, S_N also uses acceleration techniques[13] to improve convergence of its outer iteration, whereas the Monte Carlo does not. The effects of this will be examined in Section 5.7 and Chapter 11.

CHAPTER 5.
IMPLEMENTATION OF THE HYBRID METHOD

Our primary goal in this dissertation is solving problems with Monte Carlo regions either completely or partially embedded in S_N regions. Thus, it is natural to implement the hybrid method by modifying the internal structure of an S_N code to accept information from Monte Carlo routines. Since the S_N code TWODANT,[12] developed by the Los Alamos National Laboratory, is a reliable, state of the art discrete ordinates code capable of solving varying and complex transport problems for neutral particles in two dimensions, it was selected as the basis for the hybrid method. While the entire TWODANT source code is somewhat large (approximately 50,000 lines), the portion of the solver requiring changes is less than 10,000 lines, and is written in relatively well-structured FORTRAN amenable to modification. Thus, the hybrid method is implemented by modifying the discrete ordinates code, where necessary, and adding special purpose routines to perform the Monte Carlo functions and link them to the S_N. As explained in Chapter 3, it was decided to develop our own unique Monte Carlo subroutines, both because of special problems posed by the hybrid method, and for ease of debugging and efficiency.

This chapter describes the physical interface between the S_N and Monte Carlo regions, the process of transferring information between the two, the computational structure of the hybrid method's implementation in TWODANT, and

necessary modifications to the S_N method, while Chapter 6 will describe the details of the Monte Carlo method.

5.1 Physical Description of the Monte Carlo Region

While the theory developed in the preceding chapter was for an arbitrarily shaped Monte Carlo region, including possibly multiple Monte Carlo regions in the same problem, we consider only a single, rectangularly shaped region here, either partially or completely embedded in an S_N region as shown at Fig. 5–1. This enables considerable simplification in the details of tracking particles through the Monte Carlo region, and in the development of an interface between the Monte Carlo and the S_N, with little loss of generality.

In TWODANT, the problem geometry is broken up into rectangular coarse mesh cells, each of which is in turn composed of a varying number of fine mesh cells.[12] The coarse mesh cells are homogeneous in composition, and thus represent a convenient method of specifying the location of differing materials for heterogeneous problems. The actual discrete ordinates calculation is performed upon the fine mesh cell structure, however, since smaller spatial dimensions than those typically provided by the coarse mesh structure are required for accurate results in the S_N method.

Since there is no spatial discretization process involved in Monte Carlo calculations, and since the speed of a Monte Carlo calculation decreases as the number of internal boundaries increases, we choose to base the physical description of the

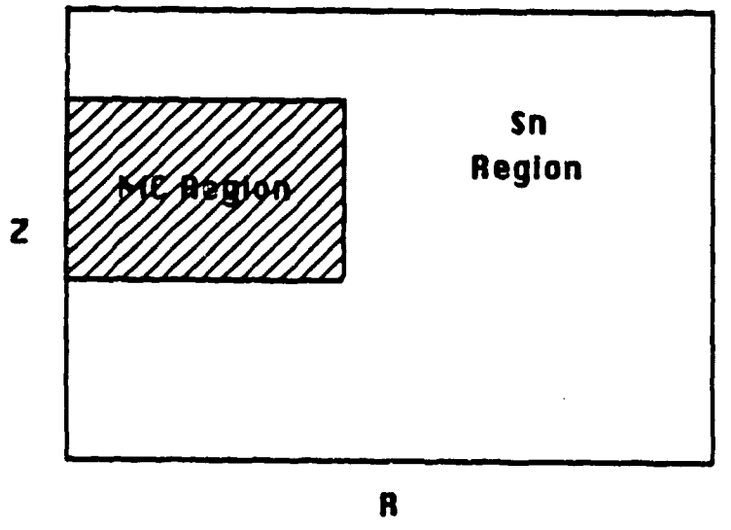

Figure 5-1 Monte Carlo Region as Implemented in the Hybrid Method

Monte Carlo region upon the coarse mesh cell structure provided by the S_N. Thus, all necessary cell boundaries and compositions are already available from the S_N, without the extensive user input and calculations commonly required by pure Monte Carlo codes. Instead, the desired Monte Carlo region for the hybrid method is described by specifying just four boundaries located along S_N mesh cell edges - top, bottom, left, and right.

5.2 The Boundary Layers

Typically, the Monte Carlo/S_N boundaries will be located along material discontinuities in the problem. However, the angular flux at such a boundary is usually highly anisotropic, requiring a large number of quadrature angles for an adequate description. From Eq. (4-5), we see that the size of the response matrix is proportional to the square of the number of quadrature directions, so that memory requirements would increase dramatically if the actual S_N/Monte Carlo interface were located along a boundary between optically thin and thick materials.[10]

Instead, we add an additional "boundary layer" by extending the Monte Carlo region from the previously specified boundary into the S_N region. This then presumably places the S_N/Monte Carlo interface in a highly scattering region where the flux is more nearly isotropic, enabling a lower quadrature order to be used. The distance required is determined by calculating the number of fine mesh cells needed to meet or exceed a user specified number of mean free paths in distance for the material along the appropriate boundary. Previous work[10] has shown that a one mean free path thick boundary is usually sufficient. It is desirable to keep the

boundary layer as small as possible, since it will usually consist of highly scattering materials for which Monte Carlo calculations are inefficient.

Note that since cross sections are energy dependent, the mean free path in a material will change with the energy group. Thus, the size of the boundary layer is also group dependent, often resulting in an "inverted wedding cake" type of structure as shown at Fig. 5-2. If the mean free path for a given energy is large enough, the Monte Carlo region may constitute the entire problem geometry for some groups, as is also shown in the sample figure. We also provide the capability to entirely eliminate the Monte Carlo region altogether for energy groups where S_N is sufficient throughout the entire problem geometry.

The type of structure shown at Fig. 5-2 provides further interface possibilities between the Monte Carlo and the S_N, in addition to the boundary type fluxes described in Chapter 4, since upscattering, downscattering, or fission may now result in particles transferring from the S_N region into the Monte Carlo region, and vice versa. Since this transference occurs over an entire mesh cell, it results in volumetric sources which are handled as specified in Section 5.4.

5.3 Interfacing Phase Space Coordinates

One of the most fundamental differences between the Monte Carlo method and the discrete ordinates method is in their treatment of the variables in phase space. While Monte Carlo treats these variables as continuous, S_N discretizes them. T us, when transferring particles from the Monte Carlo region to the S_N region,

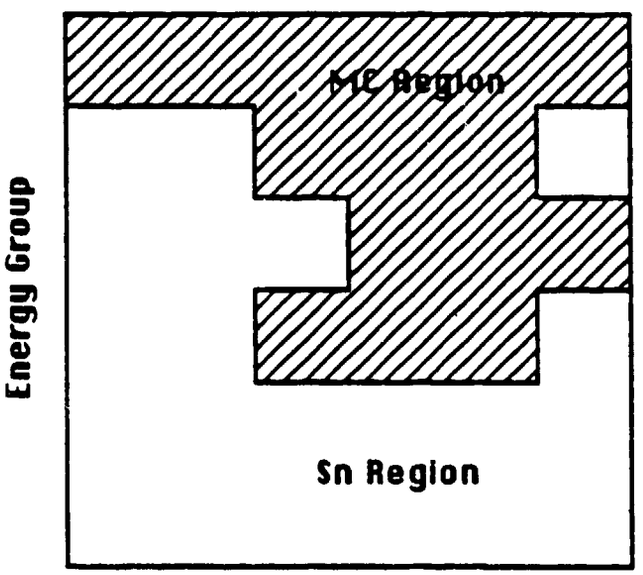

Figure 5-2 Boundary Layer Thickness as a Function of Energy Group

we must discretize the phase space coordinates, and when going from the S_N region to the Monte Carlo region we must assign continuous values to the given discrete variables used by the S_N.

Since the hybrid method employs multigroup cross sections (see Chapter 8), the energy variable has already been discretized for both the Monte Carlo and S_N regions, so no conversion is necessary. Conversion of spatial coordinates is straightforward - when passing from the Monte Carlo region into the S_N region, a particle is scored into the appropriate fine mesh cell, while particles entering the Monte Carlo region from the S_N region are assumed to be uniformly distributed over the area (or volume, for volumetric sources) of the fine mesh cell. The details of the spatial conversion process are described in the succeeding chapter.

The conversion of angular coordinates is not as clear-cut, however. While discrete ordinates represents the unit sphere of directions with a set of discrete directions, each of which is assigned an angular weight proportional to the area subtended on the unit sphere by that direction, the shape of the area associated with a discrete direction is not specified. Thus, there is no unique method of determining angular bins $\Delta\mathbf{\Omega}_n$ for assigning a set of continuous Monte Carlo directions to the discrete S_N direction $\mathbf{\Omega}_n$.

The system we have chosen to use is depicted at Fig. 5–3. The dots represent the intersections of the $\mathbf{\Omega}_n$ with the unit sphere for an S_6 quadrature set. The $\Delta\mathbf{\Omega}_n$ are defined by lines of constant azimuthal angle $\mathbf{\Omega}_\varphi$ and constant $\mathbf{\Omega}_\theta$. The lines of constant $\mathbf{\Omega}_\theta$ form levels of constant width $\Delta\eta$, where $\Delta\eta$ is equal to the sum of the weights of the discrete directions along that level divided by the total weight

of all discrete directions in the quadrant. Each level is in turn divided into bins of $\Delta\phi$, where the $\Delta\phi$ for each bin is proportional to the weight of its assigned discrete ordinate divided by the sum of the weights for all discrete ordinates along that level. The bin structure for an S_6 quadrature set, along with the corresponding discrete ordinates, is shown at Fig. 5–4. Note that the quadrature set actually used in the hybrid method is the level symmetric set that has been built-in to TWODANT.[13]

With this bin arrangement, particles are easily scored when crossing into the S_N region. The discrete directional level the particle enters is simply determined by comparing its direction cosine along $\mathbf{\Omega}_\theta$, η, with the values of the levels as determined above, while the bin that it enters is determined by comparing the projection of the direction cosine along $\mathbf{\Omega}_\phi$, $\mu/\sqrt{1-\eta^2}$, with the cosine of the azimuthal angle of the bin edge. Particles entering the Monte Carlo region from the S_N region are assumed to be uniformly distributed in η and ϕ across the $\Delta\mathbf{\Omega}_n$ corresponding to the discrete direction $\mathbf{\Omega}_n$.

While the bin arrangement described above conserves the unit area of the sphere, i.e., it preserves the zeroth moment of the flux, there is no guarantee that higher orders will be preserved. That is, quadrature sets are sometimes chosen so as to exactly integrate the highest order Legendre polynomial possible,[14] so that if an angular flux consists of only a small number of Legendre moments (in one dimension), those moments will be preserved. However, if we uniformly distribute the particles resulting from that angular flux over the finite bin areas associated with the discrete directions, and then sum the moments of the individual particles, we no longer will preserve higher order moments.

The bin structure at Fig. 5-3 does present an advantage, however, in that it preserves an important physical property of cylindrical geometry. As previously stated, when a particle streams in $R-Z$ geometry its direction cosine along Ω_φ will change, as represented by the arrows in Fig. 5-3, while the direction cosine along Ω_θ will remain constant. The bin configuration of Fig. 5-3 preserves this property since streaming particles can only flow into bins whose quadrature directions have the same Ω_θ. This is not the case for the alternative bin structure shown at Fig. 5-5.

There is an additional minor advantage to the bin structure of Fig. 5-3. To represent an isotropic point source, particles should be assigned to bins in proportion to the bin weights, except when the source is located at $r = 0$ for cylindrical geometry. The uncollided particles emanating from a point source at $r = 0$ travel exclusively in the $R-Z$ plane, that is, $\Omega_\varphi = 0$. As $r \rightarrow 0$, Ω_φ becomes meaningless and the distinction between discrete directions on the same Z-level (Ω_θ value) disappears. For this case it would be best to assign all of the particles to the bins along the $R-Z$ plane in proportion to the Z-level weights (the sum of the weights for all bins along the same Z-level). This is true for the bin structure of Fig. 5-3, but not the the alternative bin structure of Fig. 5-5. The reason becomes evident upon examining the two figures. Assume that bin n is the one that intersects the $R-Z$ plane, and let $\Omega_\theta^{n+1/2}$ and $\Omega_\theta^{n-1/2}$ represent the largest and smallest Ω_θ values of this intersection. With the arrangement of Fig. 5-3, $\Omega_\theta^{n+1/2} - \Omega_\theta^{n-1/2}$ equals the Z-level weight. Thus, for particles uniformly distributed in Ω_θ in the $R-Z$ plane (as would result from an isotropic point source at $r = 0$), the number

between $\Omega_\theta^{n+1/2}$ and $\Omega_\theta^{n-1/2}$ would be proportional to the Z-level weight. This would not be true for the arrangement of Fig. 5-5.

5.4 Particle Transference

As each particle crosses from the Monte Carlo region into the S_N region, its phase space coordinates are discretized as described above, and it is then placed into the S_N state corresponding to the resulting unique combination of spatial mesh cell, energy group, and quadrature direction. Particles physically streaming across an S_N/Monte Carlo interface are represented as a boundary flux source into the S_N region, while particles which enter the S_N region through a change in energy group due to fission or scattering are represented as a volumetric source per Section 5.2 above.

At the conclusion of a Monte Carlo calculation, the quantity actually residing in an S_N state is the total weight of all particles which have entered that state during the calculation. For boundary flux sources, this number is converted into an S_N boundary flux by dividing by the quantity $\Delta\Omega_n \, \Delta A_{ij} \, N_H$, where $\Delta\Omega_n$ is the angular weight associated with the discrete direction Ω_n, ΔA_{ij} is the appropriate area of the mesh cell face entered (Δx_i or Δy_j for Cartesian geometry, $\pi(r_{i+1/2}^2 - r_{i-1/2}^2)$ or $2\pi r_{i\pm1/2} \, \Delta z_j$ for cylindrical geometry), and N_H is a normalization factor dependent upon the number of histories run.

Volumetric sources are treated somewhat differently, since, to conserve storage, S_N codes usually store the spherical harmonic moments of the source, not the

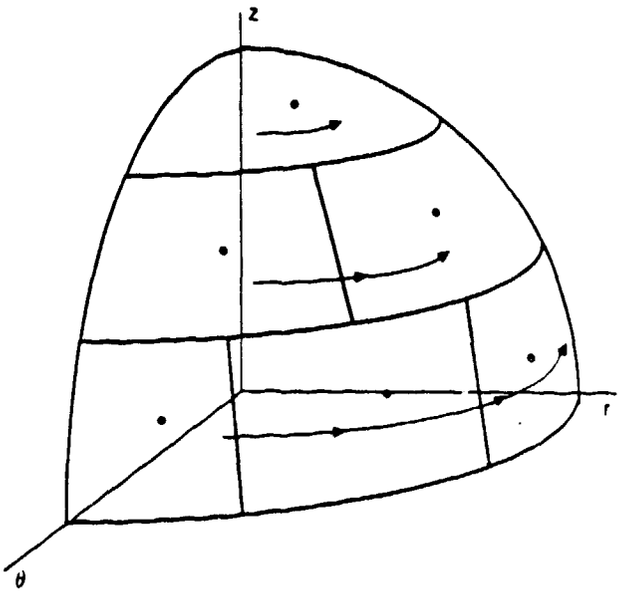

Figure 5-3 The S_N Angular Bin Arrangement for $N = 6$

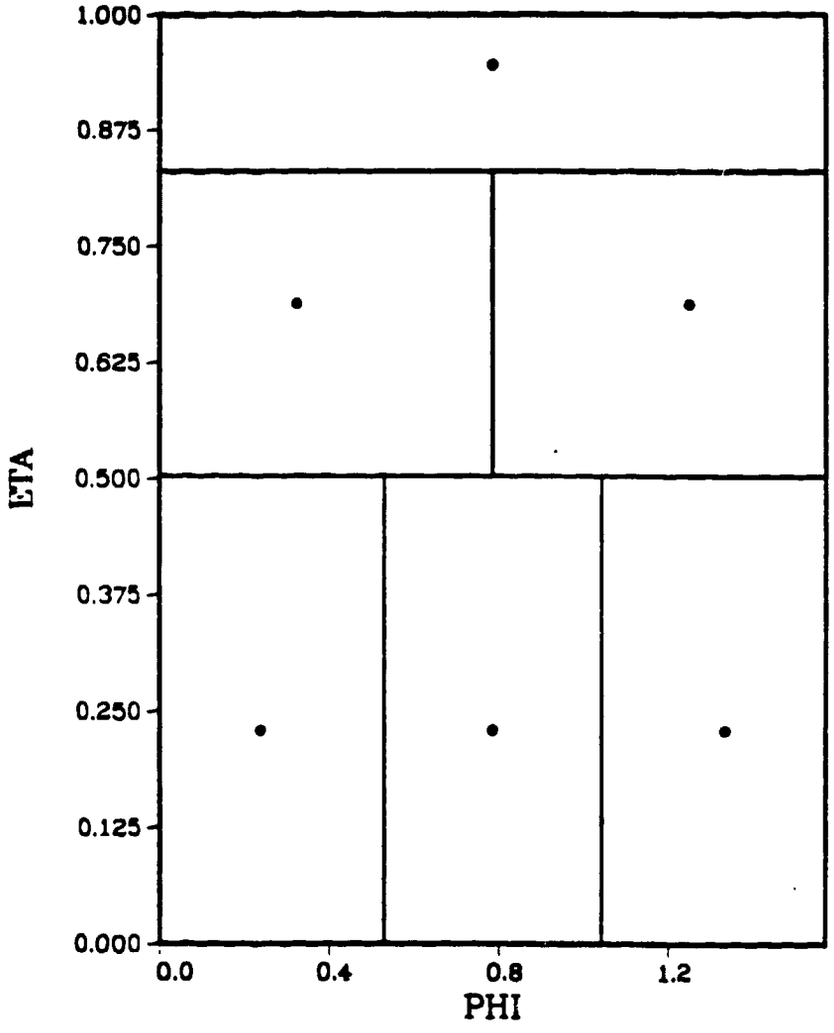

Figure 5–4 The Bin Structure for $N = 6$

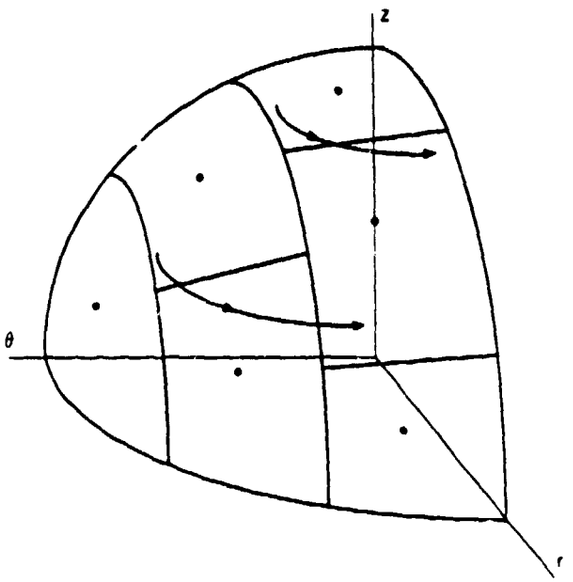

Figure 5-5 Alternative S_N Angular Bin Arrangement for $N = 6$

angular source itself.[13] Thus, if W_{ijng} represents the total weight of particles transferred to cell (i,j) with discrete direction Ω_n in energy group g, the corresponding S_N moment $\tilde{Q}_{ijlmg}$ is found from

$$\tilde{Q}_{ijlmg} = \sum_{n=1}^{N} Y^{*}_{lmn} W_{ijng}/(\Delta V_{ij} N_H). \qquad (5-1)$$

where Y^{*}_{lmn} is the complex conjugate of the lth, mth spherical harmonic evaluated at discrete direction Ω_n, and ΔV_{ij} is the volume of the cell (i,j). Note that this conversion is not performed until the Monte Carlo calculation has been completed, so sufficient storage must be allocated to allow the the Monte Carlo to score particles directly in the angular source states. However, since the response matrix is not in use when volumetric source calculations are active, its storage area may be used for the angular source states, so no additional memory is required.

5.5 Computational Structure of the Hybrid Method's Implementation

Now that we have examined in some detail the physical interface between the Monte Carlo and the S_N regions, we turn to the computational interface. That is, we now describe how the hybrid theory of Chapter 4 is meshed with the multigroup discrete ordinates method of Chapter 2.

Figure 5-6 depicts the computational flow of the unmodified discrete ordinates code TWODANT. Subroutine TIGF20 is the overall driver for the solver module, which begins by calling a series of input processing and memory allocation routines represented by the block TINP. Next, subroutine TGND25 calculates the

S_N grid structure, sets up the S_N cross sections, and performs other initialization functions. The outer iteration loop [reference Eq. (2-5)] begins with subroutine TRANSO, where, for energy groups $g = 1, \ldots, G$, TOUTER calculates the total source to the group g from fixed sources, downscatter, upscatter, and fission [i.e., the $Q_{ng}(\mathbf{r})$ in Eq. (2-22)], while SINNER performs the inner iterations. Subroutines DIFFO and DOUTER complete an outer iteration loop by updating the fission source. Additionally, DOUTER accelerates convergence of the fission source through use of a Chebyshev polynomial-based method.[13] (We note that convergence of both the inner and outer iterations are also accelerated through the use of diffusion synthetic acceleration,[13] which will be discussed in Chapter 10).

Figure 5-7 presents the computational flow as modified to incorporate the hybrid method. The additional input required for the hybrid method is processed in TINP, while TGND25 calls routines MCXS and SETUP, which define the Monte Carlo cross sections and other arrays required for unique Monte Carlo functions. Monte Carlo calculations actually begin with SRCMC, which samples the fixed source to determine the $\mathbf{S}_g^{out}$ of Eq. (4-12), while SMOM computes the volumetric sources in the S_N region resulting from SRCMC, if any, and stores the moments in the S_N inhomogeneous source array. Next, MCRM calculates the within-group response matrix $\underline{\tilde{\mathbf{R}}}_{gg}$ [reference Eq. (4-11)] for groups $g = 1, \ldots, G$, where required. Note that the calculation of the response matrix is omitted for a group g if the Monte Carlo region or the S_N region comprises the entire problem geometry. The response matrices are stored on disk for recall during the iteration process.

The basic structure of the outer loop is unaffected by the hybrid method, except for modifications to allow volumetric sources to be passed from the Monte

Carlo region to the S_N region by SMOM. TOUTER gets the volumetric source to group g for both Monte Carlo and S_N regions from S_N regions in groups $g' \neq g$ as before, while the volumetric source from any Monte Carlo regions in groups $g' \neq g$ to the S_N region for group g is obtained from the inhomogeneous S_N source array, where it has been stored by SMOM. Instead of calling SINNER to perform the S_N inner iterations, however, TOUTER calls LINK, which peforms the coupled S_N/Monte Carlo iterations of Eq. (4-11).

LINK begins by calling SNTOMC, which determines the contribution of the group g volumetric sources located in the Monte Carlo region, if any, to the $\mathbf{Q}_{g'}^{out}$, $g' = 1, \ldots, G$. SMOM is then called to compute and store any further contributions to volumetric sources in the S_N region which arose during the Monte Carlo calculation of SNTOMC. Next, the iteration process of Eq. (4-11) starts by getting the response matrix $\underline{\tilde{\mathbf{R}}}_{gg}$ from disk. The initial outgoing boundary flux used by LINK is

$$\begin{aligned} \Psi_g^{out(p+1,1)} &= \underline{\tilde{\mathbf{R}}}_{gg}\, \Psi_g^{in(p,\infty)} + \mathbf{Q}_g^{out(p)}, \qquad p > 0, \\ \Psi_g^{out(1,1)} &= \mathbf{Q}_g^{out(1)}, \qquad p = 0. \end{aligned} \tag{5-2}$$

where $\Psi_g^{in(p,\infty)}$ is the converged incoming boundary flux from the previous outer iteration. SINNER is then called to perform the S_N inner iterations, returning $\Psi_g^{in(p+1,1)}$, and $\Psi_g^{out(p+1,2)}$ is calcu'ated from Eq. (4-11). If the maximum relative error between $\Psi_g^{out(p+1,1)}$ and $\Psi_g^{out(p+1,2)}$ is less than the S_N error criteria, than convergence is considered to have been achieved. Otherwise, the iteration procedure continues until convergence is reached, or a user-input maximum number of iterations has been reached.

If the iteration process of Eq. (4-11) does not reach convergence, an extrapolation procedure developed by Filippone[10,19] is used to improve the convergence of the outgoing boundary flux. This procedure, referred to as the method of residual expansion functions, essentially assumes that, after a sufficient number of iterations, the differences between succesive iterations (residuals) are composed of only a limited number of eigenvectors (i.e., shape convergence). Although these eigenvectors (and corresponding eigenvalues) are unknown, the residuals themselves can be used as a basis, and an expression for coefficients of these residuals which minimize the remaining error in the iteration process can be derived.[20] Typically, five to eight iterations will usually provide enough residuals to allow the method of residual expansion functions to extrapolate a solution which is within the S_N error criteria. The accuracy of the method is checked by performing another set of S_N inner iterations, based on the extrapolated solution, and measuring the largest relative diference in the extrapolated boundary flux and the resulting boundary flux as calculated from Eq. (4-11).

The method of residual expansion functions will, however, occasionally fail to project a more accurate solution. This occurs when use of the diffusion synthetic accelerator, which accelerates convergence of the S_N inner iterations, results in rapid convergence of the S_N inner iterations, thus not allowing higher order harmonics in the boundary flux to die out. The presence of these harmonics violates the premises upon which the residual expansion function method was derived, and so an accurate extrapolation is not possible. The situation is easily remedied, however, either by tightening the S_N error criteria, which forces more S_N inner iterations, or allowing enough iterations on Eq. (4-11) for convergence to be met by the iteration process

itself. This is economical in this situation, since the diffusion synthetic accelerator itself enables the S_N iterations to be performed very efficiently.

Once the converged boundary fluxes have been calculated, subroutine DWN-SRC is called to sample the $\Psi_g^{in(p,\infty)}$. As described in Chapter 4, we do not evaluate Eq. (4-12) directly, since storage limitations preclude the calculation of the response matrices $\underline{\mathbf{R}}_{gg'}$. Instead, we determine the contribution of $\Psi_g^{in(p,\infty)}$ to the $\mathbf{Q}_{g'}^{out(p)}$, $g' > g$, and the $\mathbf{Q}_{g'}^{out(p+1)}$, $g' \leq g$, by sampling the incoming boundary flux into the Monte Carlo region for group g and scoring particles when they cross back into the S_N region in group g'. Particles that reenter the S_N region while in group g are separated into two classes - those that have entered a group $g' \neq g$ at some point during their history, and those that have remained entirely within group g. The first class of particles, which represent the third term on the left-hand side of Eq. (4-12), is scored in $\mathbf{Q}_g^{out(p+1)}$, while the second class is used as a comparison to the computed $\Psi_g^{out(p,\infty)}$ as described in Chapter 9. Finally, SMOM is called once again to update any contributions by the Monte Carlo calculation to volumetric sources in the S_N region.

If no response matrix calculation is required for group g, then LINK omits the iteration process of Eq. (4-11). Instead, if the problem geometry is entirely represented by the S_N, then LINK merely calls SINNER once to get the S_N fluxes. If the problem geometry is entirely in the Monte Carlo region, then no calculations (beyond SNTOMC) are required in LINK at all.

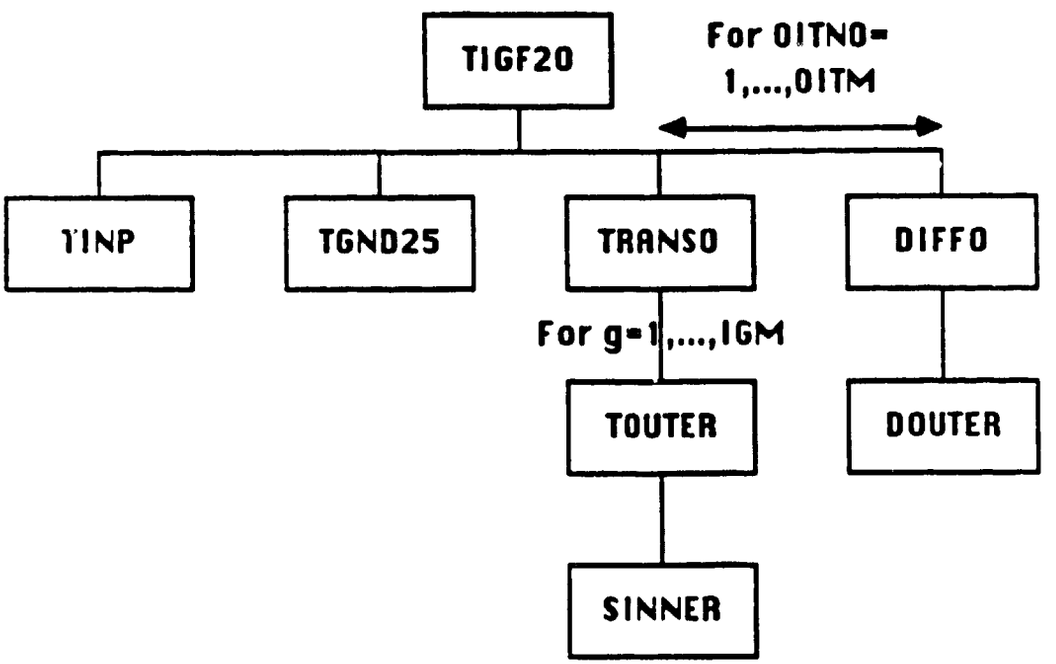

Figure 5-6 Computational Flow of the Discrete Ordinates Code

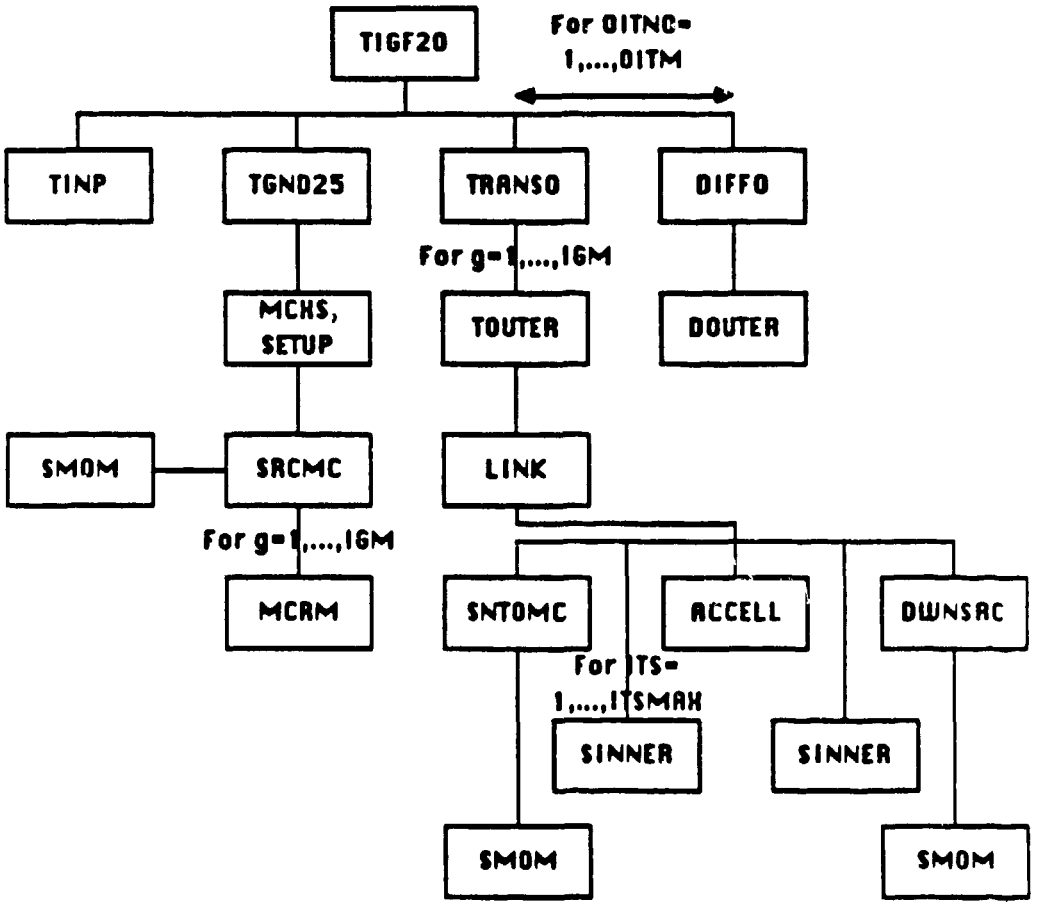

Figure 5-7 Computational Flow of the Hybrid Method

5.6 Discrete Ordinates Modifications

As pictured above, the principal requirements for implementation of the hybrid method are the addition of Monte Carlo routines, while the modifications necessary to the discrete ordinates code itself are minimal. The primary modications involved are to the logic of the S_N inner iterations, which are performed in SINNER and other associated subroutines, to allow the inclusion of interior boundary sources.

An S_N inner iteration entails sweeping through the mesh cells, from exterior boundary to exterior boundary, for a given sweeping direction. When sweeping from the Monte Carlo region into the S_N region, the S_N-calculated values for these surfaces are replaced with the $\Psi_g^{out(p+1,r)}$ from Eq. (4-11). Sweeps from the S_N region into the Monte Carlo region are performed normally, except that the converged surface fluxes are used as the $\Psi_g^{in(p+1,r)}$ in Eq. (4-11). As implemented by Alcouffe,[10] the vectorization of the sweeping algorithm is unaffected, so that the efficiency of a standard S_N calculation is retained. The introduction of these interior boundary sources does reflect a discontinuity in the solution of the transport equation, however, which in turn affects the operation of the synthetic diffusion accelerator previously mentioned. The solution to this problem will be discussed in Chapter 10.

Other necessary changes to the S_N logic include modifying the calculations of scattering and fission sources to exclude Monte Carlo regions, and alteration of the particle balance tables to include the Monte Carlo totals. While somewhat detailed changes are required, the only one which affects the logic of the S_N computation

is the treatment of the fission source in DOUTER. As previously stated, the S_N employs Chebyshev accleration to improve convergence of the fission source. Since the required information (i.e., the fission source per fine mesh cell) is not available for the Monte Carlo region, it is excluded from the acceleration procedure.

Finally, the Monte Carlo region is currently excluded from the outer iteration convergence checks which the S_N performs. That is, we assume convergence in the S_N region implies convergence throughout the entire problem geometry. While this has been sufficient for the test problems examined so far, it is possible to imagine problems where it may not hold (e.g., a highly fissile material in the Monte Carlo region, surrounded by a non-multiplying S_N region). However, this can easily be remedied, if necessary, by adding a convergence check on the fission source in the Monte Carlo region.

5.7 Sampling Boundary Fluxes and Volumetric Sources

From Section 5.5 above, we see that the $\Psi_g^{in(p,\infty)}$ must be sampled for each outer iteration p. However, a complete sampling of the $\Psi_g^{in(p,\infty)}$ at each outer iteration would be counter-productive. As the outer iterations converge, the differences between the $\Psi_g^{in(p,\infty)}$ and $\Psi_g^{in(p+1,\infty)}$ decrease, so that sampling each one from scratch is a duplication of effort. More importantly, the resulting statistical deviations in the $\mathbf{Q}_g^{out(p+1)}$ will introduce fluctations into the S_N fluxes which hinder the Chebyshev acceleration of the fission source and preclude convergence.

Instead, we sample $\delta\Psi_g^{in(p,\infty)} \equiv \Psi_g^{in(p,\infty)} - \Psi_g^{in(p-1,\infty)}$. Thus, as the outer iterations converge, the $\delta\Psi_g^{in(p,\infty)}$ tend toward zero, the number of histories per sample can be reduced, and the statistical deviations of the $\mathbf{Q}_g^{out(p+1)}$ are reduced. Similarly, with SNTOMC, we sample the change in the volumetric source in the Monte Carlo region.

If the S_N operator were strictly linear, we would expect the $\delta\Psi_g^{in(p,\infty)}$ to be non-negative. Due to the use of negative flux fix-up in the S_N, however, this is not the case. While it is generally advantageous to avoid tracking negative particles in Monte Carlo calculations, since they tend to increase the variance of the results, we assume that, in this case, the magnitude of any negative residuals will be small enough so that it will not adversely affect the hybrid results. Thus, when an element of $\delta\Psi_g^{in(p,\infty)}$ is negative, we assign negative weights to the particles used to sample that state in subroutine DWNSRC. Negative particles are also used when sampling the volumetric source in the Monte Carlo region during subroutine SNTOMC when the change in the source between outer iterations is negative.

Physically, we expect the $\Psi_g^{out(p,r)}$ as computed in LINK to be non- negative. However, since the sampled $\delta\Psi_g^{in,(p-1,\infty)}$ may be negative, as stated above, we now have no guarantee that the $\mathbf{Q}_g^{out(p)}$ will remain non-negative, and thus the $\Psi_g^{out(p,r)}$ may contain negative elements. Since negative interior boundary sources are not acceptable to the S_N, we set the $\Psi_g^{out(p,r)}$ to zero for elements that are negative and adjust the remaining elements so as to conserve particles.

Similarly, the introduction of negative particles in the Monte Carlo calculation results in the possibility of negative volumetric sources in the S_N region.

To preclude this, after the Monte Carlo particles have been scored in subroutine SMOM, the zeroth moments of the resulting volumetric sources in the S_N region are checked to ensure that they are non-negative. If a negative zeroth moment is found for a given cell, all moments for that cell are set to zero. In this case, no adjustment is made to the other cells to conserve particles.

While the appearance of negative particles in a calculation is worrisome, experience to date has not shown a significant impact on the viability of the hybrid method. Since negative residuals cannot possibly appear until the second outer iteration, negative particles are of no concern whatsoever in problems involving only pure downscatter. Even with problems with fission and/or upscatter, and thus multiple outer iterations, the magnitude of the residuals sampled in succeeding outer iterations has remained small enough in comparison to the magnitude of the first outer iteration so that the total outgoing boundary fluxes and volumetric sources have remained positive when the problem is adequately sampled.

CHAPTER 6.
HYBRID MONTE CARLO METHODS

As explained in Chapter 3, it was considered more advantageous to develop our own Monte Carlo subroutines versus attempting to incorporate a more general-purpose Monte Carlo code. Since we consider only a two-dimensional geometry with rectangular mesh cells, the algorithms for describing geometries and tracking particles are considerably simpler than those for a general three-dimensional geometry, reducing the required coding and simplifying vectorization of the Monte Carlo portion of the code. Additionally, the two-dimensional algorithms permit the use of the standard TWODANT input file with very few modifications, as opposed to the more extensive input files usually required for three-dimensional Monte Carlo codes.

The hybrid Monte Carlo method as implemented is almost entirely analog, with few variance reduction techniques used. While the lack of variance reduction techniques can lead to problems in obtaining accurate estimates in some situations, the difficulties actually encountered are minimized due to two reasons. Since the Monte Carlo routines used in the hybrid method are vectorized, it is relatively inexpensive to run a large number of particles. More importantly, it is the usual practice to designate areas where the Monte Carlo method would be expensive as

regions where the S_N method is to be used. This, in fact, is the entire rationale behind the hybrid method.

The remainder of this chapter describes the details of tracking particles across mesh cells, the treatment of collisions (scattering, capture, and fission), and the sampling of sources. Chapter 7 describes the methods used in vectorizing the Monte Carlo, while Chapter 8 describes how the Monte Carlo cross sections are formed. Chapter 9 covers the implementation of error estimates with the hybrid method and the variance reduction methods used.

6.1 Tracking Particles in Phase Space

Let $\mathbf{r}$ represent the spatial coordinates, Ω the direction, and g the energy group of a neutron which has just been emitted from a source or undergone a collision. The coordinates $\mathbf{r}$ correspond to a location within a cell (i,j) as shown at Fig. 6-1a for $X-Y$ geometry, and Fig. 6-1b for $R-Z$. In order to track the particle history, we must determine whether it has a collision within the cell or crosses its boundaries, and the coordinates in phase space of those events.

We first determine the length of the flight path. From Chapter 1, the probability of a particle having an interaction in ds about s is

$$p(s)\,ds = \Sigma_{T,ijg} \exp(-\Sigma_{T,ijg} s)\,ds, \qquad 0 \le s \le \infty. \qquad (6-1)$$

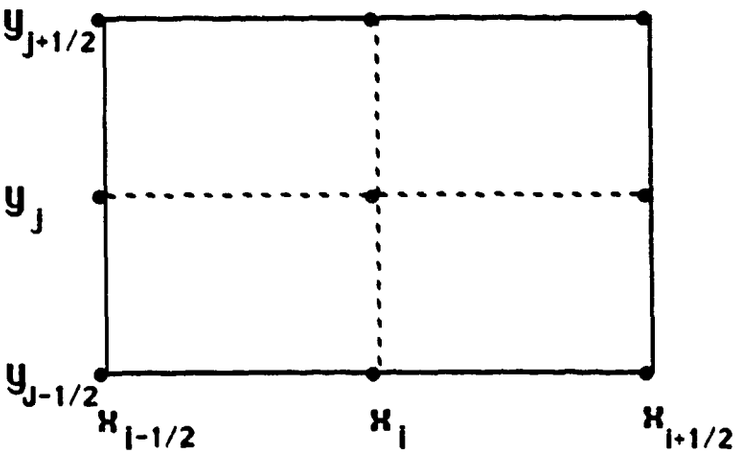

Figure 6-1a Cell Boundaries in $X - Y$ Geometry

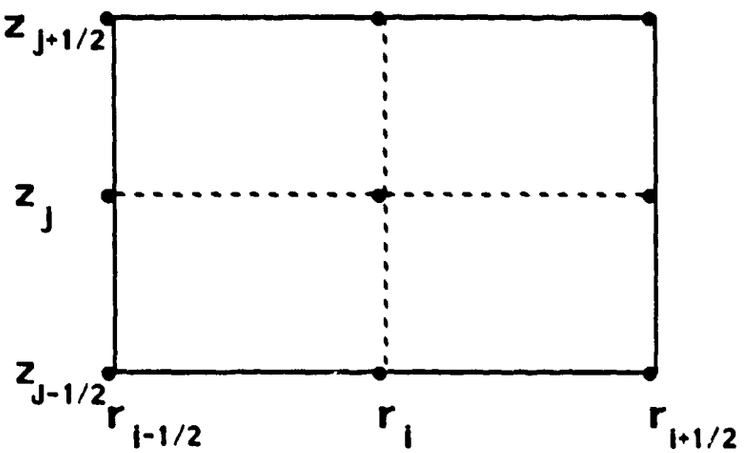

Figure 6–1b Cell Boundaries in $R - Z$ Geometry

This represents the PDF which we wish to sample uniformly. Following the methods outlined in Section 3.1, and realizing that sampling $1-\zeta$ is equivalent to sampling ζ, the track length of the particle is determined by

$$s = \frac{-\ln(\zeta)}{\Sigma_{T,ijg}}. \qquad (6-2)$$

Next, we determine if s is large enough to reach a cell boundary, and if so, which one.

In $X-Y$ geometry, the spatial coordinates are $\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$, where the origin is located at the lower left-hand corner of the problem geometry. The directional coordinates are (reference Fig. 2-1) $\mathbf{\Omega} = \mu\hat{\mathbf{e}}_x + \eta\hat{\mathbf{e}}_y + \xi\hat{\mathbf{e}}_z$. The appropriate boundaries are thus $x_B = x_{i+1/2}$ for $\mu > 0$, $x_B = x_{i-1/2}$ for $\mu < 0$, $y_B = y_{j+1/2}$ for $\eta > 0$ and $y_B = y_{j-1/2}$ for $\eta < 0$. Let $s_Y = (y_B - y)/\eta$ and $s_X = (x_B - x)/\mu$, then if s is less than both s_X and s_Y the particle undergoes a collision in the cell (i,j). Otherwise, it crosses the appropriate x-edge boundary if s_X is less than s_Y, and the y-edge boundary if s_Y is less than s_X, with s_B defined as the distance to the closest boundary (i.e., the minimum of s_X and s_Y).

Tracking particles in $R-Z$ geometry is not as straightforward as in $X-Y$ geometry, since our coordinate system is now curvilinear. The spatial coordinates are represented by $\mathbf{r} = r\hat{\mathbf{i}} + z\hat{\mathbf{j}}$, where the origin is located on the centerline of the cylinder where it passes through the lower face. The directional coordinates are (reference Fig. 2-2) $\mathbf{\Omega} = \mu\hat{\mathbf{e}}_r + \eta\hat{\mathbf{e}}_z + \xi\hat{\mathbf{e}}_\theta$. For a particle located in a cell (i,j) as shown at Fig. 6-1b, the cell boundaries are $r_{i\pm1/2}$ and $z_{j\pm1/2}$. As in Cartesian geometry, the appropriate boundaries are $z_B = z_{j+1/2}$ for $\eta > 0$, $z_B = z_{j-1/2}$ for $\eta < 0$, and $r_B = r_{i+1/2}$ for $\mu > 0$. The case for $\mu < 0$ is not determined simply

by the sign of the direction cosine, as can be seen by the cross-sectional view of a cylinder at Fig. 6-2. Instead, the cell boundary is determined by comparing the angle β formed by the projection of Ω on a cross section of the cylinder with the "critical angle" α. From Fig. 6-2, $\sin\beta = \xi/\sqrt{1-\eta^2}$, and $\sin\alpha = r_{i-1/2}/r$. Thus, if $\sin\beta > \sin\alpha$, then the appropriate boundary is $r_B = r_{i+1/2}$, else, the boundary is $r_B = r_{i-1/2}$.

The distance to boundaries in the axial direction is determined by $s_A = (z_B - z)/\eta$. The distance to the radial boundary is determined from the Law of Cosines as shown at Fig. 6-2, resulting in

$$s_R = \frac{1}{\sqrt{1-\eta^2}}\left(r\cos\beta \pm \sqrt{r^2\cos^2\beta - r^2 + r_B^2}\right). \qquad (6-3)$$

If $\beta > \alpha$, then the only root of Eq. (6-3) resulting in a positive s_R corresponds to the positive sign. For $\beta < \alpha$, there are two possibilities, since the flight path now intersects the cell boundary twice. The first intersection occurs at the lesser value of s_R, which corresponds to the minus sign in Eq. (6-3). The fate of the particle is then determined as in the case of Cartesian coordinates, i.e., boundary crossing or collision within the cell.

6.2 Tracking Particles Across Internal Boundaries

When a particle has a track length greater than the distance to the cell boundary, and that boundary is an internal boundary within the Monte Carlo region, then we must continue tracking the particle. This is accomplished by adjusting

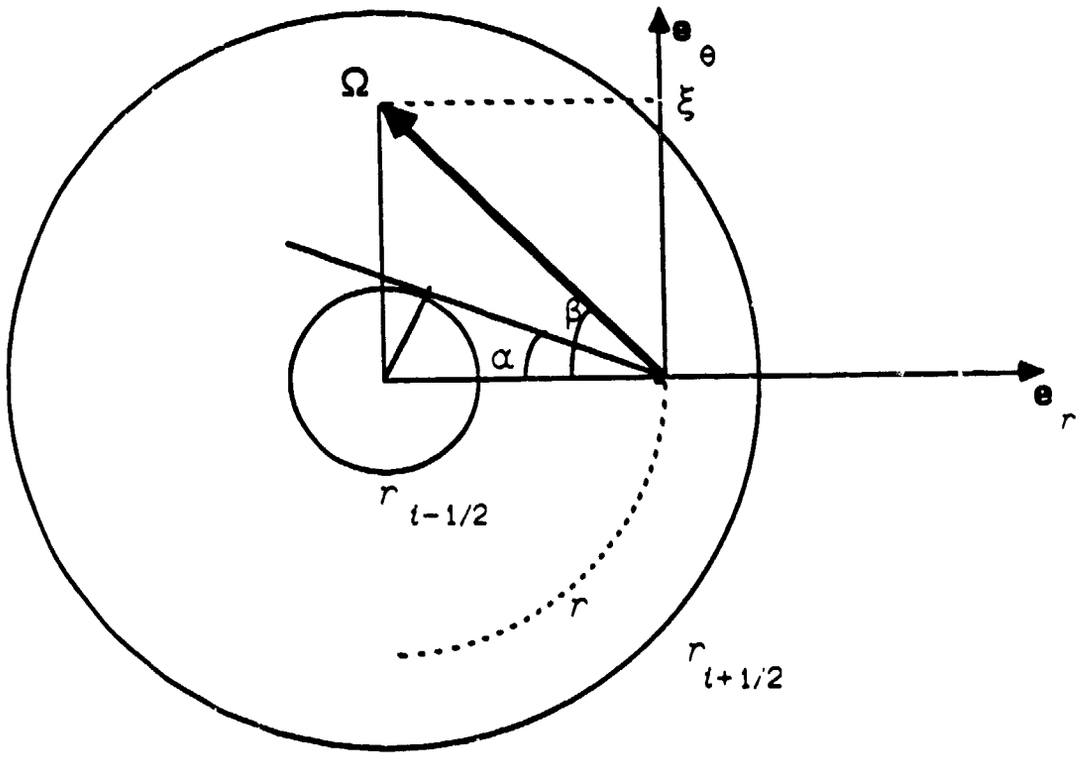

Figure 6-2 Determining Flight Direction and Distances in $R - Z$ Geometry

the track length determined by Eq. (6–2) to account for possible differences in material composition by

$$s' = s_B + (s - s_B)\frac{\Sigma_{T,ijg}}{\Sigma_{T,i'j'g}}, \qquad (6-4)$$

where the identity of the new cell (i', j') is $(i+1, j)$ if the right-hand boundary was crossed, $(i-1, j)$ for the left-hand boundary, $(i, j+1)$ for the top boundary, and $(i, j-1)$ for the bottom boundary. The particle history is then continued per Section 6.1 above, using the new cell boundaries.

6.3 Scoring Particles

If the boundary crossed represents an external boundary of the Monte Carlo region, then the particle must be scored in the appropriate boundary flux or response matrix element. First, we determine which side of the Monte Carlo region has been reached by examining the distance to cell boundaries computed above and the sign of the appropriate direction cosine (e.g., in Cartesian coordinates, if s_X is less than s_Y, and μ is negative, then the left boundary of the Monte Carlo region has been crossed). The sign of the remaining free direction cosine is used to determine which S_N quadrant the particle is to be placed into (e.g., with μ negative, η negative corresponds to the "down and in going" quadrant, η positive to the "up and in going" quadrant). The specific quadrature direction the particle is assigned to is determined by the methods described in Section 5.3.

If the particle has exited the left or right sides of the Monte Carlo region in $X-Y$ geometry, then the vertical position at which it enters the S_N region is

determined by

$$y' = y + s_B \eta. \tag{6-5}$$

For a top or bottom edge crossing, the horizontal position is found from

$$x' = x + s_B \mu. \tag{6-6}$$

The crossing position is then compared with the appropriate S_N fine mesh cell edges to determine which cell has been entered.

In $R-Z$ geometry, the axial position of a left or right edge crossing is found from

$$z' = z + s_B \eta. \tag{6-7}$$

while the radial position of a top or bottom edge crossing is

$$r' = \sqrt{(r + \mu s_B)^2 + (\xi s_B)^2}. \tag{6-8}$$

Also, since the direction cosines μ and ξ change as a particle streams in curvilinear geometry, the direction cosine along the radial direction must be recalculated prior to the quadrature direction being scored. The updated direction cosine is

$$\mu' = \left[\mu(r + \mu s_B) + \xi^2 s_B\right] \Big/ \tilde{r}. \tag{6-9}$$

where $\tilde{r} = r'$ for a top or bottom edge crossing, and r_B for a left or right edge crossing.

Regardless of whether the boundary crossed is external or internal, or if the event consists of a collision within a cell, the tracklength of the particle (modified by the particle weight) within the cell is scored. Since our tracking algorithm measures

distances from the starting point of the particle, and thus a track may cross several cells before terminating, we retain the distance to the last boundary crossed (s_L) for each particle. This distance is set to zero at the start of each particle, and updated to s_B after an internal boundary is crossed. With this information, the tracklength in the last cell entered is $s_B - s_L$ for a boundary crossing, and $s - s_L$ for a collision.

6.4 Collision Events Without Fission

When a particle undergoes a collision within a cell, we must determine its new coordinates in phase space which result from the collision. We make the standard assumption that collision times are small enough so that the position of the colliding particle is unchanged, while the new direction and energy of the scattered particle are determined by sampling from appropriate probability density functions.

In $X - Y$ geometry, the collision position is found from

$$x' = x + s\mu \qquad (6-10)$$

and

$$y' = y + s\eta, \qquad (6-11)$$

where the unprimed direction cosines refer to the pre-collision directional coordinates of the particle, and the unprimed spatial coordinates to the starting location of the particle. In $R - Z$ geometry, the collision postion is

$$r' = \sqrt{(r + \mu s)^2 + (\xi s)^2} \qquad (6-12)$$

and

$$z' = z + s\eta. \qquad (6-13)$$

Since neutron do not lose energy as they stream, the particle enters the collision in energy group g, the energy group in which it was located when it started streaming.

The collsion process itself is described by the set of multigroup, macroscopic cross sections generated from the multigroup S_N cross sections as described in Chapter 8. From Eq. (2-7), the zeroth moment group-to-group transfer cross section from group g to group g' for cell (i,j) is defined as

$$\Sigma_{ij0,g-g'} = 2\pi \int_{-1}^{+1} d\mu_0 \, \Sigma_{ij,g-g'}(\mu_0) \, P_0(\mu_0). \qquad (6-14)$$

From this definition, the probability of a neutron scattering from group g to group g' is then

$$P_{g-g'} = \Sigma_{ij0,g-g'} \, / \, (\Sigma_{T,ijg} - \Sigma_{A,ijg}) \qquad (6-15)$$

where $\Sigma_{A,ijg}$ is the macroscopic absorption cross section. The $P_{g-g'}$ are then used to form a cumulative distribution function, which is sampled to determine the group g' in which the scattered neutron emerges.

Note that to conserve storage, cross sections are actually referenced by material, not cell identifier, since the number of materials in a problem is typically much less than the number of mesh cells. Note also that while a cell's material may consist of a number of different atomic elements/isotopes, the hybrid Monte Carlo method uses the mixed S_N macroscopic cross sections.

Once the energy group of the scattered neutron has been found, we must next determine its direction. As described in Chapter 8, we generate 32 equally probable

directional bins with respective widths $\Delta\mu_{0i}$, $i = 1, \ldots, 32$, from the group-to-group transfer cross section $\Sigma_{ij,g\to g'}(\mu_0)$ of Eq. (2–6). The number 32 was chosen because, historically,[18] it seems to accurately reproduce the angular cross sections without requiring large amounts of storage. Note that μ_0 (i.e., $\mathbf{\Omega}' \cdot \mathbf{\Omega}$) is measured in the laboratory frame of reference.

We then compute the bin i the neutron is scattered into by drawing a random number ζ and setting $i = \text{Int}(1 + 32\zeta)$, where Int is the integer function. The scattered particle is assumed to be uniformly located in bin i, with the actual scattering angle computed from

$$\mu_0 = \mu_{0i,L} + (i - 32\zeta)\,\Delta\mu_{0i}, \tag{6-16}$$

where $\mu_{0i,L}$ denotes the angle corresponding to the left-hand edge of the ith bin.

Once the scattering angle μ_0 has been determined, the direction cosines of the scattered particle may be calculated, assuming that the scattered neutrons are uniformly distributed throughout the azimuthal scattering angle ϕ. In Cartesian coordinates, the new direction coordinates μ', η', and ξ' are[18]

$$\mu' = \mu\mu_0 + \sqrt{(1-\mu_0^2)/(1-\xi^2)}\,(\mu\xi\cos\phi - \eta\sin\phi), \tag{6-17a}$$

$$\eta' = \eta\mu_0 + \sqrt{(1-\mu_0^2)/(1-\xi^2)}\,(\eta\xi\cos\phi - \mu\sin\phi), \tag{6-17b}$$

and

$$\xi' = \xi\mu_0 - \sqrt{(1-\mu_0^2)(1-\xi^2)}\cos\phi. \tag{6-17c}$$

Instead of actually calculating the cosine and sine of the azimuthal scattering angle, they are determined through rejection sampling, where, given random numbers ζ_1 and ζ_2 uniformly distributed on the interval $[-1, 1]$ with rejection criteria $\zeta_1^2+\zeta_2^2 \leq 1$,

$$\begin{aligned} \cos\phi &= \zeta_1 \Big/ \sqrt{\zeta_1^2 + \zeta_2^2}, \\ \sin\phi &= \zeta_2 \Big/ \sqrt{\zeta_1^2 + \zeta_2^2}. \end{aligned} \qquad (6-18)$$

In $R-Z$ geometry, we first calculate post-collision direction cosines μ', η', and ξ' per Eqs. (6-17) and (6-18) above. Since the direction cosines vary with the position of the particle, which is updated during the collision process, the radial direction cosine corresponding to the revised particle position is

$$\tilde{\mu} = \left[\mu'(r + \mu s) + \xi'\xi s\right] \Big/ r'. \qquad (6-19)$$

The axial direction cosine η' is, of course, unchanged. Instead of directly calculating the azimuthal direction cosine $\tilde{\xi}$, we first find the cosine and sine of the angle β (see Fig. 6-2), which are required for tracking in cylindrical geometry, from

$$\cos\beta = -\tilde{\mu} \Big/ \sqrt{1-\eta'^2} \qquad (6-20a)$$

and

$$\sin\beta = \sqrt{1-\cos^2\beta}. \qquad (6-20b)$$

We then calculate

$$\tilde{\xi} = \sin\beta\sqrt{1-\eta'^2}. \qquad (6-21)$$

Once the new energy group and directional coordinates of the particle have been calculated, the updated spatial coordinates are examined to see if the particle

is now located outside the group g' Monte Carlo boundaries. If so, the fine mesh cell the particle is located in is determined and the particle is scored per Section 5.4.

In analog Monte Carlo, the weight of a particle would remain a constant when undergoing a collision, with absorption being modeled through the complete elimination of a particle based on a sampling of the probability of survival

$$c = 1 - \Sigma_{A,ijg} / \Sigma_{T,ijg}. \qquad (6-22)$$

Instead, we choose to implement a standard variance reduction technique known as "implicit capture" or "absorption suppression", where explicit absorption is replaced by reducing the particle weight by the factor c.

If $c \ll 1$, or if the Monte Carlo region is large and the particles undergo many collisions before escaping, the use of implicit capture may result in very small particle weights. Since the time required to track a particle is independent of its weight, this results in the expenditure of large amounts of computational resources on particles whose weights are too small to affect the calculation's results. This is remedied by imposing a weight cutoff known as "Russian Roulette".[18] Particles with a weight w below a user-input weight cutoff of w_2 are assigned a probability of survival w/w_1, where w_1 is also a user-input number. The survival probability is then sampled by drawing a random number ζ on the interval [0,1]. If $\zeta > w/w_1$, the particle is eliminated, otherwise, it is assigned the weight w_1. While this violates particle conservation on an individual basis, on the average it will still hold if "enough" histories are run.

As discussed in Chapter 4, explicit outscatter is replaced with a reduction in particle weight during the response matrix calculation. This is implemented by replacing the value of c normally used with

$$c_{RM} = \Sigma_{ij0,g\to g} \,/\, \Sigma_{T,ijg}, \qquad (6-23)$$

where $\Sigma_{ij0,g\to g}$ is as defined in Eq. (6–14). All particles remain within group g after a collision during the response matrix calculation for $\underline{\tilde{\mathbf{R}}}_{gg}$.

1.5 Collision Events with Fission

Typically, Monte Carlo codes model the fission process by sampling the probability that a collision results in a fission, and, if one occurs, sampling appropriate distribution functions to determine the number of fission neutrons emitted and their energies. All of the resulting particles are then tracked separately.

With the hybrid method, we chose to model fission in a manner analogous to implcit capture, which we denote by "implicit fission". That is, the probabilitv of a fission event, the number of fission neutrons emitted, and their energy distribution are modeled by adjustments in particle weight and outscatter probabilities of the colliding particle, without the creation of additional particles. This is done for two reasons:

1. The cross section data from the S_N as obtained by the Monte Carlo is given in terms of the quantity $\nu\Sigma_{F,ijg}$. Thus, tables of ν (the number of neutrons emitted per fission) are not readily available for sampling.

2. Generation of separate fission particles would entail increased storage requirements for vectorization, since excess fission particles must be stored until they can be included in the appropriate event stack (see Chapter 7).

In implicit fission, the particle weight modifier becomes

$$c' = c + \nu\Sigma_{F,ijg}/\Sigma_{T,ijg}, \qquad (6-24)$$

and, for response matrix calculations,

$$c'_{RM} = c_{RM} + \nu\Sigma_{F,ijg}\chi_g/\Sigma_{T,ijg}, \qquad (6-25)$$

where χ_g is as defined at in Eq. (2-4). Note that c', and even c'_{RM}, may now be greater than one. The probability of "scattering" to group g' from group g [Eq. (6-15)] becomes

$$P_{g\rightarrow g'} = \frac{\Sigma_{ij0,g\rightarrow g'} + \nu\Sigma_{F,ijg}\chi_{g'}}{\Sigma_{T,ijg} - \Sigma_{A,ijg} + \nu\Sigma_{F,ijg}}. \qquad (6-26)$$

Also, since we assume fission neutrons are emitted isotropically, we revise the zeroth moment of the angular scattering cross section to include the fission neutrons, i.e.,

$$\Sigma'_{ij0,g\rightarrow g'} = \Sigma_{ij0,g\rightarrow g'} + \nu\Sigma_{F,ijg}\chi_{g'}. \qquad (6-27)$$

The revised angular scattering cross section is used in calculating the angular bins $\Delta\mu_{0i}$ (see Chapter 8).

Since fission neutrons are now indistinguishable from normally scattered neutrons, but we wish to include separate categories for the two when forming integral

balance tables, we calculate the probability that a "scattered" neutron is actually a fission neutron. This probability is

$$P_{FN,g\rightarrow g'} = \frac{\nu\Sigma_{F,ijg}\chi_{g'}}{\Sigma_{ij0,g\rightarrow g'} + \nu\Sigma_{F,ijg}\chi_{g'}}. \qquad (6-28)$$

This probability is used to divide the weight w of a particle between the fission and inscatter entries of the balance table, where the fission weight entering a group g' after a collision in group g is $w\,P_{FN,g\rightarrow g'}$, and the inscatter weight is $w(1 - P_{FN,g\rightarrow g'})$.

The implicit fission method is disadvantageous in regions where c' (or c'_{RM}) is greater than one, since if a particle undergoes multiple collisions within such a region, than its weight can grow without limit. The creation of such "heavy" particles increases the variance of a problem, as will be seen in some of the benchmark problems of Chapter 11. Thus, consideration should be given towards reverting to a more traditional treatment of fission neutrons in the future, since none of the reasons listed above for using implicit fission pose insurmountable barriers to an analog treatment of fission.

1.6 Sampling the Fixed Source

As explained in Chapter 4, we sample the fixed source in order to determine the $\mathbf{S}_g^{out}$. We assume at present that the fixed source is entirely located within the Monte Carlo region, although nothing in the hybrid method inherently precludes a fixed source split between the Monte Carlo and S_N regions, or even one entirely located within the S_N region. We also assume a normalization of the fixed source

intensity to one neutron per second. Provisions are currently included within the hybrid method for four types of fixed sources:

1. An isotropic distributed source,

2. An isotropic point source,

3. An isotropic surface flux source, and

4. A monodirectional beam source ($X - Y$ geometry only).

In $X - Y$ geometry, the isotropic distributed source is implemented with input parameters x_1, x_2, y_1, and y_2, which refer to the left, right, bottom, and top limits of the distributed source, respectively. To determine the starting spatial coordinates (x, y) of a particle, we define $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$, then draw random numbers ζ to get $x = x_1 + \zeta \Delta x$ and $y = y_1 + \zeta \Delta y$. The initial coarse mesh cell is found by comparing the starting spatial position with arrays containing the S_N coarse mesh cell edge positions. Since an isotropic source is uniform in the azimuthal angle ϕ and the sine of the polar angle θ, and $d\eta = \sin\theta \, d\theta$ (reference Fig. 2–1), the initial direction cosines of the particle are found from $\eta = 2\zeta - 1$ and $\phi = \pi\zeta$, with $\mu = \sqrt{1 - \eta^2} \cos\phi$ and $\xi = \sqrt{1 - \eta^2 - \mu^2}$.

In $R - Z$ geometry, the input parameters are r_1, r_2, z_1, and z_2, which denote the inner, outer, bottom, and top limits of the distributed source, respectively. The initial axial coordinate is computed as above, with $\Delta z = z_2 - z_1$, and $z = z_1 + \zeta \Delta z$. In cylindrical geomtery, since we assume the source is distributed evenly throughout the volume of the cylinder within the limits of the input parameters,

the distribution along the radial direction is uniform in r^2, not r. In terms of the method described at Section 3.1, our probability density function (PDF) is related to a uniform distribution by

$$d\zeta' = \frac{2r'dr'}{(r_2^2 - r_1^2)} \qquad (6-29)$$

Integrating $d\zeta'$ from 0 to ζ, and dr' from r_1 to r, and solving for r, we obtain

$$r = \sqrt{\zeta(r_2^2 - r_1^2) + r_1^2}. \qquad (6-30)$$

The direction cosines μ and η are determined as in $X-Y$ geometry above, while Eqs. (6-20) and (6-21) are used to determine ξ and $\sin\beta$ (required for tracking in cylindrical geometry as discussed in Section 6.1).

For all source options and geometries, the initial energy group of a particle is determined from a user-input array containing the source spectrum S_g. The discrete PDF describing the source spectrum is thus

$$p_g = S_g \Big/ \sum_{g'=1}^{G} S_{g'}. \qquad (6-31)$$

This PDF is used to form a cumulative distribution function, which in turn is uniformly sampled to obtain the initial energy group of a particle.

The point source options are handled similarly to the distributed source options, except that the starting spatial position, and thus the initial coarse mesh cell, are fixed, and the only input parameters required are x_1 (r_1) and y_1 (z_1). Note that for $R-Z$ geometry, the "point source" option actually represents a circular ring when $r_1 > 0$.

The isotropic surface flux option in $X-Y$ geometry consists of an isotropic flux impinging on the left surface of the Monte Carlo region with top and bottom boundaries y_2 and y_1, respectively. Thus, the x coordinate is fixed at zero, while the y coordinate is determined as in the case of a distributed source. The initial directional coordinates are determined by sampling the incoming current which, since the flux is isotropic, is proportional to μ. Thus, the PDF is related to the uniform distribution by

$$2\mu' d\mu' = d\zeta'. \qquad (6-32)$$

Integrating μ' over the limits 0 to μ, since it is an incoming current from the left, and solving for μ, we obtain $\mu = \sqrt{\zeta}$. Since the surface flux is isotropic, the current is azimuthally symmetric about the μ, and the remaining direction cosines are determined by $\eta = \sqrt{1-\mu^2}\cos\phi$ and $\xi = \sqrt{1-\mu^2-\eta^2}$, where $\phi = 2\pi\zeta$.

In $R-Z$ geometry, the isotropic surface flux option represents an isotropic flux impinging on the outer face of the cylinder, with top and bottom boundaries z_2 and z_1. Thus, the option is entirely analogous to the $X-Y$ surface flux option, except that $\mu = -\sqrt{\zeta}$. The sine of the angle β required for tracking in $R-Z$ geometry is computed from $\beta = -\arctan(\xi/\mu)$.

Finally, the monodirectional beam source in $X-Y$ geometry consists of initial spatial coordinates x_1 and y_1, with fixed directional coordinates η_s and ϕ_s, from which $\mu_s = \sqrt{1-\eta_s^2}\cos\phi_s$ and $\xi_s = \sqrt{1-\eta_s^2-\mu_s^2}$.

1.7 Sampling the Response Matrix

From Chapter 4, we sample the group-within-group response matrix $\underline{\tilde{\mathbf{R}}}_{gg}$, where the element r_{ij} represents the angular flux leaving the Monte Carlo region in S_N state i due to a unit incident angular flux in S_N state j. The sampling is performed by assigning N_j histories to the jth column of $\underline{\tilde{\mathbf{R}}}_{gg}$, which are then followed through the Monte Carlo region to determine the ith state (row) in which they reenter the S_N region. As described in Section 6.4 above, outscatter is not allowed, so each history can terminate only by physically crossing a Monte Carlo/S_N interface or an outer boundary surface. (If a weight cutoff is in effect, then histories may also be terminated through particle elimination.) Thus, each particle contibutes to its respective column's score, ensuring maximum efficiency.

Currently, the same value of N_j, which we denote by N_{RM}, is used for each column, regardless of its actual importance. The efficiency of the response matrix calculation could be increased by performing a single S_N iteration, using the boundary fluxes from the sampling of the fixed source, to determine the approximate incoming boundary fluxes from the S_N into the Monte Carlo region. The incoming angular boundary fluxes would then be used as a weight to determine the relative size of the N_j. Alternatively, the entire problem could first be solved by a low order S_N calculation, with the appropriate cell edge fluxes saved to be used as a weighting function. The latter approach would be preferable in problems with large fission or upscatter cross sections, since it would allow the use of outer iterations. The determination of the relative merits of the two approaches, or of other methods, is left as a future problem.

In sampling the reponse matrix, each incoming S_N state (column) corresponds to the edge of a unique fine mesh cell, located along the S_N/Monte Carlo interface. Since the Monte Carlo region is rectangular, there are a maximum of four different locations along the interface for a given state – top, bottom, left, or right. The actual number may be less than four, since a boundary of the Monte Carlo region may be located along a vacuum or reflective ($R - Z$ geometry) boundary condition, in which case the corresponding columns of the response matrix are not needed.

In $X - Y$ geometry, if a state is located along the top boundary, then the particles used to sample that state have their initial y coordinate fixed at y_T. Assuming that the state corresponds to the ith fine mesh cell, with width Δx_i and left edge $x_{L,i}$, the initial x coordinate is found from $x = x_{L,i} + \zeta \Delta x_i$. Since the initial fine mesh cell location is already known, the starting coarse mesh cell is easily determined from previously calculated S_N arrays. For cells along the bottom interface, the identical procedure is used, except that y is fixed at y_B. Similarly, for the left and right edges, x is fixed at x_L and x_R, respectively, while y is found from $y = y_{B,i} + \zeta \Delta y_i$.

In $R - Z$ geometry, the left and right edges are sampled as above, with r fixed at r_L or r_R, and z found from $z = z_{B,i} + \zeta \Delta z_i$. Along the top and bottom edges, however, the distribution is uniform in r^2, not r, as in the case of the isotropic distributed source in the preceeding section. Thus, the initial radial coordinate is found from $r = \sqrt{\zeta(r_{L,i+1}^2 - r_{L,i}^2) + r_{L,i}^2}$.

Along each edge, there are two different quadrants of directions. For example, the direction cosine η is always negative ("downgoing") along the top edge, while the direction cosine μ may be negative ("incoming") or positive ("outgoing"). The allowable combinations are shown at Table 6.1 below. Each quadrant consists of a set of S_N directions $\mathbf{\Omega}_n$ as shown at Fig. 5-3, with an associated solid angle $\Delta\mathbf{\Omega}_n$ as described in Section 5.3. For a given state j with associated direction $\mathbf{\Omega}_j$, the incoming particles are assumed to be uniformly distributed over the surface of the solid angle $\Delta\mathbf{\Omega}_j$. Thus, if η_B is the cosine of the polar angle θ_B associated with the "bottom" edge of the $\mathbf{\Omega}_j$ bin, then the direction cosine η is determined by $\eta = \eta_B + \zeta\,\Delta\eta$, where $\Delta\eta$ is as defined in Section 5.3. The sign of η is found from Table 6-1. The azimuthal angle is determined in a similar manner, with Δo representing the bin width, and o_L the bin's "left" edge. The azimuthal angle of the incoming particle is then found from $o = o_L + \zeta\,\Delta o$, where the resulting value of o is in the range $[0, \pi/2]$. Depending upon the required sign of μ from Table 6-1, the constant $\pi/2$ is then added, and the direction cosines μ and ξ are calculated as described above.

Table 6-1
Allowable Directional Quadrants

Edge	μ	η
Top 1	−1	−1
Top 2	+1	−1
Right 1	−1	−1
Right 2	−1	+1
Bottom 1	−1	+1
Bottom 2	+1	+1
Left 1	+1	−1
Left 2	+1	+1

1.8 Sampling the Incoming Boundary Fluxes

As described in Sections 5.5 and 5.7, subroutine DWNSRC samples the residual incoming boundary fluxes $\delta\Psi_g^{in(p,\infty)}$ to determine their contribution to the $\mathbf{Q}_{g'}^{out(p+1)}$. The mechanics for sampling the incoming boundary fluxes are identical to those of the response matrix described above, with the exceptions that particles are allowed to scatter to a different group upon collision (reference Section 6.4), and the number of histories per state (N_j) is weighted to reflect the residual number of particles entering the Monte Carlo region from the S_N region in state j.

To begin, we calculate the total residual leakage L^p into the Monte Carlo region for outer iteration p from

$$L_g^p = \sum_{j=1}^{J_g} |\delta\psi_{g,j}^{in(p,\infty)}|\,\Delta\Omega_j\,\Delta A_j, \qquad (6-33)$$

where $\Delta\Omega_j$ is the solid angle associated with state j, ΔA_j the surface area, and J_g is the total number of states in the group g Monte Carlo/S_N interface. The absolute value sign is used since there is no guarantee that the residuals will be non-negative. For the first outer iteration ($p = 1$), the total number of histories to be run for group g (N_g^1) is determined from

$$N_g^1 = J_g \times N_{RM}, \qquad (6-34)$$

so that approximately the same amount of effort is used in sampling the incoming boundary fluxes as was used in sampling the response matrix. For subsequent residuals, the decreasing importance of the residuals is reflected by using

$$N_g^p = J_g \times N_{RM} \times L_g^p / L_g^{p-1}. \qquad (6-35)$$

To prevent arbitrary increases in N_g^p when $L_g^p > L_g^{p-1}$, as can sometimes occur for large p, the maximum allowable value of L_g^p/L_g^{p-1} is fixed at one. The actual number of histories run for an individual state j is

$$N_j = N_g^p \, |\delta\psi_{g,j}^{in(p,\infty)}| \, \Delta\Omega_j \, \Delta A_j \, / \, L_g^p. \qquad (6-36)$$

To correct for roundoff error in N_j, since only an integral number of histories can be run, and to incorporate the possibility of negative residuals, the weight assigned to the state j particles is

$$w_j = \mathrm{Sgn}(\delta\psi_{g,j}^{in(p,\infty)}) \, N_j \, / \, \mathrm{Int}(N_j), \qquad (6-37)$$

where Sgn is the sign function.

1.9 Sampling the Volumetric Sources

From Section 5.4, subroutine SNTOMC determines the contribution of the group g volumetric sources located in the Monte Carlo region to the $\mathbf{Q}_g^{out}$. Since the S_N stores only the moments of the volumetric source $\tilde{V}_{ijlmg}$ and, from Section 5.7, we wish to sample the residual volumetric source, subroutine SNTOMC begins by computing the residual moments for outer iteration p from

$$\delta\tilde{V}_{ijlmg}^p = \tilde{V}_{ijlmg}^p - \tilde{V}_{ijlmg}^{p-1}, \qquad (6-38)$$

where ij denotes the fine mesh cell (i,j), and lm the lth, mth spherical harmonic (reference Section 2.3). The total residual volumetric source in the Monte Carlo region is then determined from

$$T_g^p = \sum_{i,j} |\delta\tilde{V}_{ij00g}^p| \, \Delta V_{i,j}, \qquad (6-39)$$

where $\Delta V_{i,j}$ is the volume of cell (i,j), and the sum is over all fine mesh cells located within the group g Monte Carlo region. The absolute value signs are used since the $\delta\tilde{V}^p_{ij00_g}$ may be negative. Since we wish to assign approximately equal weight to the sampling of the fixed source and the sampling of the volumetric sources, the total number of histories to be run for the group g volumetric source is calculated from

$$N^p_g = \mathrm{Int}(N \times T^p_g), \tag{6-40}$$

where N is the number of histories used in sampling the fixed source.

To actually sample the volumetric source in cell (i,j), we first reconstruct the residual angular flux $\delta\phi_{ijng}$ from

$$\delta\phi^p_{ijng} = \sum_{l=0}^{L} \sum_{m=0}^{+l} Y_{lmn}\, \delta\tilde{V}^p_{ijlmg}, \tag{6-41}$$

where Y_{lmn} represents the lth, mth spherical harmonic evaluated at the discrete direction $\mathbf{\Omega}_n$.

Once the residual angular flux has been reconstructed, the number of histories assigned to it is found from

$$N^p_{ijng} = N^p_g\, |\delta\phi^p_{ijng}|\, \Delta V_{ij}\, \Delta\mathbf{\Omega}_n \,/\, T^p_g. \tag{6-42}$$

As in the outscatter calculation, since only an integral number of histories may be run, and $\delta\phi^p_{ijng}$ may be negative due to not only the use of residuals, but the use of the spherical harmonic representation, the weight assigned is

$$w_{ijng} = \mathrm{Sgn}(\delta\phi^p_{ijng})\, N^p_{ijng} \,/\, \mathrm{Int}(N^p_{ijng}). \tag{6-43}$$

The sampling of the angular phase space coordinates is identical to that of the response matrix, with the addition that since we have a volumetric source there are now a total of four possible directional quadrants per cell. The spatial phase space coordinates are also sampled in a similar manner, except that both spatial coordinates are sampled for each history, of course.

CHAPTER 7.
VECTORIZATION OF THE HYBRID MONTE CARLO METHOD

Monte Carlo is inherently a sequential process, where the natural flow of computation is to follow a single history until it terminates, and only then initiate the next history. While this method of processing histories is perfectly acceptable for machines with sequential processors, it fails to fully use the capabilities of machines with vector processors. Essentially, a vector processor is capable of performing the same operation, or series of operations, on numerous sets of data (64 on Cray machines) at the same time. Ideally, then (neglecting overhead), a vector machine could process 64 particles in the same amount of time it would take a sequential machine to process just one. The catch is the requirement that the vector processor perform an identical set of operations on a given set of data. While it is fairly easy to adapt the logic of S_N codes to form arrays that can be processed vectorially, the numerous possible logical paths available to individual Monte Carlo particles (e.g., collision, internal boundary crossing, etc.) inhibit the vectorization of Monte Carlo codes. Thus, the key to vectorizing Monte Carlo codes is to somehow group histories into sets which require identical processing actions.

Since Monte Carlo calculations are typically much more CPU-intensive than S_N calculations, it is crucial to minimize the amount of time spent in tracking particles wherever possible. This requirement also applies equally as well to the

hybrid method, where the time spent in the Monte Carlo portion of the code usually far outweighs the time spent in the S_N portion of the code. Thus, to maximize the efficiency of the hybrid method, it is desirable to vectorize the hybrid Monte Carlo method.

7.1 The Event-Based Vectorization Method

The method we have chosen to vectorize the hybrid Monte Carlo method involves the formation of event-based stacks,[21-23] where each stack consists of a group of particles undergoing an identical event, such as a collision. To maximize efficiency, since a vector processor requires roughly the same time to process a stack containing one particle as one containing sixty-four, a stack is only processed, if possible, when it is full. After a stack has been processed, the particles are distributed to other stacks for further processing, if required. Note that we do not physically transfer a particle, of course, but rather its attributes, such as phase coordinates, and other information required for stack processing, such as the distance to the next boundary. As soon as another stack fills up, it to is processed, and its particles are in turn redistributed.

Because a particle can undergo numerous events before its history is terminated, and these events have no set sequence, the event stacks necessarily have multiple connections to one another. That is, a stack can typically receive particles from more than one other stack, and can, in turn, transfer particles to multiple stacks. Thus, more than one stack may fill up at the "same" time, and, as a result

there may be an attempt to transfer more particles to a given stack than it has space available.

To solve these problems, a dynamic stack control system with individual stack reservoirs is used. For each type of Monte Carlo calculation in the hybrid method, there is a specific driver subroutine which controls the overall movement of particles into and within the system. A driver subroutine begins by introducing source particles into the system, which are then sorted according to the criteria of Section 6.1 and placed into the appropriate event stack. As an event stack fills up with these source particles, its associated subroutine assumes temporary control, processing its stack and redistributing its particles as required. If the redistribution of particles results in additional stacks filling up, control is passed down to their associated subroutines, in a prioritized order, for their execution. If necessary, these subroutines can also pass control down to yet another subroutine(s), if their execution results in additional stacks filling up. As soon as a subroutine has finished execution, including the execution of any subroutine it passed control to, it in turn releases control back to the subroutine which originally passed it control. Once control has been returned all the way back to the driver subroutine, it introduces more particles into the system, either from the source or an individual stack reservoir.

Individual stack reservoirs are required since there exists the possibility that a subroutine may need to transfer particles to a stack in which there is no more room. In this case, the excess particles are temporarily placed in a holding stack, from which they will be transferred to their original destination when space is available. The term "individual" stack reservoir is used since, unlike most other

vectorized Monte Carlo codes.[21-23] there is no single master reservoir into which all excess particles are placed. Rather, the stacks where logjams may occur have an additional stack(s) to act as a reservoir for excess particles. Individual stack reservoirs are felt to be more efficient than a single master reservoir, since particles in an individual stack reservoir have their destination and all necessary information already available, whereas particles in a master reservoir must be resorted. However, the use of individual stack reservoirs does present difficulties in problems when particles are created through fission, since a master reservoir is easily enlarged to hold the additional particles, while the creation of additional reservoirs is not as straightforward. While the use of implicit fission (reference Section 6.5) renders the problem moot, the individual stack reservoirs should probably be replaced with a master reservoir if analog fission is used.

As previously stated, a stack contains all of the particle attributes necessary for the execution of its associated subroutine. When a particle is transferred from one stack to another, these attributes must be moved from their location in the controlling stack to locations in the receiving stack. Since a single history will typically undergo several stack transfers, the overhead involved in transferring particles can easily form a substantial portion of the computational time. Two actions have been taken to minimize this overhead.

First, a set of Cray Assembly Language (CAL) utility routines are used to transfer particle attributes.[21-23] CAL masking routines are first used to encode particle destination words for each stack, where a bit value of one corresponds to a particle to be transferred. Next, the CAL routine MOVXTOB is used to actually

transfer the particle attributes into open slots in the destination stack. A particle's attributes are transferred vectorially, while each particle itself is transferred sequentially. As a particle is moved, MOVXTOB resets the appropriate bit in the particle destination word to zero, thus maintaining a record of which particles have been transferred. This is important in the event the destination stack fills up, and particle transferrence must be suspended while it is executed.

Second, the number of attributes that need to be transferred are kept to a minimum, since not all stacks require all attributes. Because the utility routine MOVXTOB requires a transferred particle's attributes to be arranged sequentially in memory, the attribute lists are structured so that the minimum number of attributes required to meet the needs of the target subroutine, or its chain of possible subsequent target subroutines, are actually moved, within the limits of the sequential memory requirement of MOVXTOB. Furthermore, some attributes are calculated only as needed, and not saved for transferrance, if the time required for their computation is minimal.

The vectorization scheme employed in the hybrid method is shown at Fig. 7-1. The names in the boxes refer to the vectorized Monte Carlo subroutines, while the associated event stack a subroutine executes is listed below. The stacks in dashed boxes without subroutines are individual stack reservoirs. The lines and arrows represent possible particle transfer paths between the subroutines, while the numbers adjacent to the lines refer to the number of particle attributes transferred between stacks. Where there are two numbers, the first refers to $X - Y$ geometry, while the second, in parenthesis, is for $R - Z$ geometry. Note that there are eight event stacks, with three individual stack reservoirs (Stack 7 is not currently used).

The number of attributes required to be transferred between stacks ranges from 9 to 20. The stack attributes are listed at Table 7-1 below, where the symbols are as defined at in Chapter 6, with the exception that x_M, y_M, r_M, and z_M refer to the appropriate Monte Carlo region boundary for a given direction and energy group. The attributes CONST, XCOSB, SINB, and YSIN are used only for cylindrical geometry calculations.

Table 7-1
Stack Attributes

Name	Symbol	S0	S1	S2	S3	S4	S5	S6	S11
CONST	$r^2(\cos^2\beta - 1)$	•	•	•	•			•	
XCOSB	$r\cos\beta$	•	•	•	•			•	
SINB	$\sin\beta$	•	•	•	•			•	
YSIN	$\sqrt{1-\eta^2}$	•	•	•	•			•	
X	$x\ (r)$	•	•	•	•	•	•	•	•
Y	$y\ (z)$	•	•	•	•	•	•	•	•
U	μ	•	•	•	•	•	•	•	•
ETA	η	•	•	•	•	•	•	•	•
TSI	ξ	•	•	•	•	•	•	•	•
WT	w	•	•	•	•	•	•	•	•
S	s	•	•	•	•	•	•	•	•
IR	(i,j)	•	•	•	•	•	•	•	•
IG*	g	•	•	•	•	•	•	•	•
SL	s_L	•	•	•	•	•	•	•	
SBX	$s_X\ (s_R)$	•	•		•	•	•	•	
SBY	$s_Y\ (s_Z)$	•	•		•	•	•	•	
XMAX	$x_M\ (r_M)$	•	•		•	•		•	
YMAX	$y_M\ (z_M)$	•	•		•			•	
BOUNDX	$x_B\ (r_B)$	•	•		•			•	
BOUNDY	$y_B\ (z_B)$	•	•		•			•	

* For response matrix calculations, IG contains the particle's starting column

The vectorization scheme begins with the determination of the starting coordinates in phase space for 64 particles located in Stack 0, which is performed by subroutine

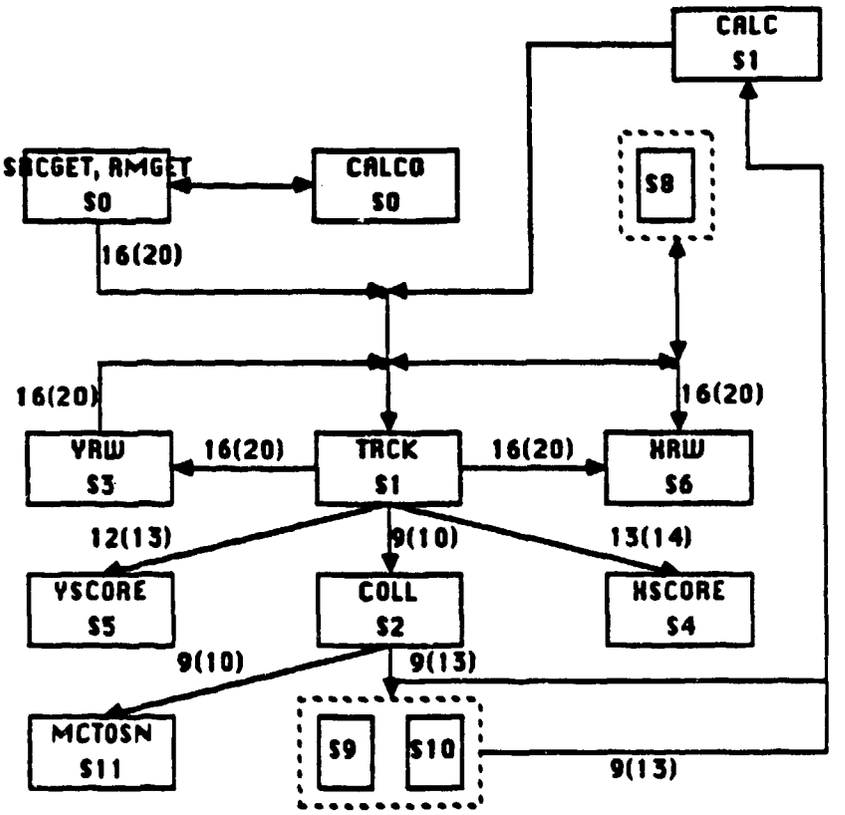

Figure 7-1 The Hybrid Monte Carlo Vectorization Scheme

SRCGET for fixed source Monte Carlo calculations (Section 6.6), and RMGET for all others (Sections 6.7 through 6.9). The remaining attributes in Stack 0 are then calculated by a call to subroutine CALC0, which determines the path length s and performs the distance to boundary calculations described in Section 6.1. Once all necessary attributes have been determined, the particles are transferred to Stack 1, and their destination (i.e., their next event stack) is determined by subroutine TRCK as also described in Section 6.1. Particles exiting the Monte Carlo region are transferred to XSCORE (Stack 4) or YSCORE (Stack 5) for scoring (reference Section 6.3), those crossing an internal Monte Carlo boundary are sent to XRW (Stack 6) or YRW (Stack 3) (reference Section 6.2), while those undergoing a collision are sent to subroutine COLL (Stack 2) (reference Sections 6.4 and 6.5). For response matrix calculations, subroutine COLL is replaced with subroutine RMCOLL, which uses the modified cross sections required for group-within-group only scattering.

If a stack or stacks fill up during the transfer process from TRCK, the particle transferrance process is suspended while the target stack is executed. For Stacks 4 and 5 this presents no problem, since a particle's history is terminated when it leaves the Monte Carlo region. Execution of Stacks 2, 3, and 6 is not as straightforward, however, since they attempt to feed back into Stack 1, which may still contain particles. This conflict is resolved by having the target stack fill the available slots in Stack 1, thus partially emptying the target stack and enabling the transferrance process from TRCK to resume. As more slots in Stack 1 open up, the remaining particles in the target stack are transferred into them. The process is very similar to juggling, where, instead of balls, we are using particles.

The juggling process breaks down, however, when two, or even all three, of Stacks 2, 3, or 6 fill up and attempt to feed back into Stack 1 simultaneously. To resolve this conflict, priority is given to one of the stacks, which places a lock on Stack 1, forcing the remaining stacks to transfer their particles to the individual particle reservoir. Stack 3 (YRW) is assigned the highest priority, followed by Stack 6 (XRW), and then Stack 2 (COLL). This scheme is used since less memory is required to store the particle attributes of Stack 2. Because of the above priority scheme and control structure, one individual stack reservoir is sufficient for XRW and YRW, while two are required for COLL.

Subroutine COLL generates new phase space coordinates and weights for all scattered particles, and performs the "Russian Roulette" weight cutoff process described in Section 6.4. COLL then checks to see if the new phase space coordinates place a particle outside the Monte Carlo region. If so, it is tagged for transfer to Stack 11. When (and if) subroutine COLL has filled up Stack 11, it passes control to subroutine MCTOSN, which scores the volumetric source in the S_V region resulting from particles scattering out of the Monte Carlo region (reference Section 5.4). Subroutine MCTOSN returns control to COLL once it finishes execution.

If there are slots available for COLL to transfer its remaining particles to Stack 1, it does so, and then calls subroutine CALC (identical to subroutine CALC0) to calculate the remaining attributes required by subroutine TRCK. If Stack 1 is already being used by another subroutine, COLL instead transfers its particles to an individual stack reservoir as described above. Control is then relinquished to the subroutine which called COLL.

When subroutines XRW or YRW are executed and transfer their contents back into Stack 1, Stack 1 should contain 64 particles, since particles are not lost (hopefully) in an internal boundary crossing. This is not the case with subroutine COLL, where particle histories may be terminated through either a weight cutoff or by scattering into the S_N region. To ensure that only full stacks are executed, when possible, the driver routine transfers enough particles from Stack 0 to fill up Stack 1. If there are not enough particles remaining in Stack 0 to fill up Stack 1, the driver routine introduces another set of 64 particles into Stack 0. Other than this "topping off" of Stack 1, the driver routine only releases source particles into the system when all event stacks contain less than 64 particles, and all individual reservoir stacks have been emptied. This ensures that the capacity of the system will not be exceeded.

Once all the source particles have been exhausted, the driver routine proceeds to "flush out" the collision and internal boundary crossing stacks. The stack with the largest number of particles is executed first, then the next largest, etc., until all particles have been forced into a scoring stack. Note that the collision and internal boundary crossing stacks may have to be executed more than once apiece, since a particle can undergo more than one collision or boundary collision before scoring. The driver routine then concludes by cleaning up the scoring stacks, which, since they do not feed particles back into the vectorization scheme, need only be executed once apiece.

7.2 Comparison of Sequential and Vectorized Code Execution Times

Ideally, the vectorization of a code would result in a 64-fold increase in speed, since we can track 64 particles in the time it previously took to track just one. However, not all portions of a code are vectorizable, and vectorization itself is subject to a considerable amount of overhead (e.g., the movement of particles between stacks). Thus, a 64-fold increase in speed is not achievable in practice.

To measure the actual increase in speed, we compare two versions of the hybrid Monte Carlo/S_N code which are identical, except that one has had its Monte Carlo tracking routines vectorized, while the other retains sequential methods. We choose four representative sample problems for comparison, since the increases in speed will vary from problem to problem. All comparisons are performed on Machine 6, a Cray YMP, at Los Alamos, and all times are measured in CPU seconds. The results obtained with the sequential and vectorized versions agree within statistical uncertainties.

The first problem consists of a one group, homogeneous 11 by 11 cm block in $X-Y$ geometry, where the Monte Carlo region (including boundary layers) runs the entire width of the block between $y = 4$ and $y = 7$ cm. The fixed source consists of an isotropic distributed source with 1 cm dimensions at the center of the block. The (fictitous) cross sections are defined so that $\Sigma_T = 1$ cm, and $\Sigma_S = .95$ cm with isotropic scattering. The CPU times required to sample the fixed source (subroutine SRCMC) and the response matrix (subroutine RM) are shown at Table 7-2 for both the sequential and vectorized versions of the code, along with relative increase in speed. The identical number of histories were used in both versions (approximately

455,000 for the fixed source, and 1.3 million for the response matrix). Both the source and response matrix calculations show approximately a factor of four and a half increase in speed with the vectorized version.

Table 7-2
Sequential vs. Vector Comparison,
Problem # 1

Type	SRCMC	RM
SEQ	57.53 sec	114.7 sec
VEC	12.55 sec	24.16 sec
SEQ/VEC	4.6	4.7

The second sample problem is a three group, homogeneous 10 by 21 cm block in $X - Y$ geometry with isotropic scattering, where the Monte Carlo region (including boundary layers) runs the width of the block between $y = 9$ and $y = 12$ cm. The boundaries of the Monte Carlo region are group-independent, since $\Sigma_T = 1$ cm for all three groups. Downscattering only is allowed, with $\Sigma_{S,1-2} = \Sigma_{S,1-3} =$ .025 cm, $\Sigma_{S,2-3} = .05$ cm, and $\Sigma_{S,g-g} = .95$ cm. There is a small amount of absorption present in the third group, with $\Sigma_{A,3} = .05$ cm. The fixed source consists of an isotropic distributed source located along the width of the block between $y = 10$ and $y = 11$ cm. Approximately 710,000 histories were used to sample the fixed source, and 160,000 for each response matrix.

A comparison of the source and response matrix calculation times is shown below at Table 7-3. While the response matrices again show about a factor of four and a half increase in speed with the vectorized version, the source calculation's increase is "only" a factor of four. This reduction in efficiency is due to the presence of

two loops in subroutine COLL which could not be vectorized for multigroup problems. These two loops, of which one determines a scattered particle's new group, and the other the group a particle is to be scored within for the integral balance tables, require multiple passes for multigroup problems. Since the response matrix calculations do not use subroutine COLL, but rather subroutine RMCOLL, which requires group-within-group scattering only (reference Section 6.4), they are not subject to this fixed overhead and subsequent reduction in efficiency for multigroup problems. Note also that since the size of the Monte Carlo region is identical for each group, as are the response matrix cross sections, the calculation times are also almost identical betwen groups.

Table 7-3
Sequential vs. Vector Comparison.
Problem # 2

Type	SRCMC	RM1	RM2	RM3
SEQ	77.35 sec	13.91 sec	13.91 sec	13.96 sec
VEC	19.23 sec	3.05 sec	3.05 sec	3.05 sec
SEQ/VEC	4.0	4.6	4.6	4.6

The third sample problem is a one group, homogeneous cylinder with a radius and height of 11 cm, and a isotropic distributed source located between $r = 5$ to 6 cm, $z = 5$ to 6 cm. The Monte Carlo region consists of the area from $z = 4$ cm to $z = 7$ cm. Linearly anisotropic scattering is used, with $\Sigma_T = 1$ cm and $\Sigma_S = .95$ cm. Approximately 85,000 histories were used to sample the source, and 190,000 for the response matrix. The respective computation times for the sequential and vectorized codes are shown below at Table 7-4. For $R - Z$ geometry, the increase in speed with the vectorized version is between a factor of five and five and a half. This increase is greater than that for $X - Y$ geometry, since more calculations are

required when tracking a particle in cylindrical geometry, and thus more benefit is obtained from vectorization.

Table 7-4
Sequential vs. Vector Comparison,
Problem # 3

Type	SRCMC	RM
SEQ	12.5 sec	24.8 sec
VEC	2.29 sec	4.82 sec
SEQ/VEC	5.5	5.1

The final sample problem is a two group, heterogeneous cylinder with a radius of 10 cm and a height of 25 cm. The designated Monte Carlo region is composed of a strong absorber with cross sections $\Sigma_T = .1$ cm, $\Sigma_{S,1-1} = .005$ cm, $\Sigma_{S,1-2} = .005$ cm, $\Sigma_{S,2-2} = .01$ cm, and $\Sigma_A = .09$ cm. The boundary layer and S_N regions consist of a pure scattering region with $\Sigma_T = 1$ cm, $\Sigma_{S,1-1} = \Sigma_{S,1-2} = .5$ cm, and $\Sigma_{S,2-2} = 1$ cm. The Monte Carlo region (including boundary layers) is the area between $z = 10$ and $z = 15$ cm, and the fixed source consists of an isotropic point source located along the axis at $r = 12.5$ cm. Approximately 750.00 histories were used to sample the fixed source, and 265.000 for each response matrix. The increase in speed of the vectorized version over the sequential version (Table 7-5) is again between a factor of five and five and a half, with the source calculation being slightly less efficient since it is a multigroup problem. Since the total cross sections are group independent, the Monte Carlo region is also group independent, and the

response matrices require similar computational times.

Table 7-5
Sequential vs. Vector Comparison.
Problem # 4

Type	SRCMC	RM1	RM2
SEQ	71.78 sec	24.86 sec	13.96 sec
VEC	14.09 sec	4.54 sec	3.05 sec
SEQ/VEC	5.1	5.5	5.5

The percentage of the total computational time used in transferring particles between stacks (subroutine MOVXTOB) in the sample problems varied between 6% to 12%, with the other vectorized Monte Carlo routines (e.g., XSCORE, COLL, TRCK, etc.) using from less than 5% to almost 15% each, and totaling between 35% and 55% of the total execution time. About 5% to 10% of the time was spent in routines for intrinsic FORTRAN functions (square roots, logs, cosines, etc.), with another 5% to 10% used in specialized Cray functions (random number generators, vector mask functions, etc.). Of the remaining time, the "outer" Monte Carlo routines (SRCMC, MCRM) used less than 5%, while the S_N required between 8% and 17%.

While the proportion of execution time required for the S_N is highly dependent upon such factors as the relative sizes of the Monte Carlo and S_N regions, the S_N quadrature order, and the number of particle histories, it is clear from the above sample problems that, even with vectorization, the time required for the Monte Carlo calculations will generally outweigh that used in the S_N. Also, the overhead involved in vectorization, such as transferring particles between stacks and constructing particle destination words, is significant, and must be minimized

when developing a vectorization scheme as described above. When the FORTRAN equivalent of the subroutine MOVXTOB was used in place of the CAL utility routine, the Monte Carlo calculation times for the vectorized version approximately doubled.

CHAPTER 8.
CROSS SECTIONS AND THE HYBRID METHOD

The process of neutron interactions with nuclei in the medium of interest is quantified through the concept of cross sections, where the microscopic cross section σ is defined as the number of interactions per unit atom density per unit tracklength, and is commonly measured in units of barns, where 1 barn is equal to 10^{-24} cm^2. With this definition, the macroscopic cross section Σ as defined in Chapter 1 can be calculated from the relationship $\Sigma = N_1\sigma_1 + \ldots + N_J\sigma_J$, where N_j represents the atom density of the type jth nuclei in the medium. Microscopic neutron cross section data is obtained through both theoretical and experimental means, with the data being stored in computer-based files such as the Evaluated Nuclear Data File (ENDF). In essence, all the "physics" of the problem is contained in the set of cross sections used when solving the transport equation. Applying identical solution methods to identical problems with different cross section data will thus result in different solutions, since the physical basis of the problem has been altered.

Both discrete ordinates and Monte Carlo codes require some processing of the primary ENDF cross sections prior to their use, since the primary data is not in a format appropriate for use by either technique. The processing requirements vary for each technique, however, introducing differences into the cross section sets actually used. Thus, care must be used when comparing S_N and Monte Carlo results, since

even if they are based on the same set of primary cross section data, the data sets actually used in the codes differ due to the processing. More importantly (from our viewpoint), integration of the hybrid Monte Carlo algorithm with the S_N requires the generation of a set of Monte Carlo cross sections that is consistent with the cross sections used in the S_N.

8.1 Background

The ENDF files contain extensive amounts of data on neutron reactions for most isotopes, including cross section values for a large number of energies, angular distributions of scattered neutrons for both elastic and inelastic collisions, fission spectrums, etc. Although there are a large number of possible types of neutron reactions (reference Lamarsh,[24] Chapter 2), for our purposes we will consider the set of all possible interactions to be divided into just ten types:

a. Radiative capture (σ_γ),

b. Fission (σ_f),

c. Elastic scattering (σ_s),

d. Inelastic scattering (σ_i),

e. $(n,2n)$ and $(n,3n)$ interactions (σ_{2n} and σ_{3n}),

f. "Second chance" fissions ($\sigma_{i,f}$ and $\sigma_{2n,f}$), and

g. (n,p) and (n,α) reactions (σ_p and σ_α).

The use of cross sectional data by standard Monte Carlo methods is straightforward in principle.[18] Random numbers are drawn to determine the species of nucleus struck (if more than one isotope is present in a given region), and the type of resulting reaction. Typically, the processed data has been tabulated at sufficient energy values so that linear interpolation for non-tabulated energy values is within required accuracy bounds (i.e., "continuous-energy" data). The scattering angle for elastic and inelastic reactions is determined by sampling from 32 equally probable angular bins formed when the primary nuclear data is processed. The exiting neutron's energy for elastic and some inelastic scattering is then calculated from physical laws for conservation of energy and momentum, while energy distribution tables are provided for most inelastic scattering. The number of neutrons resulting from a fission reaction is determined by sampling a distribution function, while the direction and energy of emitted neutrons are sampled from fission angular distribution and fission spectrum tables, respectively.

The use of cross section sets in discrete ordinates codes is somewhat more convoluted, since the Boltzmann transport equation is formulated in terms of a particle balance, and energy is now a discrete variable (see Chapters 1 and 2). The primary data set is first processed to generate multigroup reaction cross sections by

$$\sigma_g = \int_{E_{g-1}}^{E_g} dE'\, g(E')\,\sigma(E'), \qquad (8-1)$$

where $g(E)$ represents the spectral weighting function discussed in Section 2.2. Scattering cross sections are also processed to generate scattering matrices for each

Legendre order (reference Sections 2.3 and 2.4), where

$$\sigma_{l,g\to g'} = 2\pi \int_{-1}^{+1} d\mu_0 \, P_l(\mu_0) \int_{E_{g'-1}}^{E'_g} dE' \int_{E_{g-1}}^{E_g} dE \, g(E)\, \sigma(E, \mu_0). \qquad (8-2)$$

Although we have now generated group reaction cross sections for each specific interaction, the S_N requires cross sections that provide overall interaction rates, i.e., total, scattering (elastic and inelastic), fission, and absorption. These S_N microscopic cross sections must be defined in terms of the group reaction cross sections generated by processing the primary data. These definitions, dropping the group (and Legendre order) indices for simplicity, are[25]

$$\sigma_T \equiv \sigma_\gamma + \sigma_s + \sigma_i + \sigma_f + \sigma_{2n} + \sigma_{3n} + \sigma_{i,f} + \sigma_{2n,f} + \sigma_\alpha + \sigma_p. \qquad (8-3a)$$

$$\sigma_S \equiv \sigma_s + \sigma_i + 2\sigma_{2n} + 3\sigma_{3n} + \sigma_{i,f} + 2\sigma_{2n,f}. \qquad (8-3b)$$

$$\sigma_F \equiv \sigma_f + \sigma_{i,f} + \sigma_{2n,f}. \qquad (8-3c)$$

and

$$\sigma_A \equiv \sigma_\gamma + \sigma_\alpha + \sigma_p + \sigma_f - \sigma_{2n} - 2\sigma_{3n} - \sigma_{2n,f}. \qquad (8-3d)$$

where σ_T, σ_S, σ_F, and σ_A are the total, scattering, fission, and absorption cross sections, respectively. The integer multiplier signs are used in Eqs. (8-3b) and (8-3d) since some reactions result in the creation, or loss, of more than one neutron, while the minus signs are required in Eq. (8-3d) in order to preserve the total particle balance. Note that although σ_A is not explicitly required for solution of the transport equation, it is needed when calculating integral balance tables, and that the integral absorption in a problem can be negative when materials with large (n,2n) cross sections (e.g., beryllium) are present. Once the S_N microscopic cross sections have been formed, the macroscopic cross sections actually used in the

S_N code are easily constructed as above, given the densities and locations of the isotopes comprising the problem.

Thus, even though the cross sections used by a Monte Carlo code and an S_N code may have been derived from the same primary set of data, they are no longer equivalent, but rather more like apples and oranges. The Monte Carlo cross sections reproduce each reaction type for each isotope, and are "continuous" in energy, while the S_N cross sections are in terms of overall reaction rates, represent a mixture of isotopes, and are averaged over an energy group. More importantly, angular distributions for scattering events in Monte Carlo codes are given in terms of tables of equiprobable cosine bins, while angular distributions in S_N codes are repsented by tables of Legendre moments. The effects on the hybrid method of this dichotomy in cross sections is the subject of the remainder of this chapter.

8.2 Hybrid Method Cross Sections

The hybrid Monte Carlo method has been implemented in the solver portion of the discrete ordinates code TWODANT, as described in Chapter 5. At this point in the code, the material types and densities specified in the input file have already been used to form macroscopic cross sections for each region in the problem. Thus, the cross sectional information available to the hybrid Monte Carlo method from the S_N consists of the total cross section ($\Sigma_{T,g}$), the absorption cross section ($\Sigma_{A,g}$), the average number of neutrons per fission times the total fission cross section ($\nu\Sigma_{F,g}$), the discrete fission spectrum (χ_g), and tables of Legendre scattering moments ($\Sigma_{l,g\to g'}$). The individual microscopic cross sections are no longer available

in the S_N solver, and the original angular distribution data for scattered particles was never available, since it was lost when the primary ENDF file was processed to generate the S_N cross sections. Thus, it is clearly impossible to employ standard Monte Carlo methods with the available S_N cross section data.

One obvious solution is to use a separate set of continuous-energy cross sections within the Monte Carlo region. This would enable the use of standard Monte Carlo methods, and presumably an increase in accuracy as well, since energy would now be treated in a more realistic (i.e., continuous) fashion. With appropriate definitions, neutron interactions could be grouped into the categories required for the S_N integral balance tables, and particles entering the S_N region from the Monte Carlo could easily be scored into the appropriate energy group. While the basis for formation of a probability density function (PDF) in energy for particles entering the Monte Carlo region from the S_N is not as clear-cut [e.g., uniform, $g(E)$, etc.], it would not appear to present insurmontable difficulties.

This approach was not selected for two reasons. First, it would require the user to undergo the not inconsequential effort of preparing two separate and distinct sets of cross sections for each problem, not to mention the additional code necessary for processing two cross section sets with different formats. Second, it would inhibit the benchmarking of the hybrid code. Since the hybrid method is implemented as an enhancement to the discrete ordinates code TWODANT, the simplest, and most effective, means of validating the hybrid code is to bench it against the S_N only version of TWODANT (reference Chapter 11). If the hybrid code used an additional set of cross sections entirely different from those used in the standard version of TWODANT, it would present another potential source of

difference between the discrete ordinates and the hybrid method, which might mask subtle errors in the hybrid code. Thus, it was decided to use the already available S_N cross sections in the Monte Carlo region of the hybrid code, at least as an initial effort.

Since we are employing implicit fission in the hybrid method (reference Section 6.5), we do not require explicit values of $\Sigma_{F,g}$, and the values of c', c'_{RM}, $P_{g-g'}$, $\Sigma'_{0,g-g'}$, and $P_{FN,g-g'}$ are readily computed from Eqs. (6-23) through (6-27). The determination of a scattered neutron's scattering angle presents difficulties, however.

The S_N furnishes tables of Legendre moments $\Sigma_{l,g-g'}$ for $l = 0, \ldots, L$, where L is the order at which the series representation is terminated. From Eq. (2-6), the distribution in angle μ_0 for neutrons scattering from group g to group g' is then (where we have included a factor of 2π in the definition of $\Sigma_{l,g-g'}$)

$$\Sigma_{L,g-g'}(\mu_0) = \frac{1}{2}\Sigma'_{0,g-g'} + \sum_{l=1}^{L} \frac{2l+1}{2}\Sigma_{l,g-g'} P_l(\mu_0). \qquad (8-4)$$

Thus, for $L > 0$, there is no guarantee that the scattering function $\Sigma_{g-g'}$ is non-negative, and so it does not constitute a PDF which may be directly sampled (reference Section 3.1).

It is possible to generate a PDF from the scattering function of Eq. (8-4) by employing absolute values with an appropriate normalization, i.e.,

$$f_{L,g-g'}(\mu_0) = |\Sigma_{L,g-g'}(\mu_0)| \Big/ \int_{-1}^{+1} d\mu' \, |\Sigma_{L,g-g'}(\mu')|. \qquad (8-5)$$

The scattering angle μ_0 is determined by sampling $f_{L,g\to g'}(\mu_0)$, while the particle weight is multiplied by the factor

$$\Sigma_{L,g\to g'}(\mu_0)/[f_{L,g\to g'}(\mu_0)\Sigma'_{0,g\to g'}] \qquad (8-6)$$

to conserve the total particle weight (on average).

However, this method has several disadvantages, the most important of which is the introduction of negative particle weights into the calculation.[26] Although we already allow the possibility of negative residuals, and subsequently negative particle weights, for multiple outer iteration problems as discussed in Section 5.7, the residuals are presumed to be small enough so that the overall boundary fluxes and volumetric sources will remain positive, and the increase in statistical errors due to the negative weights will be minimal. If we use Eqs. (8-5) and (8-6) to sample the distribution of scattered neutrons, however, not only will we have introduced the possibility of negative particle weights into the first outer iteration, but into the sampling of the fixed source as well. Thus, negative weight particles will have a place of much greater importance in the calculation, and the problems associated with their use will be correspondingly magnified. Therefore, an alternative approach, which we will refer to as the equiprobable bin (EPB) method, is selected.

We divide the scattering function $\Sigma_{L,g\to g'}(\mu_0)$ of Eq. (8-4) into 32 equiprobable bins $\Delta\mu_{0,i}$ by determining $\mu_{0i,L}$ and $\mu_{0i,R}$ such that

$$1/32 = \frac{1}{2\Sigma'_{0,g\to g'}}\int_{\mu_{0i,L}}^{\mu_{0i,R}} d\mu'\Sigma_{L,g\to g'}(\mu'), \qquad (8-7)$$

where $\mu_{0i,L}$ and $\mu_{0i,R}$ represent the angles corresponding to the left and right-hand edges of the ith bin, respectively, and $\Delta\mu_{0,i} = \mu_{0i,R} - \mu_{0i,L}$. To determine the

$\mu_{0i,L}$ and $\mu_{0i,R}$, we use the following iterative process. We assume that the bins are contiguous, so that for the ith bin, we set $\mu_{0i,L} = \mu_{0i-1,R}$, with $\mu_{01,L} = -1$. We know that $\mu_{0i,R}$ must lie within the range $\mu_L = \mu_{0i,L}$ to $\mu_R = +1$, so we estimate $\mu_{0i,R}$ by $\tilde{\mu} = (\mu_R + \mu_L)/2$. We then begin an iterative process on $\tilde{\mu}$ by computing the value of the integral

$$\tilde{A} = \frac{1}{2\Sigma'_{0,g\to g'}} \int_{\tilde{\mu}}^{+1} d\mu' \, \Sigma_{L,g\to g'}(\mu') \qquad (8-8)$$

via a simple recursive procedure,[27] and comparing it with the exact value if $\tilde{\mu} = \mu_{0i,R}$,

$$A = \frac{32 - i}{32}. \qquad (8-9)$$

If $\tilde{A} > A$, we set $\mu_L = \tilde{\mu}$, otherwise we set $\mu_R = \tilde{\mu}$. We then re-estimate $\tilde{\mu}$ from the new values of μ_L and μ_R, and perform another iteration. The iterative process is terminated after 25 iterations, ensuring an absolute error in the placement of $\mu_{0i,R}$ of less than 10^{-7}.

8.3 Benchmark Problems for Hybrid Method Cross Sections

Now that we have developed an algorithm for constructing Monte Carlo cross sections from the available S_N cross sections, we examine it for possible discrepancies or limitations in actual use. To do this, we choose four sample problems and compare the results obtained with the hybrid method to those obtained with pure discrete ordinates. The geometry for all four problems is identical, consisting of a homogeneous four by five cm cylinder with an isotropic distributed source located between $z = 2$ to $z = 3$ cm, and extending from $r = 0$ to $r = 1$ cm. The coarse mesh

size in both the S_N and hybrid method geometries is set at 1 cm by 1 cm. with a varying number of fine mesh cells per coarse mesh used in the S_N. The boundary layers for the designated Monte Carlo region are defined so as to encompass the entire problem; thus, the hybrid Monte Carlo method is used throughout the entire problem geometry, maximizing any differences with the S_N.

The first benchmark is a one group problem with a parabolic scattering function defined by $f(\mu_0) = 3\mu_0^2/2$, so that $f_L(\mu_0)$ represents $f(\mu_0)$ exactly for $L = 2$ [reference Eq. (8-5)]. We let $\Sigma_T = 1.0$ cm and $\Sigma_A = .05$ cm, so that $\Sigma_{0,1\to 1} = .95$ cm, and $\Sigma_{2,1\to 1} = .38$ cm. Table 8-1 presents a comparison of the integral self-scattering (SS) and net leakage (NL) results between the hybrid Monte Carlo (HMC) and S_N methods, along with the associated absolute error (one standard deviation) in the hybrid Monte Carlo self-scattering. The S_N entries reflect the number of fine meshes per coarse mesh, which was varied to examine the effects of mesh sizing on the S_N, as was the quadrature order. Both S_N and hybrid Monte Carlo runs were performed with isotropic scattering (P_0) for comparison with the P_2 results. The total execution time (in CPU seconds) is also shown for all runs. Note that 250,000 histories were used to sample the fixed source in both hybrid Monte Carlo runs, and that since the Monte Carlo region comprises the entire problem geometry, no response matrix or boundary flux sampling was required. The exact scattering function $f(\mu_0)$ and the equiprobable bin (EPB) approximation to it for $L = 2$ are

shown at Fig. 8-1.

Table 8-1
Cross Section Benchmark Problem #1

Type	P_L	SS	NL	Time
$S_{20}/8 \times 8$	P_0	6.582	.6536	7.3 sec
HMC	P_0	$6.576 \pm .009$	.6539	31.0 sec
$S_6/2 \times 2$	P_2	6.701	.6473	2.5 sec
$S_{20}/2 \times 2$	P_2	6.729	.6459	4.3 sec
$S_{20}/4 \times 4$	P_2	6.666	.6492	6.3 sec
$S_{20}/8 \times 8$	P_2	6.647	.6501	11.7 sec
HMC	P_2	$6.641 \pm .007$	.6505	31.5 sec

From Table 8-1, it can be seen that the use of the EPB method with hybrid Monte Carlo cross sections results in excellent agreement with the S_N, especially for small S_N mesh sizes and large quadrature orders, which better approximate the Monte Carlo's continuous treatment of spatial and angular phase coordinates. Although the differences in integral results between the P_0 and P_2 representations are small, as expected due to the parabolic (even) shape of the scattering function, they are real, and show up quite clearly in both the S_N and hybrid Monte Carlo calculations. It is interesting to note that the calculation times for the P_0 and P_2 hybrid Monte Carlo runs are almost identical, as expected for the EPB method, while the comparable S_N runs show over a 60% increase in execution time for $L = 2$ over $L = 0$, due to the increased number of moments.

The second benchmark problem is identical to the first, except that the parabolic scattering function is replaced with the step function $f(\mu_0) = \Theta(\mu_0)$, where Θ represents the Heaviside step function. For this function, a Legendre expansion requires $L = \infty$ for an exact representation. Figures 8-2 and 8-3 present

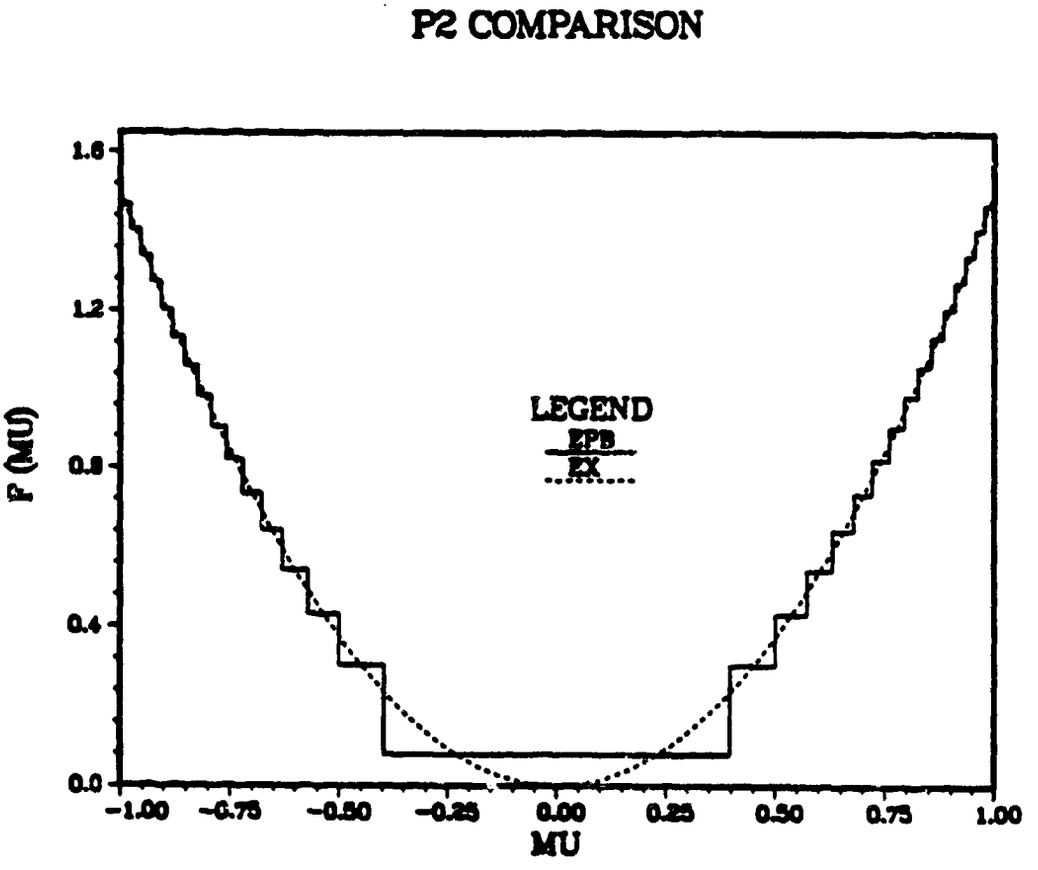

Figure 8-1 Parabolic Scattering Function, $L = 2$

comparisons of the exact step function (EX) with P_1 and P_5 approximations, and the EPB methods's representations of the Legendre approximations. The first four non-zero moments of the scattering function are $\Sigma_{0,1\to 1} = .95$ cm, $\Sigma_{1,1\to 1} = .475$ cm, $\Sigma_{3,1\to 1} = -.11875$ cm, and $\Sigma_{5,1\to 1} = .059375$ cm. Results for the discrete ordinates and hybrid Monte Carlo methods are shown below at Table 8-2. For the final hybrid Monte Carlo run listed, the EPB scattering function was replaced with the exact scattering function. Although the hybrid Monte Carlo results using the exact scattering function are in good agreement with the S_{20} results, even for S_N values of $L = 1$, the hybrid Monte Carlo fails to reproduce the correct integral values when using the EPB approximation of the scattering function, even with $L = 5$.

Table 8-2
Cross Section Benchmark Problem #2

Type	P_L	SS	NL	Time
$S_6/8 \times 8$	P_1	4.880	.7432	5.7 sec
$S_{20}/8 \times 8$	P_1	4.868	.7438	16.8 sec
HMC	P_1	$4.974 \pm .006$	.7382	23.6 sec
$S_{20}/8 \times 8$	P_3	4.866	.7439	18.9 sec
HMC	P_3	$4.933 \pm .005$	.7404	22.5 sec
$S_{20}/8 \times 8$	P_5	4.866	.7439	22.9 sec
HMC	P_5	$4.932 \pm .007$	.7404	22.5 sec
HMC	EX	$4.856 \pm .006$	.7444	22.1 sec

To provide a more realistic test, the remaining two benchmark problems use a set of sixteen-group Hansen-Roach cross sections. The Hansen-Roach cross section set provides multigroup cross sections for energies ranging from 10 MeV down to the thermal range ($\sim$.025 eV), and includes transport-corrected[1] values for σ_T, σ_A, $\nu\sigma_F$, and $\sigma_{0,g\to g'}$ (downscattering and self-scattering only). In addition, tables of

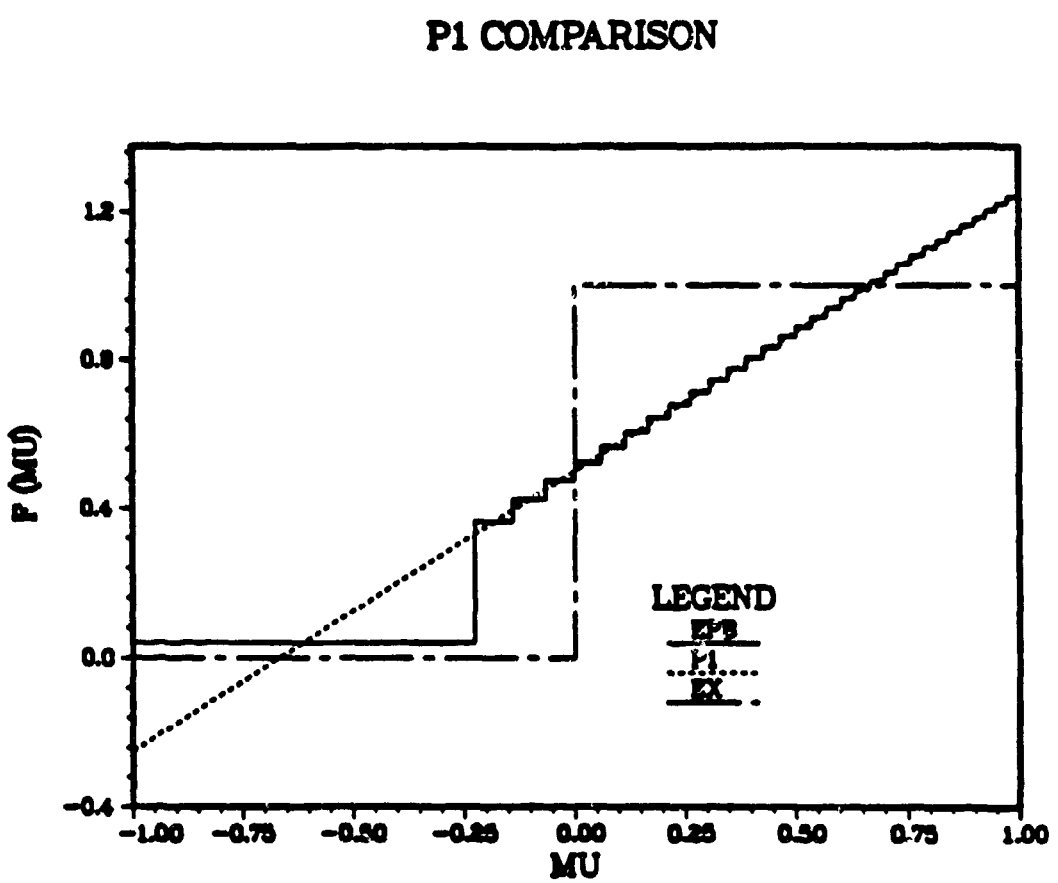

Figure 8-2 Step Scattering Function, $L = 1$

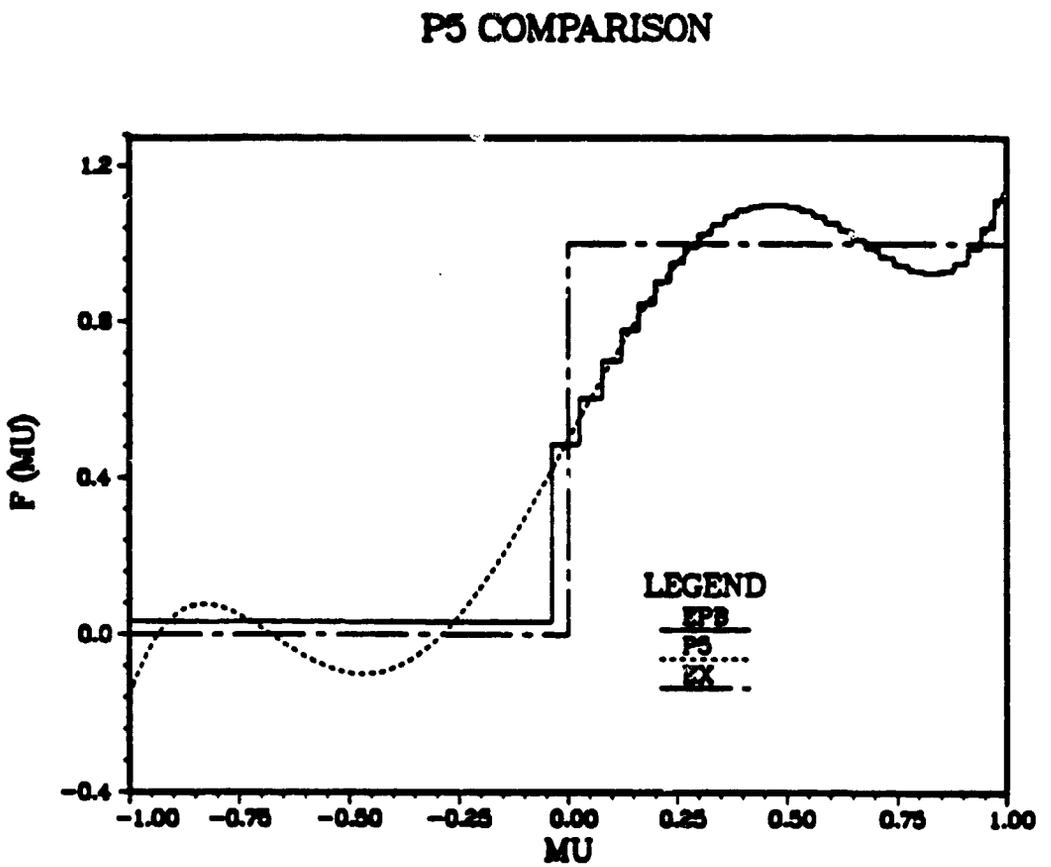

Figure 8-3 Step Scattering Function, $L = 5$

$\sigma_{1,g\to g'}$ are provided for hydrogen and deuterium. The problem geometry remains identical to the previous problems, with the fixed source located in the first group.

For the third benchmark problem, the cylinder material is composed of a mixture of O^{16} (.16 atoms/b-cm) and hydrogen (.02 atoms/b-cm). Once again, the S_N mesh size and quadrature order was varied, and the problem was run with both $L = 0$ (isotropic) and $L = 1$ (linearly anisotropic) scattering. From Table 8-3 below, we see that that the hybrid Monte Carlo method is in good agreement with S_N for both isotropic and linearly anisotropic scattering.

Table 8-3
Cross Section Benchmark Problem #3

Type	P_L	SS	NL	Time
$S_6/2 \times 2$	P_0	1.002	.9776	3.8 sec
$S_{20}/2 \times 2$	P_0	1.002	.9774	14.8 sec
$S_{20}/4 \times 4$	P_0	1.000	.9775	29.4 sec
$S_{20}/8 \times 8$	P_0	0.9986	.9775	64.6 sec
HMC	P_0	0.9989 ± .0030	.9775	14.6 sec
$S_6/2 \times 2$	P_1	0.9328	.9776	3.9 sec
$S_{20}, 2 \times 2$	P_1	0.9326	.9774	16.9 sec
$S_{20}/4 \times 4$	P_1	0.9306	.9775	39.9 sec
$S_{20}/8 \times 8$	P_1	0.9291	.9775	75.3 sec
HMC	P_1	0.9296 ± .0026	.9775	13.9 sec

Finally, to examine the effects of increasing the $L = 1$ moment, the composition of the fourth benchmark problem consists of hydrogen only, with a density of .18 atoms/b-cm. Although excellent agreement is still obtained for $L = 0$, as can be seen from Table 8-4 below, the hybrid Monte Carlo and S_N results do not agree for $L = 1$, with the divergence clearly being outside the limits of reasonable statistical

deviation. (The last entry will be discussed subsequently.)

Table 8-4
Cross Section Benchmark Problem #4

Type	P_L	SS	NL	Time
$S_6/2 \times 2$	P_0	31.12	.8026	3.4 sec
$S_{20}/2 \times 2$	P_0	31.34	.8042	12.8 sec
$S_{20}/4 \times 4$	P_0	30.77	.8049	22.6 sec
$S_{20}/8 \times 8$	P_0	30.72	.8052	44.56 sec
HMC	P_0	$30.77 \pm .08$	.8049	304.1 sec
$S_6/2 \times 2$	P_1	13.32	.9167	4.0 sec
$S_{20}/2 \times 2$	P_1	13.04	.9185	19.0 sec
$S_{20}/4 \times 4$	P_1	13.01	.9186	39.0 sec
$S_{20}/8 \times 8$	P_1	13.00	.9187	88.7 sec
HMC	P_1	$13.75 \pm .07$	.9012	145.0 sec
HMC	P_1^*	$13.77 \pm .10$	.9139	109.9 sec

8.4 Discussion

The hybrid Monte Carlo method, when using multigroup cross sections based on the available S_N cross sections, provides results in agreement with the discrete ordinates method, for the selected benchmark problems, when isotropic scattering is used. However, the EPB method of forming a PDF from the existing Legendre moments fails to accurately reproduce the angular distributions of scattered neutrons for $L > 0$ in some cases, and in these cases the hybrid Monte Carlo method results and the S_N results diverge. Specifically, the EPB method appears to fail when $f_L(\mu_0)$ is not positive definite, as is the case for cross sections with large P_1 components. This failure of the EPB method is exacerbated since the S_N cross

sections are in the laboratory frame of reference, so that elements such as hydrogen, which are generally isotropic in the center-of-mass frame of reference, appear as strongly anisotropic.

The failure of the EPB method is due to the requirement that the entire scattering range ($\mu_0 = -1$ to $+1$) be divided into contigous, non-negative bins. Thus, a particle has a finite probability of scattering into any given angle in range -1 to $+1$ for all events $g \rightarrow g'$. This violates physical laws of conservation of momentum and energy, since, depending upon the value of g', some angles of μ_0 are impossible.

Mathematically, we state that the EPB method fails because it does not conserve the moments of the scattering cross section, other than the zeroth moment, while, in fact, conservation of at least the first several moments is important for an accurate representation of the scattering function.[26] The S_N cross sections do conserve moments, and thus its succes in approximating the step function of the second benchmark problem with just two moments ($L = 1$). The EPB method fails to provide accurate results for the same problem, however, even with $L = 5$, since its approximation of the S_N scattering function $\Sigma_{L,g\rightarrow g'}(\mu_0)$ does not conserve the moments of $\Sigma_{L,g\rightarrow g'}(\mu_0)$.

The EPB method does not fail in all cases, just those in which the scattering function $\Sigma_{L,g\rightarrow g'}(\mu_0)$ is negative for some range of μ_0. For isotropic scattering, or for cross sections in which the scattering function is anisotropic but positive definite, as in the first and third benchmark problems, the EPB method will provide a good

representation of $\Sigma_{L,g\to g'}(\mu_0)$, and thus implicitly conserve moments, at least approximately. However, this severely limits the problems to which the EPB method, and hence the hybrid Monte Carlo/S_N method, can sucessfully be applied, since most multigroup cross section sets cannot be guaranteed to be positive definite for a finite order L. Since it is impossible to resconstruct the (presumably) positive definite original scattering function $\Sigma_{g\to g'}(\mu_0)$ from the finite order Legendre representation $\Sigma_{L,g\to g'}(\mu_0)$, and we do not wish to sample $\Sigma_{L,g\to g'}(\mu_0)$ directly because it might entail a large number of histories with negative particle weights, we are left with two alternatives.

The first is, of course, to use continuous-energy cross sections in the Monte Carlo region, as described above. Although this has the disadvantages of requiring two sets of cross sections for every calculation, and integration of continuous-energy Monte Carlo data with multigroup S_N data, as well as requiring an additional calculation in the collision subroutine (conversion of center-of-mass scattering coordinates to the laboratory frame), the approach is reliable and well established. Furthermore, the EPB method can be retained as an available code option for benchmarking isotropic or weakly anisotropic scattering problems.

The second is to use a method which forms a PDF from the Legendre series representation which conserves the polynomial moments $\int_{-1}^{+1} d\mu\, \mu^l\, \sigma(\mu)$, at least approximately. Two examples of such a method are the discrete angle representation, and the equally probable step function representation, which are described in Ref. 26. Although each representation has its advantages and disadvantages, we note that the processing required to generate the discrete angles or equally probable

step functions is not negligible, but is instead probably comparable to that of using continuous-energy cross sections.

As an example of the effect of conserving moments, we replace the EPB method of generating bins with a derivative of the equally probable step function for the fourth benchmark problem (Table 8–4), where we now generate a single bin with one boundary fixed at -1 or $+1$, depending upon the sign of $\sigma_{1,g\rightarrow g'}$, and the other boundary chosen at the value of μ which preserves the first moment. The results of this method are shown at the last entry in Table 8–4, and are clearly more accurate than those of the EPB method.

In conclusion, the EPB method is sufficient for the benchmarking of the hybrid Monte Carlo/S_N code, and the investigation of problems with simplified scattering functions (i.e., isotropic scattering, or scattering with small anisotropic components), but is not robust enough for general use. For general use of the hybrid Monte Carlo/S_N method, consideration should be given towards use of one of the alternative approaches discussed above.

CHAPTER 9.
VARIANCE MEASURES AND REDUCTION

As discussed in Section 3.2, information obtained using Monte Carlo methods has an associated uncertainty (variance). Most Monte Carlo codes provide a variance estimate for all calculated results, as well as numerous methods of reducing the variance. In addition, a figure of merit (FOM), as defined at Chapter 3, is also provided for selected results as a measure of the reliability of the variance estimates.

With standard Monte Carlo methods, one runs a single calculation of K batches of N histories each, from which an estimate of the variance of the results is directly obtained from Eq. (3–10). Thus, only a single Monte Carlo calculation is required to determine the variance. However, with the hybrid method, such a straightforward evaluation of the variance is not possible due to the coupling of the Monte Carlo and S_N regions, as can be seen by the following argument.

With the hybrid method, we first sample the fixed source using Monte Carlo, thus obtaining the boundary flux $\mathbf{S}^{out}$ at the Monte Carlo/S_N interface, of which each element has an associated variance $\sigma^2_{S,i}$, which is clearly obtainable. Next, we calculate the incoming boundary flux for iteration 1 from Eq. (4–2) (assuming a one-group problem) by

$$\Psi^{in(1)} = \underline{\mathbf{T}}\,\mathbf{S}^{out}, \qquad (9-1)$$

where $\underline{\mathbf{T}}$ represents an S_N calculation. Although we do not calculate an S_N response matrix, for reasons discussed in Chapter 4, consider for the moment that $\underline{\mathbf{T}}$ is represented by such a matrix, where $T_{i,j}$ corresponds to the angular flux entering the Monte Carlo region in state i due to a unit incident flux upon the S_N region in state j. The incoming boundary flux for state i is then determined from

$$\psi_i^{in(1)} = \sum_{j=1}^{N} T_{i,j} S_j^{out}. \qquad (9-2)$$

where N is the number of states along the Monte Carlo/S_N interface, and S_j^{out} is the jth element of $\mathbf{S}^{out}$. Given the $\sigma_{S,j}^2$, the variance in $\psi_i^{in(1)}$ is[28]

$$\sigma_{in,i}^2 = \sum_{j=1}^{N} T_{i,j}^2 \sigma_{S,j}^2 + \sum_{j}{}' \sum_{k} T_{i,j} T_{i,k} \mathrm{Cov}[S_j^{out}, S_k^{out}]. \qquad (9-3)$$

where the prime upon the summation indicates that the summation is over the values $j \neq k$, and Cov represents the covariance of S_j^{out} and S_k^{out}. The covariance is required because the individual elements of $\mathbf{S}^{out}$ are clearly not independent. Since the covariance of $\mathbf{S}^{out}$ is not known, nor easily determined, the direct calculation of the variance in the Monte Carlo/S_N boundary fluxes is not feasible.

An alternative solution is to simply run K sets of the entire problem, thus providing K independent sets of results for both the boundary fluxes, and the information most likely desired by the user, the cell fluxes. While, in theory, this is entirely analogous to the Monte Carlo procedure of running K batches of histories, the costs are quite different. For Monte Carlo calculations, the time required to run K batches of N histories per batch is approximately the same as the time required to run one batch of $N \times K$ histories. However, with S_N, the time required per calculation is fixed, so that running K sets of a calculation would increase the S_N

calculation time by a factor of K. Since the minimum value of K needed to get a valid statistical sample is on the order of 20, the time spent in S_N computations would be similarly increased by a factor of at least 20. Even though the time spent in S_N calculations does not usually constitute a majority of the CPU time, it is still significant, and an increase of this magnitude is not desirable. In addition, since the number of histories used in the Monte Carlo calculation for each set would now be N, instead of NK, each S_N calculation would now be based on a reduced number of Monte Carlo histories. This could result in increased problems with negative flux fixup in the S_N, and negative residuals at the Monte Carlo/S_N interface. Thus, we choose to implement a less rigorous measure of the variance in the hybrid method.

9.1 Variance Estimation in the Hybrid Method

Currently, the user has two separate inputs into the number of histories run in the Monte Carlo portions of the hybrid code, the number of histories used in sampling the fixed source, and the number of histories per column used in sampling the response matrix. The number of histories used in sampling the incoming boundary fluxes and any volumetric sources located in the Monte Carlo regions are determined from the above values (reference Sections 6.8 and 6.9). Thus, what is needed is not just some measure of the variance of the results, but the relative merits of increasing the number of histories used in sampling the response matrix, versus the number of histories used with the fixed source.

To acquire some measure of the variance in the fixed source calculation, we use the standard technique of dividing the number of histories into K batches

of N_S histories each. For each batch, the tracklength for each coarse cell in the Monte Carlo region is calculated, as well as the integral self-scattering. Once the sampling of the fixed source has been finished, the variance of the tracklengths is calculated from Eqs. (3-7), (3-8), and (3-10). In addition, the FOM for the integral self-scattering is printed out for each batch as a measure of the reliability of the variance estimation.

While the variance in $\mathbf{S}^{out}$ is not directly calculated, measuring the variance of the coarse meshes due to the fixed source does indicate how well the various cells of the Monte Carlo region are being sampled during the fixed source calculation. More importantly, it is anticipated that oftentimes one of the cells in the Monte Carlo region will represent a physical detector for which the flux is desired, in which case some knowledge of the variance in that cell is important.

We do not attempt to directly measure the variance of the individual elements of the response matrix, both because the variance of the individual elements themselves is of little importance, and because storage of the $\sigma^2_{R,ij}$ would approximately double the memory requirements. Instead, when sampling the converged incoming boundary flux $\Psi_g^{in(\infty)}$ during the first outer iteration (reference Section 5.5), we recalculate the outgoing boundary flux, denoted by $\tilde{\Psi}_g^{out}$. We then perform an $L2$ error comparison between the two outgoing boundary fluxes, that is, we calculate the relative error

$$R_{ERR} = ||\tilde{\Psi}_g^{out} - \Psi_g^{out(\infty)}|| \,/\, ||\tilde{\Psi}_g^{out} - \mathbf{S}_g^{out}||. \qquad (9-4)$$

As the number of histories used to sample the reponse matrix goes to infinity, that is, as $N_{RM} \rightarrow \infty$, we expect $R_{ERR} \rightarrow 0$. For a finite N_{RM}, R_{ERR} gives a better

"feel" of the variance due to the response matrix than the individual $\sigma^2_{R,ij}$, since the importance of a given column in the response matrix may be zero, if the the value of the incoming flux for the associated S_N state is also zero.

While the variance measures described above are not rigorous in the sense that they do not provide a variance estimate for each cell, they do provide a means of determining how well the fixed source calculation samples the individual cells in the Monte Carlo region, and how precise the response matrix calculation is. Furthermore, by examining the variances in the fixed source calculation, the size of R_{ERR}, and the relative leakages from the Monte Carlo region due to $\mathbf{S}^{out}$ and Ψ^{out}, one has some idea if N_{RM}, N_S, or both should be increased.

9.2 Variance Results in Sample Problems

We now apply the variance estimation methods described above to a one group, sample problem consisting of a homogeneous, 4 by 5 cm cylinder with Σ_T = 1.0 cm and Σ_S = 0.95 cm. The fixed source consists of an isotropic distributed source located radially between $r = 0$ to $r = 1$ cm, with axial limits $z = 2$ to $z = 3$ cm. The problem geometry is divided into coarse mesh cells with 1 cm by 1 cm dimensions.

We begin by letting the Monte Carlo region comprise the entire problem geometry, so that we may examine the effects on the variance estimates of varying N_S. We perform three groups of problems, each with $K = 200$, but with N_S values of 81, 625, and 2401. Each group consists of three separate runs, where each run

has a different initial random seed. Table 9–1 lists the number of histories per batch (N_S), the integral self-scattering (SS) with the relative error (one standard deviation), the difference (in standard deviations) from the S_{20} benchmark, and the resulting FOM, while Table 9–2 presents the coarse mesh fluxes (multiplied by a factor of 10, with a relative error of one standard deviation) for the center cells (z = 2.5 cm, r = .5, 1.5, 2.5, and 3.5 cm). Table 9–3 shows the differences (in standard deviations) in the center coarse mesh cell fluxes between the Monte Carlo runs and the S_{20} benchmark. Each table identifies the run by the initial random seed. For comparison, the results of an S_{20} calculation with 64 fine meshes (8×8) per coarse mesh (CM) are also included in Tables 9–1 and 9–2.

Table 9–1
Integral SS and FOM Results, Monte Carlo Results

Seed	N_S	SS	Diff	FOM
240	81	6.5865±.0044	+0.17	15.485
250	81	6.5666±.0046	−0.49	14.642
260	81	6.5252±.0045	−1.92	15.114
220	625	6.5557±.0016	−2.46	21.711
230	625	6.5778±.0017	−0.33	20.322
270	625	6.5783±.0017	−0.29	21.806
200	2401	6.5745±.0008	−1.33	23.995
210	2401	6.5699±.0008	−2.21	24.467
280	2401	6.5787±.0008	−0.53	22.317
S_{20}	-	6.5815	-	-

Table 9-2

Center Coarse Mesh Cell Fluxes, Monte Carlo Results

Seed	CM#1	CM#2	CM#3	CM#4
240	10.262±.0054	8.8777±.0079	4.7323±.0120	2.2489± .0164
250	10.177±.0057	8.8137±.0075	4.7442±.0121	2.3278± .0170
260	10.104±.0057	8.7082±.0078	4.6864±.0121	2.2610± .0182
220	10.178±.0020	8.8447±.0029	4.6412±.0043	2.2784± .0057
230	10.179±.0020	8.8162±.0029	4.6747±.0044	2.2927± .0060
270	10.180±.0021	8.8351±.0028	4.6853±.0041	2.3004± .0060
200	10.197±.0011	8.8202±.0015	4.6837±.0023	2.2875± .0032
210	10.186±.0011	8.8008±.0013	4.6864±.0022	2.2948± .0031
280	10.181±.0010	8.7946±.0014	4.6794±.0022	2.2943± .0030
S_{20}	10.180	8.8231	4.6810	2.2878

Table 9-3

Center Coarse Mesh Cell Differences, Monte Carlo Results

Seed	N_S	CM#1	CM#2	CM#3	CM#4
240	81	+1.48	+0.78	+0.90	−1.05
250	81	−0.05	−0.14	+1.10	+1.01
260	81	−1.32	−1.69	+0.10	−0.65
220	625	−0.10	+0.84	−1.95	−0.72
230	625	−0.05	−0.27	−0.31	+0.36
270	625	0.00	+0.49	+0.22	+0.91
200	2401	+1.52	−0.22	+0.25	−0.04
210	2401	+0.54	−1.95	+0.52	+0.98
280	2401	+0.10	−2.31	−0.16	+0.94

Examining Table 9-1, we see that the integral self-scattering results from the Monte Carlo runs, although appearing to be be systematically smaller than the S_{20} benchmark value, are within one to two standard deviations. Note that the error (standard deviation) behaves approximately as the $\sqrt{N}$. The overall Figures of Merit appear to be consistent within a given group (i.e., the same value of N_S), while between groups the FOM increases with increasing N_S. This is due to the vectorization

of the Monte Carlo, where particles are followed in groups of 64 particles until all source particles have been exhausted, at which time the remaining particles are "flushed" from the system (reference Section 7.1). Since it requires approximately the same time to execute a stack with 64 particles as to execute a stack with one, the runs with smaller values of N_S require proportionally more time to flush particles from their system, reducing their FOM.

The Figures of Merit for each run as a function of batch number are shown at Fig. 9-1 for $N_S = 81$, Fig. 9-2 for $N_S = 625$, and Fig. 9-3 for $N_S = 2401$, where each curve corresponds to a differing initial random seed. Although the Figures of Merit for a specified value of N_S eventually settle upon approximately the same value, as expected, it takes a fairly large number of batches (50-100). This implies a large variance of the variance, which could be reduced by decreasing the batch size, since the variance of the estimated variance is minimized with a batch size of one.[29,30] Indeed, this can be seen in Figs. 9-1 through 9-3, where the behavior of the FOM's for $N_S = 81$ is much steadier than that for an equivalent number of histories with $N_S = 625$ (26 batches) and $N_S = 2401$ (7 batches). However, decreasing the batch size also reduces the efficiency of the vectorization, as previously stated, and as is apparent in the respective FOM's for the differing batch sizes.

In examining the coarse mesh cell fluxes (Tables 9-2 and 9-3), we see that the Monte Carlo fluxes are within one to two standard deviations of the S_{20} benchmark fluxes, and that the error in the fluxes behaves approximately as the $\sqrt{N}$. Note also that the error in the fluxes increases with radius, since the source is located at the centerline cell. Thus, the cells closest to the centerline have more particle

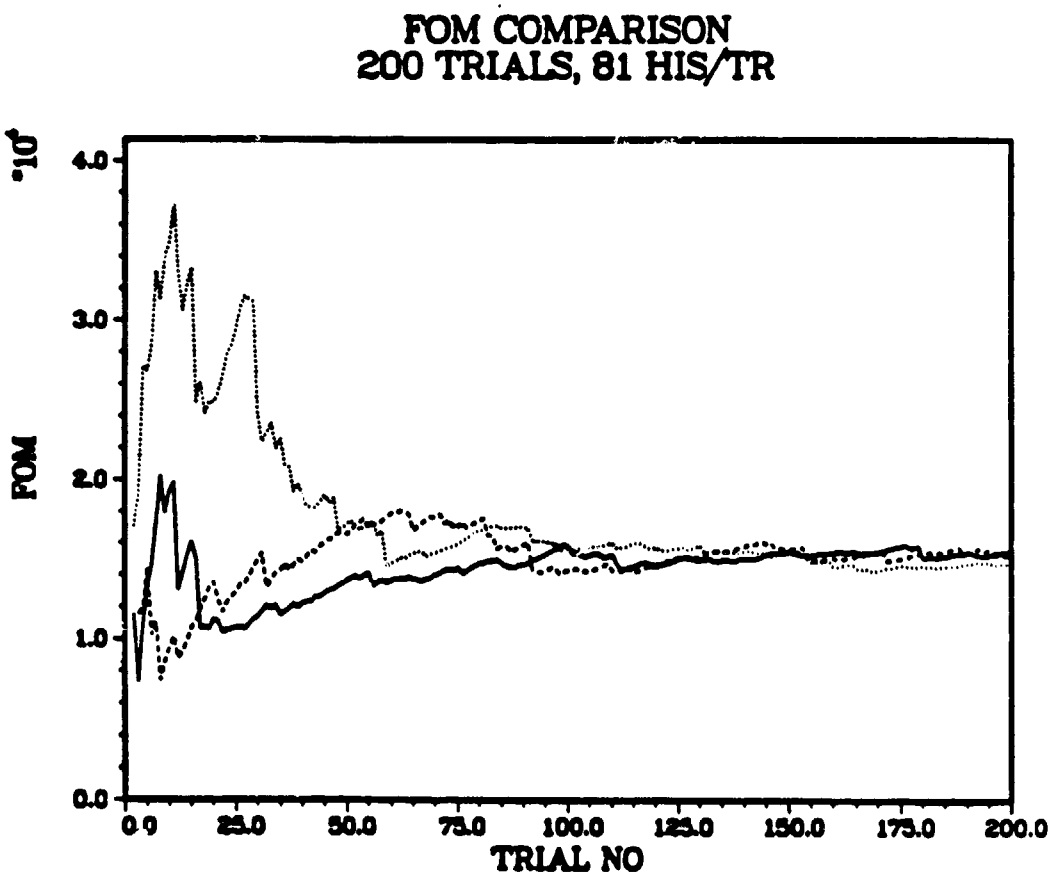

Figure 9-1 Figure of Merit Comparison, $N_S = 81$ Particles/Batch

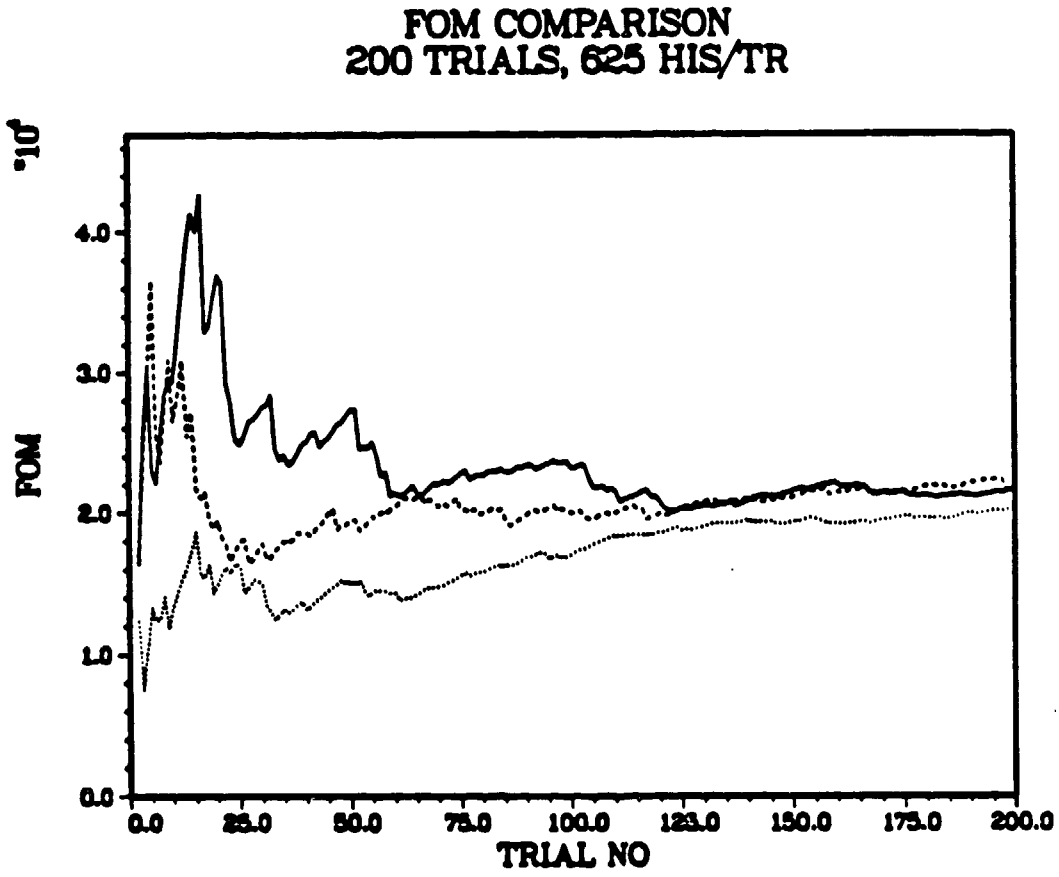

Figure 9–2 Figure of Merit Comparison, $N_S = 625$ Particles/Batch

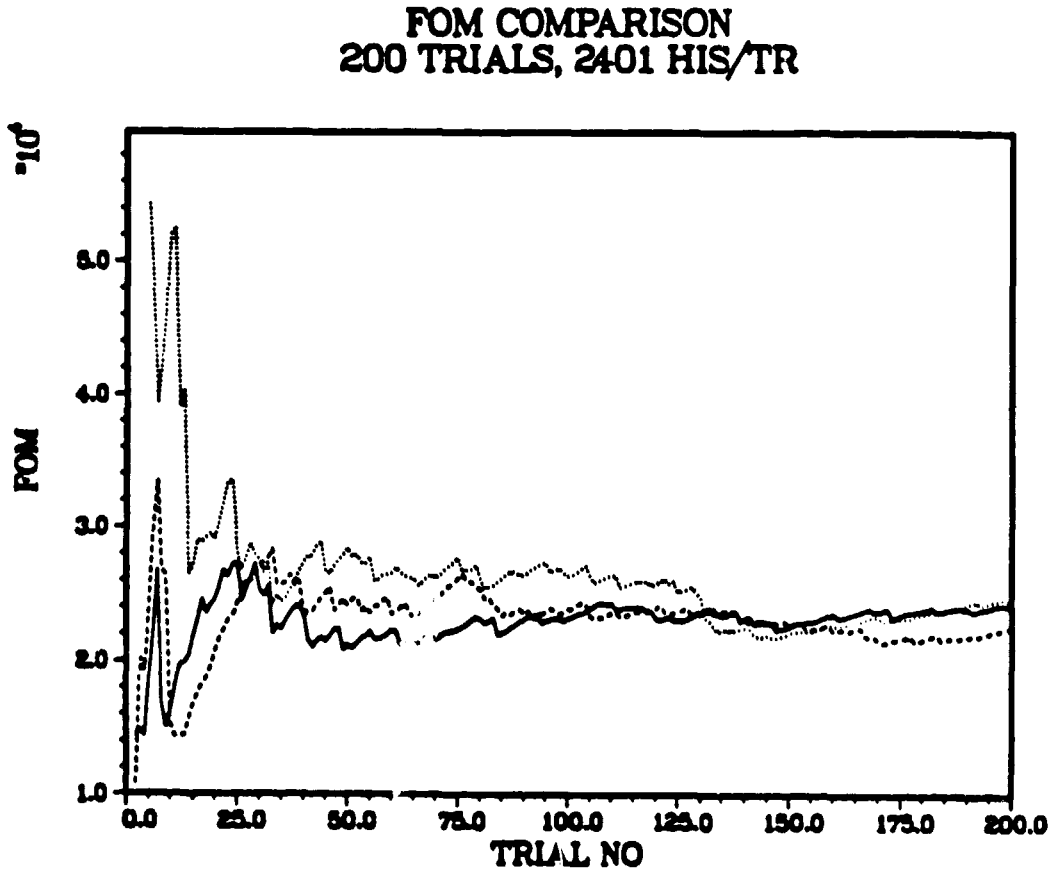

Figure 9-3 Figure of Merit Comparison, N_S = 2401 Particles/Batch

tracks than those nearer to the outer edge, and as a result their fluxes have a smaller variance.

We now examine a problem where the geometry is composed of both Monte Carlo and S_N regions. We use the same problem geometry as before, except that the region $r = 0$ to $r = 1$ cm, extending the length of the cylinder, is designated as the Monte Carlo region. A one mean-free-path boundary layer is specified, so the total Monte Carlo region extends out to $r = 2$ cm. An S_6 solver is used in the S_N region, with 4 fine meshes (2×2) per coarse mesh. For the Monte Carlo calculation, we fix K at 200, while N_S and N_{RM} are varied. To obtain estimates of the coarse mesh cell variances, we perform 15 runs, with differing initial seeds, for each combination of N_{RM} and N_S. Table 9–4 below shows the resulting average value of R_{ERR} and associated relative error (one standard deviation) for each combination of N_{RM} and N_S, along with the average value of the absolute particle balance (PBAL). Tables 9–5a and 9–5b list the average coarse mesh cell fluxes (multiplied by a factor of 10), with relative errors, for the center cells, along with the flux values in the Monte Carlo region due to the fixed source alone (i.e., the flux values for $N_{RM} = 0$. Note that the first two cells are in the Monte Carlo region, while the last two are located in the S_N. For comparison, Table 9–5c lists the coarse mesh cell fluxes calculated from an S_{20} solution with 64 fine meshes per coarse mesh, and an S_6 solution with 16 fine meshes per coarse mesh (8×2) in the first two meshes, and four fine meshes (2×2) in the last two. Finally, Tables 9–6 and 9–7 contain the differences, in terms of standard deviations, between the coarse mesh cell fluxes calculated by the hybrid

method and those calculated by the pure S_N solutions.

Table 9-4

Average R_{ERR} and Particle Balance Values, Hybrid Results

N_S/N_{RM}	R_{ERR}	PBAL
81/16	.0612 ± .0932	9.3526×10^{-4} ± .7062
81/81	.0227 ± .0936	4.8715×10^{-4} ± .8421
81/256	.0103 ± .0703	2.2609×10^{-4} ± .6687
625/16	.0605 ± .0954	11.7599×10^{-4} ± .6758
625/81	.0229 ± .0655	5.8763×10^{-4} ± .7344
625/256	.0098 ± .0522	1.6055×10^{-4} ± .8932

Table 9-5a

Average Coarse Mesh Cell Fluxes, $N_S = 81$

N_{RM}	CM#1	CM#2	CM#3	CM#4
0	9.760±.0054	7.2171±.0076	-	-
16	10.169±.0060	8.8104±.0105	4.7937±.0118	2.2667±.0112
81	10.172±.0055	8.7787±.0067	4.8120±.0092	2.2728±.0072
256	10.168±.0058	8.7830±.0069	4.8173±.0079	2.2765±.0071

Table 9-5b

Average Coarse Mesh Cell Fluxes, $N_S = 625$

N_{RM}	CM#1	CM#2	CM#3	CM#4
0	9.760±.0019	7.1664±.0028	-	-
16	10.184±.0038	8.8005±.0073	4.8215±.0106	2.2690±.0104
81	10.181±.0028	8.8080±.0037	4.8240±.0055	2.2774±.0043
256	10.182±.0019	8.7922±.0028	4.8196±.0036	2.2741±.0030

Table 9-5c

Coarse Mesh Cell Fluxes, S_N Results

N_{RM}	CM#1	CM#2	CM#3	CM#4
S_6	10.137	8.7526	4.7636	2.2743
S_{20}	10.180	8.8231	4.6810	2.2878

Table 9-6
Coarse Mesh Cell Differences, Hybrid Results vs. S_6

N_S/N_{RM}	CM#1	CM#2	CM#3	CM#4
81/16	+0.52	+0.62	+0.53	−0.30
81/81	+0.63	+0.44	+1.09	−0.09
81/256	+0.53	+0.50	+1.42	+0.14
625/16	+1.21	+0.75	+1.13	−0.22
625/81	+1.54	+1.70	+2.28	+0.32
625/256	+2.33	+1.61	+3.23	−0.03

Table 9-7
Coarse Mesh Cell Differences, Hybrid Results vs. S_{20}

N_S/N_{RM}	CM#1	CM#2	CM#3	CM#4
81/16	−0.18	−0.14	+1.99	−0.82
81/81	−0.14	−0.76	+2.96	−0.92
81/256	−0.20	−0.66	+3.60	−0.70
625/16	+0.10	−0.35	+2.75	−0.80
625/81	+0.04	−0.46	+5.39	−1.06
625/256	+0.10	−1.26	+7.99	−2.01

From Table 9-4, we see that the decrease in R_{ERR} is roughly proportional to $\sqrt{N_{RM}}$, as expected, and that R_{ERR} is a good predictor of the precision of the response matrix, as indicated by its small relative error values. We note that the average particle balance also decreases approximately as the $\sqrt{N_{RM}}$, but that the relative error in the particle balance is much larger, so it cannot be relied upon as an indication of the response matrix accuracy. Particle balance is affected by the value of N_{RM} through the response matrix calculation and the sampling of the incoming boundary fluxes. Using $\underline{\mathbf{R}}$, we iterate to find the $\Psi^{out(\infty)}$ and $\Psi^{in(\infty)}$. We then sample the $\Psi^{in(\infty)}$ and determine $\tilde{\Psi}_g^{out}$. Since $\Psi^{out(\infty)} \neq \tilde{\Psi}_g^{out}$, there will be a discrepancy in the particle balance, because the particle balance in the S_N region is

based upon $\Psi^{out(\infty)}$, while that of the Monte Carlo region is based upon $\tilde{\Psi}_g^{out}$. As $N_{RM} \rightarrow \infty$, $\tilde{\Psi}_g^{out} \rightarrow \Psi^{out(\infty)}$, and the discrepancy in the particle balance decreases.

From Tables 9-5 through 9-7, it is clear that the behavior of the coarse mesh cell variances is quite different in the hybrid case than that of the pure Monte Carlo case. Examining the first coarse mesh cell, we see that the variance is almost completely independent of the value of N_{RM}. Since the source is located in this cell, almost all of the flux is due to particles emanating directly from the source, with very little attributable to particles which reenter the Monte Carlo region from the S_N. In fact, if we examine the values in Tables 9-5a and 9-5b for $N_{RM} = 0$, we see that over 95% of the flux in the first cell is directly due to the calculation of $\mathbf{S}^{out}$, so that the error in this cell is insensitive to the value of N_{RM}. For the second coarse mesh cell, a value of $N_{RM} = 16$ results in an error somewhat greater than that due solely to $\mathbf{S}^{out}$, while values of $N_{RM} = 81$ and 256 give errors approximately equal to that of the fixed source calculation. Since the second coarse mesh is adjacent to the S_N region, it sees a higher flux from the S_N region, as can be seen in Tables 9-5a and 9-5b, and thus is more strongly affected by a poorly defined response matrix ($N_{RM} = 16$) than the first coarse mesh is.

The third and fourth coarse meshes are in the S_N region, and at a greater distance from the fixed source, so their variances are more strongly influenced by the response matrix and S_N transport operator. Comparing Tables 9-2 and 9-5, we see that while in the pure Monte Carlo problem the variance increases monotonically with distance from the fixed source, this is not the case in the hybrid

problem. There, the variance in the S_N region tends to remain constant, or decrease somewhat, as the distance from the Monte Carlo/S_N interface decreases. As N_{RM} increases, the error initially decreases, then remains constant.

Consider that we have some variance σ^2_{out} in $\Psi^{out(\infty)}$, due to both the sampling of the fixed source and the response matrix. As we increase N_{RM}, we decrease the response matrix component of σ^2_{out}, but we are still left with the component due to the sampling of the fixed source. Thus, as $N_{RM} \rightarrow \infty$, σ^2_{out} goes to a constant, but non-zero, value. Consider also that the S_N transport operator is a deterministic operator, and as such tends to average out the statistical fluctations in the problem. Thus, as we proceed away from the Monte Carlo/S_N interface into the S_N, the variance tends towards a constant value. However, the sampling of the fixed source introduces a certain inherent variance into the problem, which no amount of increase in N_{RM}, or distance from the Monte Carlo/S_N interface, can reduce.

Examining Tables 9–6 and 9–7, we see that the coarse mesh fluxes calculated by the hybrid method are generally within one to two standard deviations of both the S_6 and S_{20} results, with the exception of the third coarse mesh cell. In the third cell, there appears to be a definite, systematic difference with the S_N results. However, this is not completely unexpected, as the third mesh cell lies just beyond the border of the S_N/Monte Carlo interface. Thus, while the fluxes in the first two mesh cells are determined principally by the Monte Carlo method, those in the last two are determined by the S_N method, using an S_6 quadrature order with 4 fine meshes (2×2) per coarse mesh. In contrast, Table 9–7 is based on an S_{20} operator with 64 fine meshes per coarse mesh, so it has a much more detailed fine mesh structure in the last two coarse mesh cells than the hybrid method, as well

as a higher quadrature order. When we attempt to approximate the hybrid mesh structure by using 16 fine meshes per coarse mesh in the first two coarse meshes, and four fine meshes per coarse mesh in the last two (Table 9-6), we see that we do obtain a somewhat better agreement between the fluxes in the third and fourth coarse mesh cells.

This illustrates an important point to be considered when employing the hybrid method. While the the fluxes in the Monte Carlo region are effectively determined by an S_∞ quadrature order with a fine mesh structure of size zero (neglecting the effects of any coupling with the S_N region), the fluxes in the S_N are determined by a low quadrature order with a finite mesh size. Thus, when we employ the hybrid method in order to extend the benefits of a more accurate Monte Carlo solution in one region to a lower order S_N solution in another, we also, to some extent, extend the less accurate S_N solution into the Monte Carlo problem. The importance of this effect will depend upon the amount of coupling between the Monte Carlo and S_N regions, and the accuracy of the S_N solution. However, since we place the S_N/Monte Carlo interface at least one mean-free-path from the physical material interface, presumably placing it in an area where a low S_N quadrature order solution is suffucient, the effects should be minimal.

As a final sample problem, we use the same cross sections and definition of the Monte Carlo region as above, but enlarge the S_N region by increasing the radius of the cylinder to 10 cm. With $N_S = 625$, $K = 200$, and $N_{RM} = 256$, we once again perform 15 separate runs with different initial random seeds. In comparison with Table 9-4, the average particle balance was $2.2847 \times 10^{-4} \pm .5952$, while the average difference in the computed outgoing boundary fluxes was $R_{ERR} = .0099 \pm .0806$.

The average coarse mesh cell fluxes (multiplied by a factor of 10) and relative errors (one standard deviation) are presented below at Table 9-8, along with the fluxes from an S_6 calculation with 4 fine meshes per coarse mesh, and an S_{20} calculation with 64 fine meshes per coarse mesh. We see that the error in the fluxes in the S_N region does remain approximately constant, and that the hybrid flux values generally fall between those of a pure S_6 and S_{20} calculation, with the exception of the third mesh cell.

Table 9-8
Coarse Mesh Cell Fluxes, 5 Cm by 10 Cm Cylinder

CM#	MC/S_6	S_6	S_{20}
1	10.220 ± .0023	10.231	10.219
2	8.9503 ± .0025	9.1288	8.9625
3	5.1411 ± .0030	4.9251	5.0062
4	3.0017 ± .0026	2.9493	3.0096
5	1.7890 ± .0024	1.7689	1.3281
6	1.0652 ± .0026	1.0540	1.0219
7	0.6272 ± .0025	0.6217	0.6401
8	0.3621 ± .0022	0.3589	0.3685
9	0.2000 ± .0018	0.1982	0.2030
10	0.0932 ± .0022	0.0924	0.0944

9.3 Variance Reduction Measures Employed in the Hybrid Monte Carlo Method

As discussed in Chapters 3 and 6, the Monte Carlo used in the hybrid method is almost entirely analog, with few variance reduction techniques currently employed. However, this is not as disadvantageous as it may first appear, as can be seen by the following arguments. Aside from the reasons previously mentioned in

Chapter 6 (vectorization and designation of S_N regions), all particle histories followed by the hybrid Monte Carlo method result in a score, where a score is defined as exiting the Monte Carlo region, since implicit capture is used. The only exception to this is in the sampling of the incoming boundary fluxes in subroutine DWNSRC, where particles which enter and remain in the same group for their entire history are not scored in $\mathbf{Q}_g^{out}$. However, at least for the first outer iteration, even these histories are not entirely wasted, since they are used in the computation of R_{ERR}.

While almost all particle histories do result in a score, this is not sufficient to ensure that all areas of the problem geometry are thoroughly sampled. One means of increasing the sampling rate of a particular area, aside from increasing the overall number of histories, is through biasing in direction, space, or energy, so that more particles are directed into the required area. At present, the hybrid Monte Carlo method does not include any such methods, although they could be included in the future, perhaps at the expense of some interference with the vectorization.

The hybrid Monte Carlo method does use stratified sampling,[16] however, in which the possible initial spatial and angular coordinates of a particle are subdivided, with equal numbers of particles forced to start within each subdivision. Stratified sampling is used when sampling the fixed source, the response matrices, and the incoming boundary fluxes, but not the S_N volumetric sources located within the Monte Carlo region.

Assume that we have m independent spatial and angular coordinates in phase space, and that we wish to run N histories in a given Monte Carlo calculation. For an isotropic point source, the spatial coordinates are fixed, so the value of m is two,

while for an isotropically distributed source, $m = 4$. When sampling a column of the response matrix, we have one independent spatial and two angular variables, so $m = 3$. Given m, we then let $N_m = \text{Int}(N^{1/m})$, and divide each independent phase space variable into N_m equal intervals. Thus, we have now subdivided our phase space into N_m^m "boxes". By forcing one particle to start somewhere within each "box", with the specific location determined by a random number, we can ensure a more uniform sampling of the initial phase space coordinates. Note that for the sampling of the fixed source and response matrix, the actual number of histories run will be N_m^m, not the specified value of $N_S K$ or N_{RM}. When sampling the incoming boundary fluxes, we run an additional $N - N_m^m$ histories without st atified sampling, after first runnning N_m^m stratified histories, in order to maintain the proper weighting between the individual states.

When using stratified sampling with the fixed source, we stratify the angular coordinates by $\Delta o_m = \pi/N_m$ and $\Delta\eta_m = 2/N_m$, except for the isotropic surface flux source, where $\Delta\mu_m = 1/N_m$ (reference Section 6.6), and the beam source, where the starting angular coordinates are fixed. The spatial coordinates are sampled, where appropriate, using "boxes" of $\Delta x_m = \Delta x/N_m$ and $\Delta y_m = \Delta y/N_m$ for $X - Y$ geometry, and $\Delta r_m^2 = \Delta r^2/N_m$ and $\Delta y_m = \Delta z/N_m$ for $R - Z$ geometry. The spatial coordinates are sampled in a similar manner for the incoming boundary fluxes and the response matrix, and the angular bin $\Delta\Omega_j$ is subdivided into "boxes" of $\Delta\phi_m = \Delta\phi/N_m$ and $\Delta\eta_m = \Delta\eta/N_m$.

Essentially, stratified sampling is "free", since the determination of a particle's initial coordinates in phase space is performed outside of the tracking routines, and is easily vectorized. Thus, the amount of CPU time required is negligible, and

stratification is not expected to adversely affect the FOM under any circumstances. To see how much of a gain we achieve, if any, we examine our sample problem (4 by 5 cm cylinder) again from two different perspectives, one in which stratified sampling is employed in sampling the fixed source, and one in which it is not. For simplicity, the Monte Carlo region is defined as the entire problem region, with $N_S = 2401$ and $K = 200$. The results, in terms of integral self-scattering and FOM, are shown below at Tables 9–9 and 9–10 for three runs apiece, each with a different initial random seed. The average FOM for the runs without stratified sampling was 25.165, while that for the runs using stratified sampling was 28.144, an increase of 12%. Thus, the use of stratified sampling results in a clear, if modest, increase in the FOM. (We note that the FOM's in Table 9–10 are slightly higher than those for the equiavlent runs in Table 9–1 because the runs in Table 9–10 were performed with a later version of the code, containing improved vectorization of some loops.)

Table 9–9
Sample Problem Without Stratification

Seed	SS	FOM
0	6.5805 ± .0009	25.162
10	6.5771 ± .0009	24.472
20	6.5640 ± .0008	25.860

Table 9–10
Sample Problem With Stratification

Seed	SS	FOM
30	6.5765 ± .0008	26.677
40	6.5699 ± .0008	29.448
50	6.5704 ± .0008	28.306

Another applicable variance reduction method is the use of forced collisions. With this method, the total optical distance to the nearest boundary of the Monte Carlo region is determined, and the probability of a particle reaching that boundary without a collision is calculated. The particle is then split into two separate particles with appropriate weights, one of which reaches the boundary without a collision, and thus scores, and the other which has a collision at some point, determined by random sampling, along the flight path. Particle histories are now terminated by the weight cutoff procedure described in Section 6.4, since otherwwise they would continue indefinitely. Each particle history now provides several scores, since it will undergo multiple collsions (and subsequent weight reductions) before being terminated. Additionally, the use of forced collisions increases the probability of a particle scattering into another group during the sampling of the incoming boundary fluxes, which in turn increases the number of scores in $\mathbf{Q}_{g'}^{out}$.

However, the implementation of forced collisions is not "free", unlike the use of stratified sampling, since it substantially increases the amount of computational time required to track a particle. This increase occurs beacuse we now must trace the flight path of a particle from its collision point to the nearest boundary of the Monte Carlo region, not just the nearest boundary of its current coarse mesh cell as described in Chapter 6. Since the flight path to the nearest boundary may cross several coarse mesh cells, the relevant calculations for the distance to the nearest cell boundary are correspondingly multiplied. In addition, the calculations of probability of survival to the nearest boundary and the collision point along the flight path entail the use of exponnential and natural log functions,[13] which are in themselves expensive.

Some preliminary studies on employing forced collisions with the hybrid Monte Carlo/S_N method have been performed during the course of this work. Clearly, the use of forced collsions as a variance reduction method in the hybrid method is feasible, and offers the possibility of a reduction in the FOM. However, the preliminary results indicate that the additional computational time required precludes any decrease in the FOM, unless a more efficient means of implemetation is used. Such an implementation would require a redesign of the vectorization scheme described in Chapter 7, in order to allow particles to be followed across multiple coarse mesh cells. Since this would require a not insubstantial amount of effort, and the use of forced collisions is not central to the hybrid Monte Carlo/S_N problem, the implementation of forced collions has been left as a future problem.

CHAPTER 10.
DIFFUSION SYNTHETIC ACCELERATION

Consider the discretized representation of the discrete-ordinates form of the transport equation, where we have assumed a one-group, one-dimensional Cartesian geometry with isotropic scattering and source for simplicity. In this case, the inner iteration [Eqs. (2-22) and (2-23)] becomes

$$\mu_n(o^{k+1/2}_{n,i+1/2} - o^{k+1/2}_{n,i-1/2}) + \Sigma_{T,i}\, o^{k+1/2}_{n,i}\Delta x_i = \Sigma_{S,i}\, \tilde{o}^{k}_{0,i}\Delta x_i + Q_i \Delta x_i. \qquad (10-1)$$

where

$$\tilde{o}^{k+1}_{0,i} = \sum_{n=1}^{N} w_n\, o^{k+1/2}_{n,i}. \qquad (10-2)$$

Larsen[31] has shown via Fourier stability analysis that the source iterative method [Eqs. (10-1) and (10-2)], while stable for all mesh sizes Δx_i, has a spectral radius of $\rho = \Sigma_S/\Sigma_T$, where ρ represents the slowest possible reduction in error from one iteration to the next. Thus, for optically thick, highly scattering regions, the source iterative method will have a slow rate of convergence. Since these are the very types of regions we desire to use S_N in with the hybrid method, we require some means of accelerating the rate of convergence.

The Fourier stability analysis by Larsen also shows that the most slowly converging modes of the angular flux are those that are nearly linear functions of μ. Thus, if we could generate the exact scalar flux $\tilde{o}_{0,i}$ in one iteration, given an

angular flux $\phi_i(\mu)$ which is linear in μ, we will have devised an effective acceleration method for the source iterative method. Since the assumption that the angular flux is linear in μ corresponds to the P_1, or diffusion theory, approximation of the transport equation,[1] the essential idea behind diffusion synthetic acceleration is to replace Eq. (10-2) with a diffusion-like solution for $\tilde{\phi}_{0,i}^{k+1}$. We use the term diffusion-like, since the diffusion equation itself is often a poor approximation to the transport equation, so an appropriate correction term must be included.

The diffusion synthetic acceleration (DSA) equation for Eq. (10-1), i.e., the equation for $\tilde{\phi}_{0,i}^{k+1}$ that replaces Eq. (10-2), is

$$\frac{-1}{3\Sigma_{T,i+1}\,\Delta x_{i+1}}(\tilde{\phi}_{0,i+3/2}^{k+1}-\tilde{\phi}_{0,i+1/2}^{k+1})+\frac{1}{3\Sigma_{T,i}\,\Delta x_i}(\tilde{\phi}_{0,i+1/2}^{k+1}-\tilde{\phi}_{0,i-1/2}^{k+1})$$
$$+\frac{1}{4}\Sigma_{R,i+1}\,\Delta x_{i+1}(\tilde{\phi}_{0,i+3/2}^{k+1}+\tilde{\phi}_{0,i+1/2}^{k+1})+\frac{1}{4}\Sigma_{R,i}\,\Delta x_i(\tilde{\phi}_{0,i+1/2}^{k+1}+\tilde{\phi}_{0,i-1/2}^{k+1})=$$
$$\frac{1}{2}(Q_{i+1}\,\Delta x_{i+1}+Q_i\,\Delta x_i)-R, \tag{10-3}$$

where the correction term R is

$$R=\frac{1}{3\Sigma_{T,i+1}\,\Delta x_{i+1}}(\tilde{\phi}_{0,i+3/2}^{k+1/2}-\tilde{\phi}_{0,i+1/2}^{k+1/2})-\frac{1}{3\Sigma_{T,i}\,\Delta x_i}(\tilde{\phi}_{0,i+1/2}^{k+1/2}-\tilde{\phi}_{0,i-1/2}^{k+1/2})$$
$$+\tilde{\phi}_{1,i+1}^{k+1/2}-\tilde{\phi}_{1,i}^{k+1/2}. \tag{10-4}$$

$\Sigma_{R,i}$ is defined as the removal cross section $\Sigma_{T,i}-\Sigma_{S,i}$, and

$$\tilde{\phi}_{l,i}^{k+1/2}=\sum_{n=1}^{N} w_n\,P_l(\mu_n)\,\phi_{n,i}^{k+1/2}. \tag{10-5}$$

Eqs. (10-3) and (10-4) are obtained from Eq. (10-1) by taking the zeroth and first moments, as defined at Eq. (10-5), then defining acceleration equations by

$\tilde{o}_{0,i}^{k+1/2} \rightarrow \tilde{o}_{0,i}^{k+1}$ and $\tilde{o}_{1,i}^{k+1/2} \rightarrow \tilde{o}_{1,i}^{k+1}$, and eliminating $\tilde{o}_{2,i}^{k+1/2}$ in terms of $\tilde{o}_{0,i}^{k+1/2}$ and $\tilde{o}_{1,i}^{k+1/2}$. Note also that the diamond difference relationship

$$o_{n,i} = \frac{1}{2}(o_{n,i+1/2} + o_{n,i-1/2}) \qquad (10-6)$$

is used.

To construct boundary conditions[32] for the left-hand edge, consider that we have an incident boundary flux $J_{n,1/2}$ ($\mu_n > 0$) which is linear in angle, i.e.,

$$J_{n,1/2} = \frac{1}{2}\tilde{o}_{0,1/2}^{k+1} + \frac{3}{2}\mu_n \tilde{o}_{1,1/2}^{k+1}. \qquad (10-7)$$

from which we obtain, by multiplying by μ_n and summing over $\mu_n > 0$.

$$\tilde{J}_{1,1/2}^{+} = \frac{1}{4}\tilde{o}_{0,1/2}^{k+1} + \frac{1}{2}\tilde{o}_{1,1/2}^{k+1}. \qquad (10-8)$$

For vacuum boundary conditions, $\tilde{J}_{1,1/2}^{+} = 0$, and we have the constraint $\tilde{o}_{1,1/2}^{k+1} = -\tilde{o}_{0,1/2}^{k+1}/2$. Thus, the DSA equation at the left boundary becomes

$$\frac{-1}{3\Sigma_{T,1}\,\Delta x_1}(\tilde{o}_{0,3/2}^{k+1} - \tilde{o}_{0,1/2}^{k+1}) + \frac{1}{4}\Sigma_{R,1}\,\Delta x_1(\tilde{o}_{0,3/2}^{k+1} + \tilde{o}_{0,1/2}^{k+1}) + \frac{1}{2}\tilde{o}_{0,1/2}^{k+1} = \frac{1}{2}Q_1\,\Delta x_1 - R'. \qquad (10-9)$$

where the modified correction term R' is

$$R' = \frac{1}{3\Sigma_{T,1}\,\Delta x_1}(\tilde{o}_{0,3/2}^{k+1/2} - \tilde{o}_{0,1/2}^{k+1/2}) + \tilde{o}_{1,1}^{k+1/2} - \tilde{o}_{1,1/2}^{k+1/2} - \frac{1}{2}\tilde{o}_{0,1/2}^{k+1/2}. \qquad (10-10)$$

Similar conditions can be derived for the right boundary.

The method of diffusion synthetic acceleration was first succesfully implemented by Alcouffe,[33] who determined that stability was dependent upon a consistent spatial differencing scheme between the transport and diffusion-like equations.

Later, Larsen[31] showed that the use of diffusion synthetic acceleration results in a reduction of the spectral radius to $\rho = .2247\,\Sigma_S/\Sigma_T$. Diffusion synthetic acceleration schemes have been developed for two-dimensional geometries, including spherical and cylindrical, as well as the outer iteration,[13,33] and are fully implemented into TWODANT.[34]

10.1 Incorporating DSA with the Hybrid Method

As expained in Section 5.6, the hybrid method is implemented in the code TWODANT through the inclusion of interior boundary sources, so that the sweeping algorithms are unaffected. However, the inclusion of interior boundary sources does affect the operation of the DSA, either reducing its effectiveness, or eliminating it entirely for some problems.

This loss of effectiveness is essentially due to the introduction of a singularity in the transport equation. Consider Fig. 10-1, which represents a one-dimensional Monte Carlo/S_N hybrid problem. The interface boundary is located at $x_{J+1/2}$, with the Monte Carlo region consisting of the left-hand cells, and includes some fixed source Q, while the S_N region consists of the remaining right-hand cells. In the hybrid method, Eq. (10-1) is essentially replaced by

$$\begin{aligned}\mu_n(\phi^{k+1/2}_{n,i+1/2} - \phi^{k+1/2}_{n,i-1/2}) + \Sigma_{T,i}\,\phi^{k+1/2}_{n,i}\,\Delta x_i = \Sigma_{S,i}\,\tilde{\phi}^{k}_{0,i}\,\Delta x_i \\ + \mu_n(B_{n,J+1/2} - \phi^{k+1/2}_{n,i-1/2})\,\delta_{i,J+1}, \qquad \mu_n > 0,\end{aligned} \tag{10-11}$$

where $\delta_{i,J+1}$ represents the delta Kroniker function, and $B_{n,J+1/2}$ is the outgoing boundary flux determined by Eq. (4-11). Note that sweeps from right to left ($\mu_n <$

0) are unchanged, in this case, and that the fluxes for cells in the Monte Carlo region are determined separately from Eq. (10-1) (reference Section 6.3).

Our goal is to derive, starting from Eq. (10-11), an equivalent form of the acceleration equations of (10-3) and (10-4). We begin by replacing the interior boundary source with an inhomogeneous source located in cell J, where the inhomogeneous source is defined so that it gives the identical boundary flux at $x_{J+1/2}$ for each iteration $k+1/2$. Thus, we define

$$Q_{n,J}^{k+1/2}\Delta x_J = B_{n,J+1/2}(\mu_n + \frac{1}{2}\Sigma_{T,J}\Delta x_J) - o_{n,J-1/2}^{k+1/2}(\mu_n - \frac{1}{2}\Sigma_{T,J}\Delta x_J). \qquad \mu_n > 0. \tag{10-12a}$$

and

$$Q_{n,J}^{k+1/2}\Delta x_J = 0. \qquad \mu_n < 0. \tag{10-12b}$$

where we have set $\Sigma_{S,J} = 0$. Equation (10-11) then becomes

$$\mu_n(o_{n,i+1/2}^{k+1/2} - o_{n,i-1/2}^{k+1/2}) + \Sigma_{T,i}o_{n,i}^{k+1/2}\Delta x_i = \Sigma_{S,i}\tilde{o}_{0,i}^{k}\Delta x_i + Q_{n,J}^{k+1/2}\Delta x_J\delta_{i,J}. \tag{10-13}$$

Taking the zeroth and first moments of Eq. (10-13), we obtain

$$\tilde{\phi}_{1,i+1/2}^{k+1/2} - \tilde{\phi}_{1,i-1/2}^{k+1/2} + \Sigma_{T,i}\tilde{\phi}_{0,i}^{k+1/2}\Delta x_i = \Sigma_{S,i}\tilde{o}_{0,i}^{k}\Delta x_i + \tilde{Q}_{0,J}^{k+1/2}\Delta x_J\delta_{i,J} \tag{10-14}$$

and

$$\frac{2}{3}(\tilde{\phi}_{2,i+1/2}^{k+1/2} - \tilde{o}_{2,i-1/2}^{k+1/2}) + \frac{1}{3}(\tilde{\phi}_{0,i+1/2}^{k+1/2} - \tilde{\phi}_{0,i-1/2}^{k+1/2}) + \Sigma_{T,i}\tilde{o}_{1,i}^{k+1/2}\Delta x_i = \tilde{Q}_{1,J}^{k+1/2}\Delta x_J\delta_{i,J}. \tag{10-15}$$

where the moments of $Q_{n,J}^{k+1/2}$ are

$$\tilde{Q}_{0,J}^{k+1/2} = \tilde{B}_{1,J+1/2}^{+} - \tilde{\phi}_{1,J-1/2}^{+,k+1/2} + \frac{1}{2}\Sigma_{T,J}\Delta x_J(\tilde{B}_{0,J+1/2}^{+} + \tilde{o}_{0,J-1/2}^{+,k+1/2}) \tag{10-16a}$$

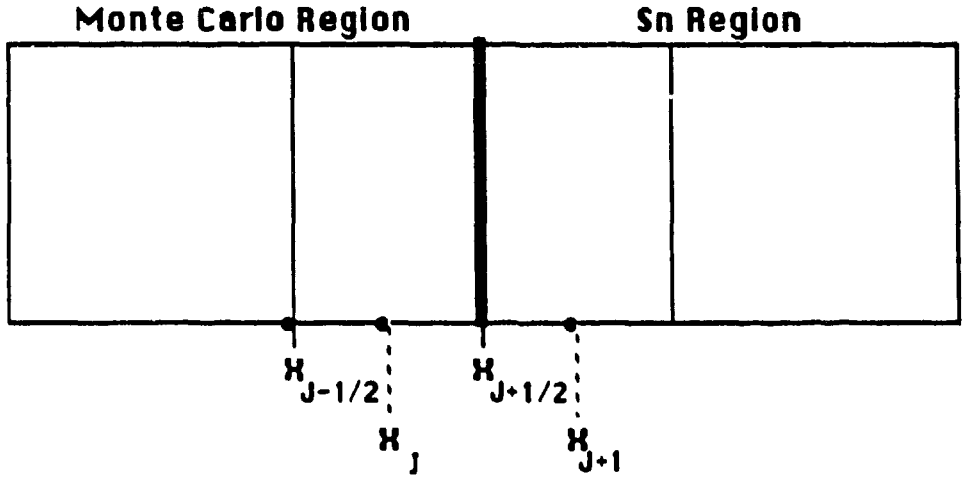

Figure 10-1 One-Dimensional Hybrid Monte Carlo/S_N Geometry

and

$$\tilde{Q}_{1,J}^{k+1/2} = \frac{2}{3}(B_{2,J+1/2}^{+} - \tilde{\varphi}_{2,J-1/2}^{+,k+1/2}) + \frac{1}{3}(\tilde{B}_{0,J+1/2}^{+} - \tilde{\varphi}_{0,J-1/2}^{+,k+1/2}) + \frac{1}{2}\Sigma_{T,J}\,\Delta x_J\,(\tilde{B}_{1,J+1/2}^{+} + \tilde{\varphi}_{1,J-1/2}^{+,k+1/2}), \tag{10-16b}$$

and the plus sign denotes summation only over the positive μ_n.

We then define acceleration equations

$$\tilde{\varphi}_{1,i+1/2}^{k+1} - \tilde{\varphi}_{1,i-1/2}^{k+1} + \Sigma_{T,i}\,\tilde{\varphi}_{0,i}^{k+1}\Delta x_i = \Sigma_{S,i}\,\tilde{\varphi}_{0,i}^{k+1}\Delta x_i + \tilde{Q}_{0,J}^{k+1/2}\Delta x_J\,\delta_{i,J} \tag{10-17}$$

and

$$\frac{2}{3}(\tilde{\varphi}_{2,i+1/2}^{k+1/2} - \tilde{\varphi}_{2,i-1/2}^{k+1/2}) + \frac{1}{3}(\tilde{\varphi}_{0,i+1/2}^{k+1} - \tilde{\varphi}_{0,i-1/2}^{k+1}) + \Sigma_{T,i}\,\tilde{\varphi}_{1,i}^{k+1}\Delta x_i = \tilde{Q}_{1,J}^{k+1/2}\Delta x_J\,\delta_{i,J}. \tag{10-18}$$

Subtracting Eq. (10-17), evaluated at $i+1$, from the same equation evaluated at i, and using the diamond difference relationship, we obtain

$$\tilde{\varphi}_{1,i+1}^{k+1} - \tilde{\varphi}_{1,i}^{k+1} + \frac{1}{2}(\Sigma_{R,i+1}\,\Delta x_{i+1}\,\tilde{\varphi}_{0,i+1}^{k+1} + \Sigma_{R,i}\,\Delta x_i\,\tilde{\varphi}_{0,i}^{k+1}) = \frac{1}{2}(\delta_{i+1,J} + \delta_{i,J})\,\tilde{Q}_{0,J}^{k+1/2}\Delta x_J. \tag{10-19}$$

From Eq. (10-18), we substitute in for $\tilde{\varphi}_{1,i+1}^{k+1}$ and $\tilde{\varphi}_{1,i}^{k+1}$ to get

$$\begin{aligned}
&\frac{-1}{3\Sigma_{T,i+1}\,\Delta x_{i+1}}(\tilde{\varphi}_{0,i+3/2}^{k+1} - \tilde{\varphi}_{0,i+1/2}^{k+1}) + \frac{1}{3\Sigma_{T,i}\,\Delta x_i}(\tilde{\varphi}_{0,i+1/2}^{k+1} - \tilde{\varphi}_{0,i-1/2}^{k+1}) \\
&\quad + \frac{1}{2}\Sigma_{R,i+1}\,\Delta x_{i+1}\,\tilde{\varphi}_{0,i+1}^{k+1} + \frac{1}{2}\Sigma_{R,i}\,\Delta x_i\,\tilde{\varphi}_{0,i}^{k+1} = \\
&\quad \frac{1}{2}(\delta_{i+1,J} + \delta_{i,J})\,\tilde{Q}_{0,J}^{k+1/2}\,\Delta x_J - \frac{1}{\Sigma_{T,J}}(\delta_{i+1,J} - \delta_{i,J})\,\tilde{Q}_{1,J}^{k+1/2} \\
&+ \frac{2}{3\Sigma_{T,i+1}\,\Delta x_{i+1}}(\tilde{\varphi}_{2,i+3/2}^{k+1/2} - \tilde{\varphi}_{2,i+1/2}^{k+1/2}) - \frac{2}{3\Sigma_{T,i}\,\Delta x_i}(\tilde{\varphi}_{2,i+1/2}^{k+1/2} - \tilde{\varphi}_{2,i-1/2}^{k+1/2}).
\end{aligned} \tag{10-20}$$

Finally, substituting in for the second moments from Eq. (10-15), we obtain the acceleration equation

$$\frac{-1}{3\Sigma_{T,i+1}\,\Delta x_{i+1}}(\tilde{\phi}^{k+1}_{0,i+3/2}-\tilde{\phi}^{k+1}_{0,i+1/2})+\frac{1}{3\Sigma_{T,i}\,\Delta x_i}(\tilde{\phi}^{k+1}_{0,i+1/2}-\tilde{\phi}^{k+1}_{0,i-1/2})$$
$$+\frac{1}{4}\Sigma_{R,i+1}\,\Delta x_{i+1}(\tilde{\phi}^{k+1}_{0,i+3/2}+\tilde{\phi}^{k+1}_{0,i+1/2})+\frac{1}{4}\Sigma_{R,i}\,\Delta x_i(\tilde{\phi}^{k+1}_{0,i+1/2}+\tilde{\phi}^{k+1}_{0,i-1/2})=$$
$$\frac{1}{2}(\delta_{i+1,J}+\delta_{i,J})\tilde{Q}^{k+1/2}_{0,J}\,\Delta x_J-R, \qquad (10-21)$$

where R is as defined at Eq. (10-4). Thus, by replacing the interior boundary source with a distributed source in cell J, and setting $\Sigma_{S,J}=0$, we obtain an acceleration equation almost identical in form to that of Eq. (10-3). The boundary conditions are identical to those of Eqs. (10-9) and (10-10).

When Eq. (10-21) is converged, and we let $\Sigma_{T,i+1}=\Sigma_{T,i}$ and $\Delta x_{i+1}=\Delta x_i$, it reduces to

$$\tilde{\phi}_{1,i+1}-\tilde{\phi}_{1,i}+\frac{1}{2}\Sigma_R\,\Delta x(\tilde{\phi}_{0,i+1}+\tilde{\phi}_{0,i})=\frac{1}{2}(\delta_{i+1,J}+\delta_{i,J})\tilde{Q}_{0,J}\,\Delta x. \qquad (10-22)$$

Comparing this with the zeroth moment of Eq. (10-13), we see that the diffusion acceleration equation [Eq. (10-21)] converges to the transport equation, as required.

10.2 Application of the Hybrid DSA Method to a Sample Problem

We now apply the diffusion synthetic method derived above to an appropriate sample problem. The sample problem consists of a one-dimensional mesh in Cartesian geometry, 10 cm in length, with isotropic scattering, $\Sigma_T = 1.0$ cm, and

$\Sigma_S = 0.95$ cm. To avoid negative fluxes, we set $\Delta x = .25$ cm, so there are a total of 40 meshes. Vacuum boundary conditions are used on the left and right-hand edges.

We begin by studying the single region, pure S_N problem, with a fixed source located in cell $J = 20$, and a convergence criteria of 10^{-4}. We first apply the source iterative method of Eqs. (10-1) and (10-2), which requires 98 iterations for convergence. Next, we apply the DSA method of Eqs. (10-3) and (10-4), and find that the number of iterations required for convergence is reduced to 6.

For the hybrid method, we divide the problem geometry into S_N and "Monte Carlo" regions, where the S_N regions consists of cells 21 through 40. The fixed source in cell 20 is now replaced by known boundary fluxes at the Monte Carlo/S_N interface, where the boundary fluxes used in this case are those determined by the source iterative method above. We first apply the hybrid equivalent of the source iterative method, Eqs. (10-11) and (10-2), where $\Sigma_{S,J} = 0$, and find that it requires 73 iterations for convergence, and results in scalar fluxes in the S_N region that are within .1% of the standard source iterative method. Next, we implement Eq. (10-21) with the hybrid problem, and find that the hybrid DSA method reduces the number of iterations required for convergence to 8, again with fluxes that are within .1% of the S_N region values.

Comparing the actual rates of convergence (i.e., the ratio of the error in one iteration to the error in preceeding iteration) between the four different methods, we see that the standard source iterative method demonstrates a spectral radius of $\rho = .93$, while the hybrid source iterative method has $\rho = .89$. The maximum ratio in the standard DSA method is .20, while that in the hybrid DSA method is .28.

Thus, we see that Fourier stability analysis accurately predicts the spectral radii for the source iterate and standard DSA methods (.93 versus .95, and .20 versus .21, respectively), while the hybrid source iterate method has a spectral radius slightly lower than the standard source iterate method (.89 versus .93), and the hybrid DSA method has a higher spectral radius than the comparable standard DSA method (.28 versus .20).

Since Eq. (10-11) is very similar to Eq. (10-1), the standard source iterate method, we would expect the two methods to posses similar convergence properties. Comparing the respective DSA equations, however, [Eqs. (10-3) and (10-21)], we see that the hybrid DSA method contains the unaccelerated source term $\tilde{Q}_{0,J}^{k+1/2}$ [Eq. (10-16a)], so that, depending upon the magnitudes of $\tilde{o}_{0,J-1/2}^{+,k+1/2}$ and $\tilde{o}_{1,J-1/2}^{+,k+1/2}$, the spectral radius of the hybrid DSA method is increased. However, by setting $\Sigma_{S,I} = 0$ throughout the entire Monte Carlo region, $\tilde{o}_{0,J-1/2}^{+,k+1/2}$ and $\tilde{o}_{1,J-1/2}^{+,k+1/2}$ vanish for our problem geometry, and the source term in Eq. (10-21) goes to a constant. In this case, we expect the spectral radius to be the same as, or less than (since we have reduced the overall amount of scattering), the standard DSA method. Upon implementing the change, we find that the number of iterations required for convergence is reduced to 7, with a spectral radius of .16.

10.3 Implementation of the Hybrid DSA Method in TWODANT

The above discussion and sample problem is based on a one-dimensional geometry. In the two-dimensional geometry actually used by the hybrid method,

however, the replacement of the interior boundary sources with an exact inhomogeneous source [Eq. (10–12)] is not feasible. Instead, as implemented by Alcouffe,[35] we construct a fixed source, located inside the Monte Carlo region, from the interior boundary sources, and zero out all within-group scattering sources. This construction has proven to be as successful in accelerating the inner iterations as the standard DSA method, i.e., the number of inner iterations required for an S_N calculation in the hybrid method is less than or equal to the number of inner iterations required by the standard S_N method.

CHAPTER 11.
BENCHMARKS

We now apply the hybrid Monte Carlo/S_N method to several relatively simple problems, where the results of standard S_N calculations may be used as benchmarks. Although we have examined the behavior and validity of the individual components of the hybrid Monte Carlo/S_N theory in the preceeding chapters, the purpose of these comparisons are to assess the overall performance and accuracy of the hybrid method in a semi-realistic environment. Thus, we use the set of 16-group Hansen-Roach cross sections previously described in Chapter 8 with problems of varying geometry and composition, where the selected materials and specified densities are representative of those encountered in typical calculations. Note that these problems have been selected for their ease of computation by the standard S_N method, and are thus not necessarily typical of the types of problems the hybrid method was designed for.

11.1 The Graphite Block Benchmark

We begin by examining a 22 cm by 22 cm homogeneous block ($X-Y$ geometry), where the block is composed of graphite with a density of 1.6 gm/cm^3. The block is divided into 121 coarse meshes with dimensions of 2 cm by 2 cm, with an

isotropic distributed source located in the center of the block between $x = 10$ to 12 cm, $y = 10$ to 12 cm, the source neutrons being emitted in group 1. The mean free path of neutrons in this material ranges from $\lambda_1 = 10.1$ cm, for group 1, to $\lambda_{16} = 2.8$ cm, for group 16. Since there are no fissionable materials present in the problem geometry, and the Hansen-Roach cross sections do not include upscatter, the calculation is a pure downscatter problem and requires only one outer iteration. Note that, for group 1, the source is located only approximately 1 mfp from the block edge, so that transport effects may be expected to be important.

For the hybrid Monte Carlo/S_N code, we designate the region $x = 8$ to $x = 12$ cm, $y = 8$ to $y = 12$ cm as the "fixed" Monte Carlo region (Fig. 11-1), and specify a boundary layer thickness of 1 mfp. If we examine a group-dependent cross-sectional view of the block (Fig. 11-2), we see that the actual Monte Carlo portion of the problem geometry resembles an inverted wedding cake, and that the Monte Carlo region comprises the entire problem geometry for groups 1 and 2. Thus, the calculations for the first two groups will be done entirely by Monte Carlo, while all other groups will require some linkage between the Monte Carlo regions and S_N regions via response matrices. An S_6 solver is used in the S_N region, with one fine mesh per coarse mesh.

We use 250,000 histories to sample the fixed source, located in group 1. While this results in relative errors (one standard deviation) of less than 3% for the coarse mesh tracklengths in the first two groups, and less than 5% in the fifth group, the error rapidly increases as we move further down in group structure, since the number of particles reaching a given group decreases. However, the need for an accurate determination of the flux due to the source alone also decreases, since most

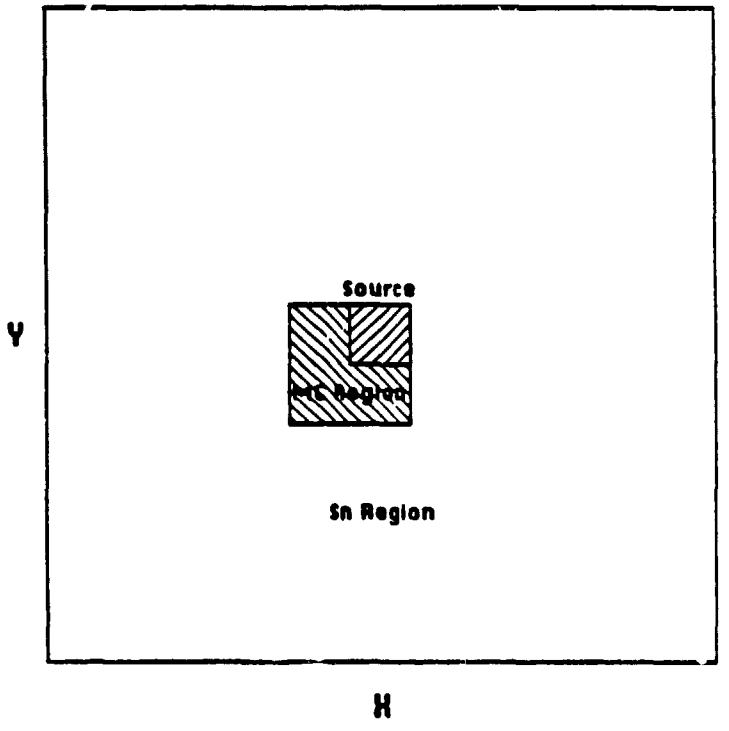

Figure 11-1 Graphite Block Problem Geometry

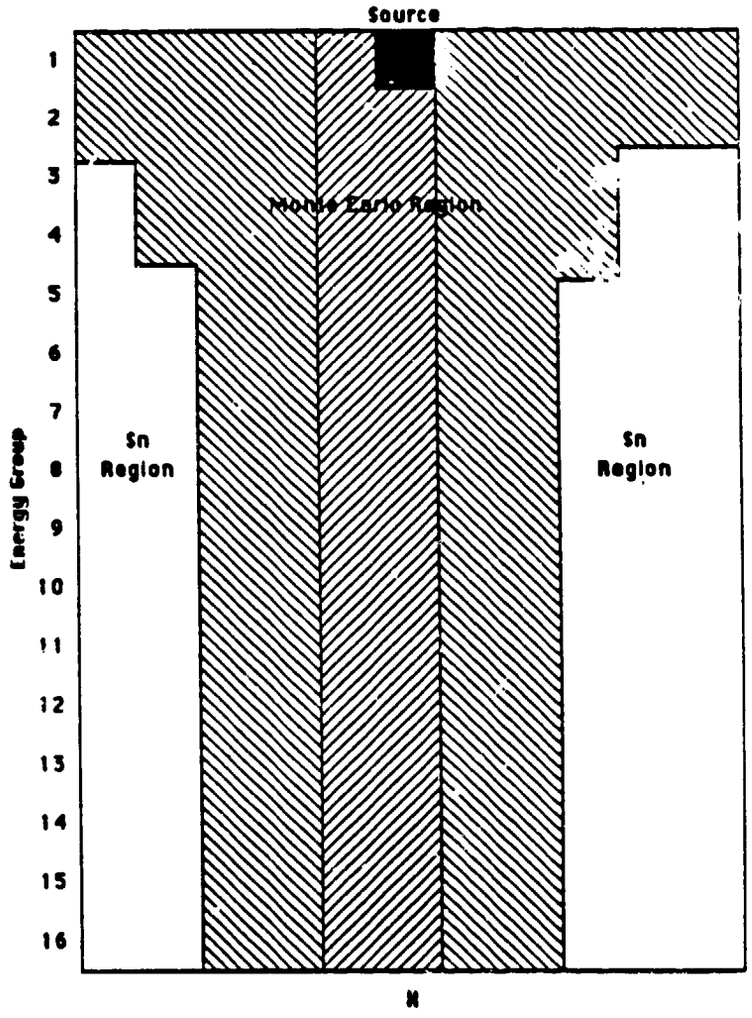

Figure 11-2 Monte Carlo/S_N Structure, Graphite Block Problem

of the flux in lower energy groups is due to particles that have crossed between the Monte Carlo and S_N regions, so the large variances in the fixed source calculation at lower energy groups are not disastrous. In contrast, as can be seen at Table 11-1 below, approximately the same number of histories are used to sample the incoming boundary fluxes for each group, regardless of their size. This preserves the accuracy of the solution at lower energy groups,.

Table 11-1

Numbers of Histories and Particle Balances, Graphite Block Problem

Group	L_{FX}	L_{BF}	T_{RM}	T_{BF}	R_{ERR}	PBAL
3	1.5739e-1	3.2717e-2	82.944	95.797	.0071	-2.5107e-4
4	8.3656e-2	3.2090e-2	82.944	95.803	.0069	-8.3290e-4
5	2.8144e-2	3.2354e-2	62.208	71.856	.0074	2.4083e-4
6	8.3440e-3	1.9599e-2	62.208	71.852	.0072	9.7063e-4
7	2.6200e-3	1.0462e-2	62.208	71.851	.0074	2.3276e-3
8	8.1600e-4	5.8015e-3	62.208	71.854	.0069	-4.2840e-4
9	2.5600e-4	3.2610e-3	62.208	71.861	.0072	-1.6513e-3
10	7.2000e-5	1.4992e-3	62.208	71.853	.0071	-7.0129e-4
11	2.4000e-5	9.0443e-4	62.208	71.855	.0076	8.6142e-4
12	1.6000e-5	6.3568e-4	62.208	71.863	.007.	2.3851e-4
13	8.0000e-6	3.8876e-4	62.208	71.853	.0076	1.5884e-4
14	4.0000e-6	2.3105e-4	62.208	71.861	.0074	8.1158e-5
15	4.0000e-6	2.1708e-4	62.208	71.850	.0070	5.1439e-4
16	3.9981e-6	3.5249e-4	62.208	71.846	.0063	6.0797e-5

L_{FX} - Particle Source in Monte Carlo Region Due to Sampling of Fixed Source
L_{BF} - Particle Leakage from S_N Region into Monte Carl - Region
T_{RM} - Total Number of Histories Used in Response Mr Calculation
T_{BF} - Total Number of Histories Used to Sample Incoming Boundary Fluxes
R_{ERR} - Relative Error Between Calculated Outgoing Boundary Fluxes
PBAL - Total Particle Balance for Group

For comparison, we perform several standard S_N calculations with varying quadrature order and fine mesh structure. The CPU times required for the various S_N calculations, as well as the hybrid calculation, are shown below at Table 11-2, where

the notation 1×1 or 2×2 refers to the fine mesh structure used in the calculation (1 fine mesh per coarse mesh or four fine meshes per coarse mesh, respectively). The particle leakage as a function of energy group is shown at Fig. 11-3, while the total reaction rates for the coarse mesh cells along the left edge of the block are shown at Fig. 11-4 for group 1, and Fig. 11-5 for group 16.

Table 11-2
Computational Times, Graphite Block Problem

Type	CPU Time
$S_6/1 \times 1$	4.5 sec
$S_{12}/2 \times 2$	14.3 sec
$S_{20}/2 \times 2$	32.9 sec
MC/S_6	120.4 sec

From Fig. 11-3, we see that the leakage calculated by the hybrid method closely matches that calculated by the S_N method over all 16 energy groups, even though there is over a factor of 10^3 decrease from group 1 to group 16. In Fig. 11-4, the S_6 and S_{12} calculations clearly show ray effects, while the MC/S_6 and S_{20} calculations do not. Of course, ray effects are no longer a factor by the time group 16 is reached, so that all S_N calculations are within 5% of one another, as is the hybrid calculation. It is somewhat surprising that the hybrid calculation shows little statistical variation in Fig. 11-5, even though the reaction rates are two orders of magnitude less than those in group 1. This may be due to the "smoothing" effects of the S_N operator in the hybrid code, since the left edge is entirely contained within the S_N region for group 16 (reference Fig. 11-2). An attempt to obtain the same accuracy with analog Monte Carlo alone would require a much larger number of histories to be run than for the hybrid case, since the probability of an individual particle reaching group 16 without leaking is less than .04%. Thus, for an equal

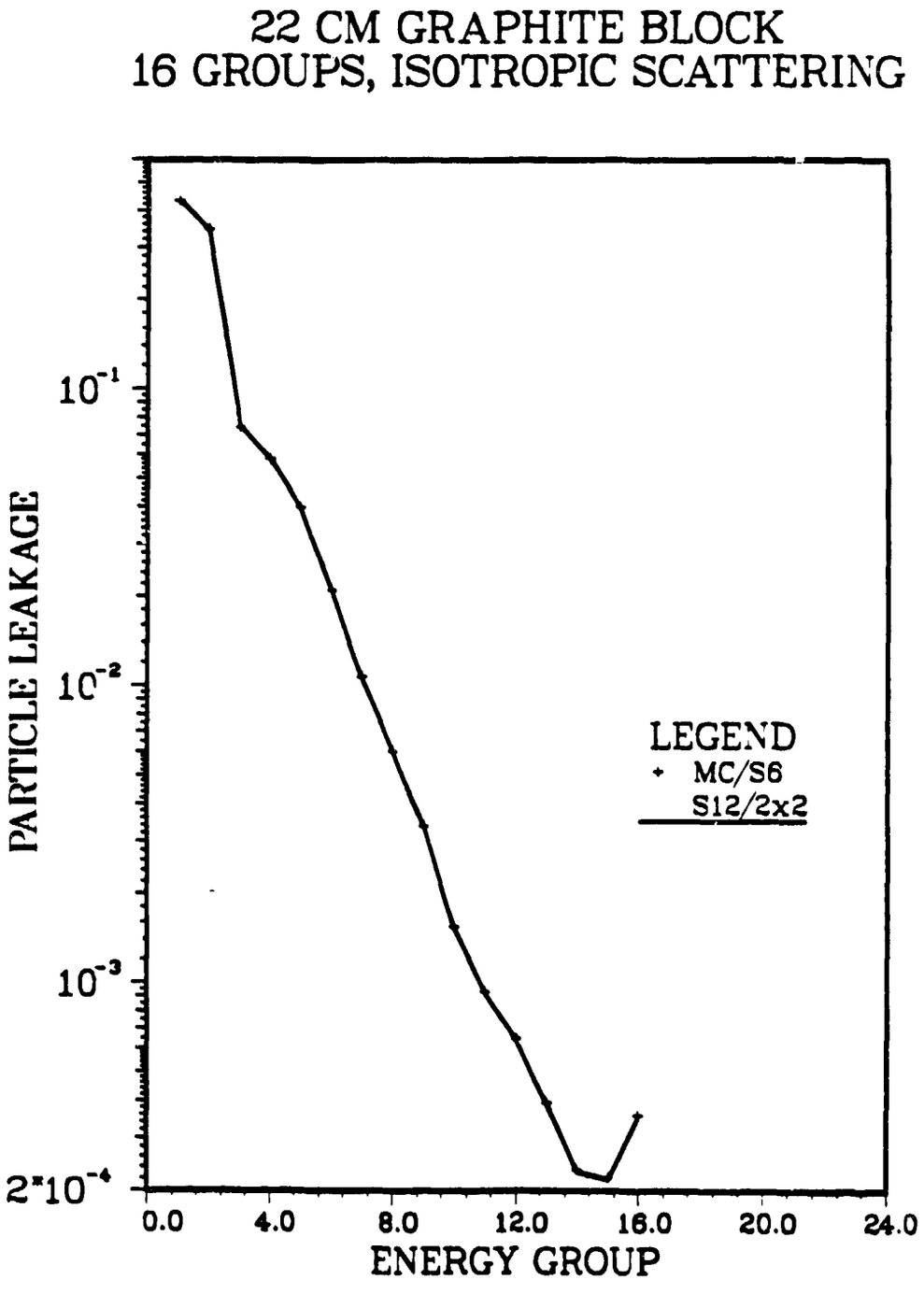

Figure 11-3 Particle Leakage, Graphite Block Problem

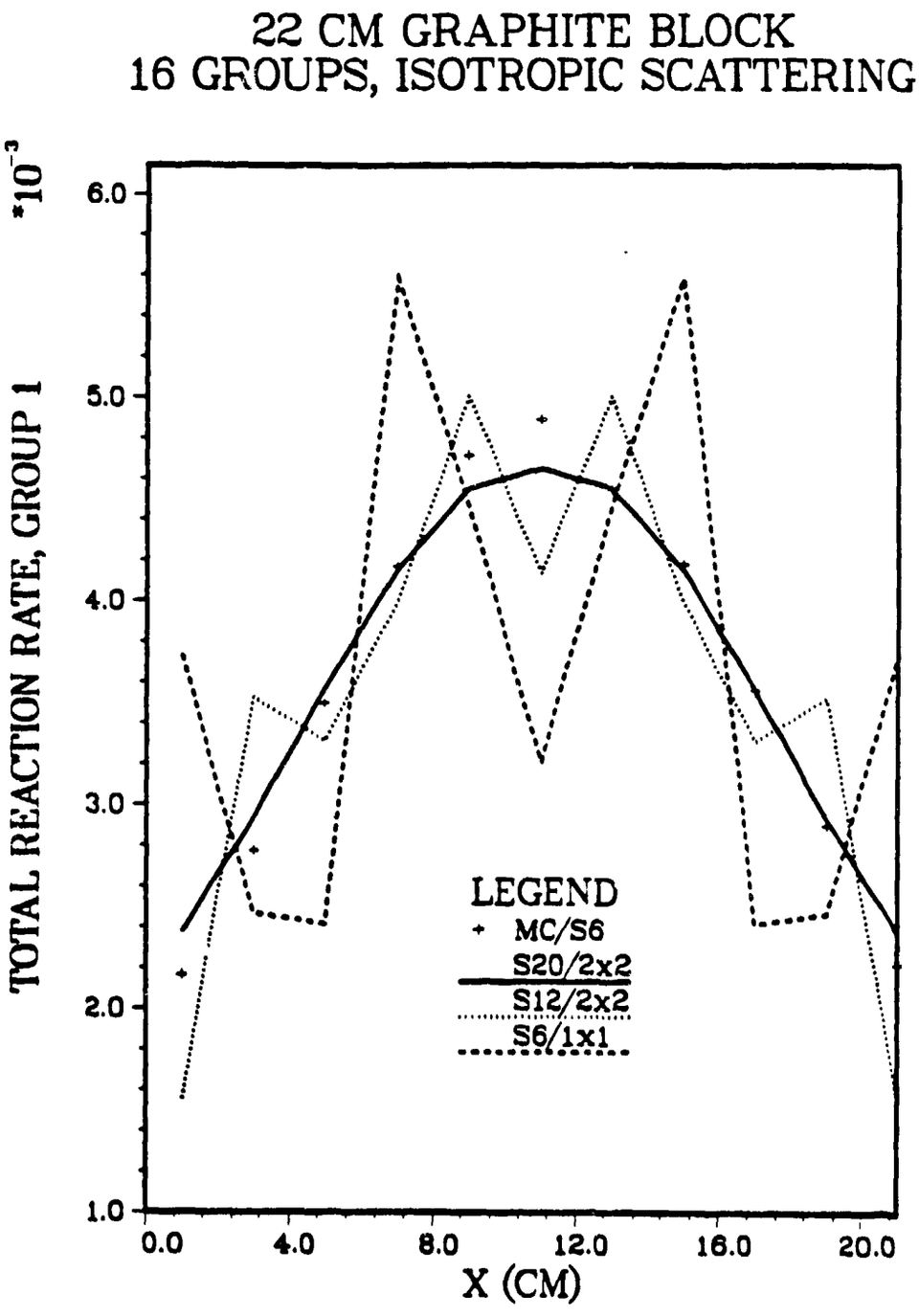

Figure 11-4 · Total Reaction Rates, y = 1 Cm. Group 1, Graphite Block Problem

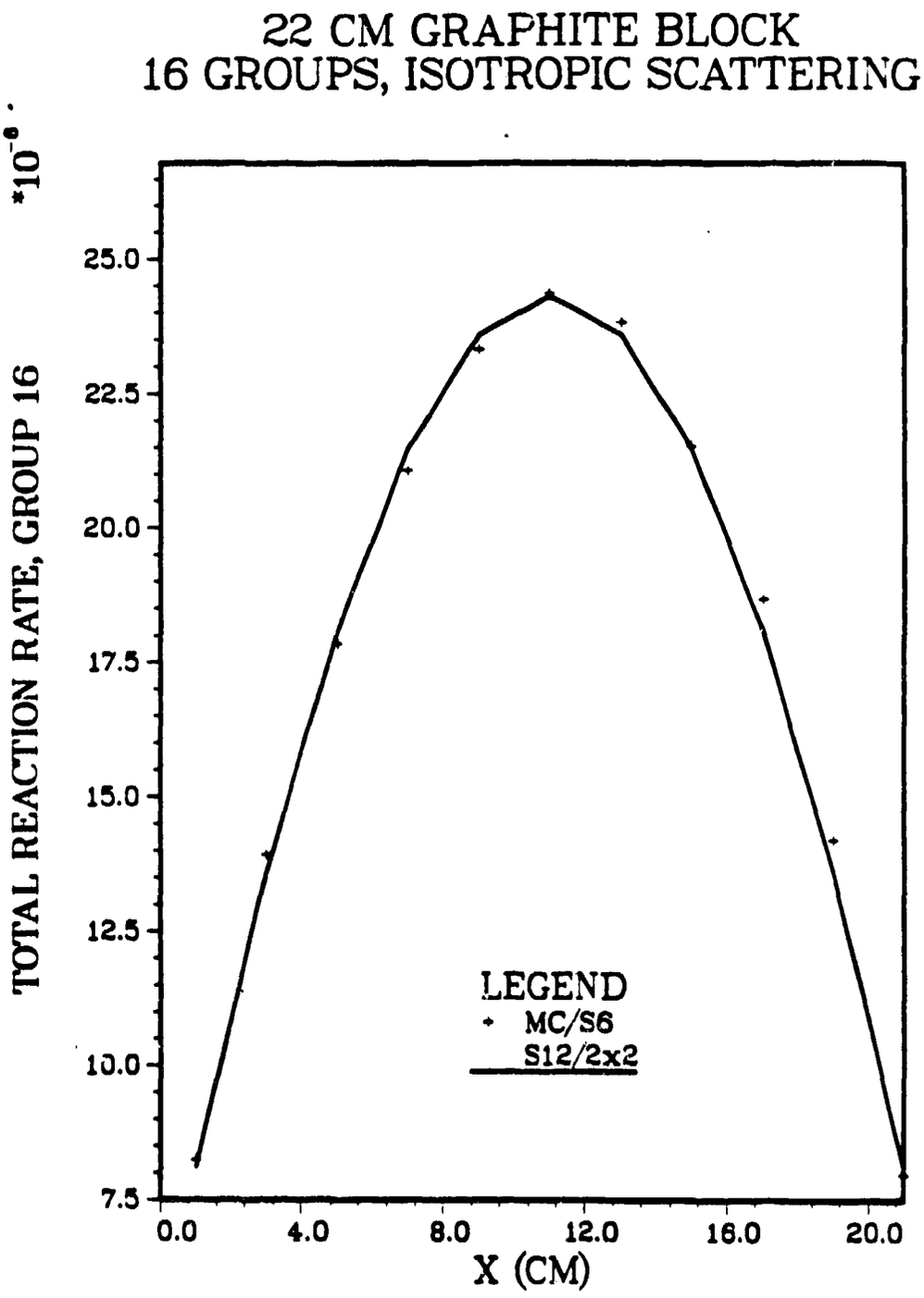

Figure 11-5 Total Reaction Rates, $y = 1$ Cm, Group 16, Graphite Block Problem

number of total histories (approximately 2.2 million), less than 900 would even reach group 16, clearly not enough to provide an adequate sample. The agreement in Fig. 11-5 also illustrates that the hybrid method is capable of producing accurate results even after a large amount of coupling has occurred between the Monte Carlo and S_N regions.

11.2 The Uranium Rod Benchmark

The next benchmark problem consists of a uranium (UO_2) rod surrounded by a graphite reflector in $R-Z$ geometry. The uranium rod has a radius of 2 cm, a density of 10 gm/cm^3, and is 10% enriched in U235, while the graphite reflector is 13 cm thick and has a density of 1.6 gm/cm^3. The height of the cylinder is 30 cm (reference Fig. 11-6). An isotropic source is spatially distributed uniformly throughout the uranium rod, with an energy spectrum identical to the fission spectrum of U235, for which only the first six groups have non-zero values. The mean free path in uranium ranges from $\lambda_1 = 6.3$ cm to $\lambda_{16} = .6$ cm, while that in the graphite varies between $\lambda_1 = 10.1$ cm and $\lambda_{16} = 2.8$ cm. The problem is divided into coarse meshes of size 1 cm by 2 cm.

For the hybrid calculation, the region consisting of the uranium oxide rod is designated as the Monte Carlo region, and a 1 mfp boundary layer is specified, resulting in the Monte Carlo/S_N structure shown at Fig. 11-7. An S_6 solver is used in the S_N region, with two fine meshes (1×2) per coarse mesh. 250,000 histories are used to sample the fixed source, resulting in relative errors in the tracklengths for the first five groups of less than 3%, and increasing afterwards as the group sampling

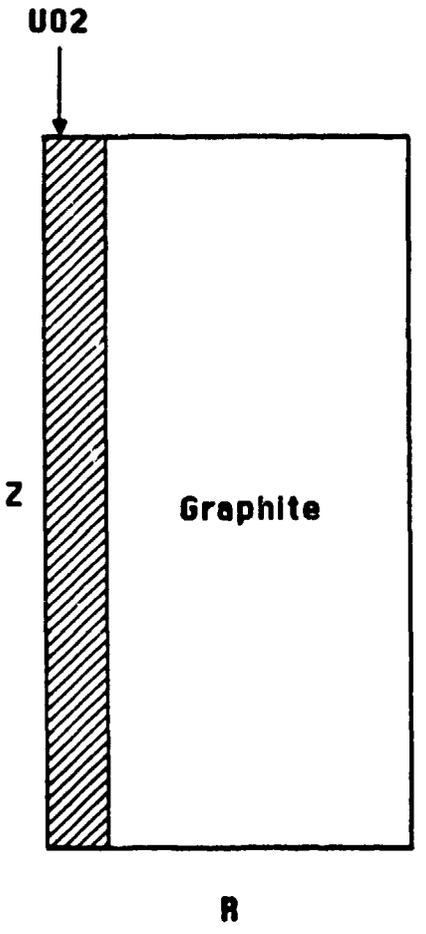

Figure 11-6 Uranium Rod Problem Geometry

frequency decreases. The size of each response matrix is identical, with 77,760 histories used to sample each one. The Monte Carlo region source (including fission and downscatter) due solely to the fixed source, the size of the incoming boundary currents, and the numbers of histories used to sample the incoming boundary fluxes are shown below for each group and outer iteration. The particle balance for all groups was $\leq 7 \times 10^{-4}$.

Table 11-3
Numbers of Histories, Uranium Rod Problem

Group	L_{FX}	L_{BF1}	T_{BF1}	P_{ERR}	L_{BF2}	T_{BF2}
1	2.1809e-1	2.8732e-3	64.325	.0105	2.5020e-5	441
2	4.4949e-1	2.8694e-2	61.306	.0093	2.5017e-4	454
3	2.8584e-1	2.8222e-2	56.821	.0104	2.4830e-4	467
4	3.4305e-1	6.8156e-2	55.327	.0089	5.5757e-4	444
5	2.3938e-1	1.0609e-1	53.819	.0083	8.5701e-4	439
6	7.1481e-2	7.8252e-2	52.325	.0085	6.0819e-4	420
7	1.3344e-2	4.2133e-2	52.314	.0079	3.2074e-4	422
8	2.5531e-3	2.2454e-2	52.322	.0078	1.6661e-4	411
9	4.6359e-4	1.1559e-2	52.316	.0079	8.5007e-5	410
10	5.0839e-5	4.7356e-3	52.319	.0090	3.4983e-5	414
11	4.6383e-6	2.5691e-3	52.319	.0081	1.8800e-5	418
12	0.0000e-0	1.6111e-3	52.323	.0089	1.1960e-5	424
13	0.0000e-0	9.2686e-4	52.321	.0085	6.8299e-6	421
14	0.0000e-0	5.2495e-4	52.322	.0083	3.9655e-6	431
15	0.0000e-0	4.5132e-4	52.310	.0087	3.3694e-6	422
16	0.0000e-0	5.5720e-4	52.312	.0074	4.1072e-6	420

Note that the size of the incoming boundary fluxes for the second outer iteration is approximately a factor of 100 less than the first outer iteration. Since the fissile material is contained entirely within the Monte Carlo region, most of the multiplication within the system is determined when the fixed source is sampled. This is

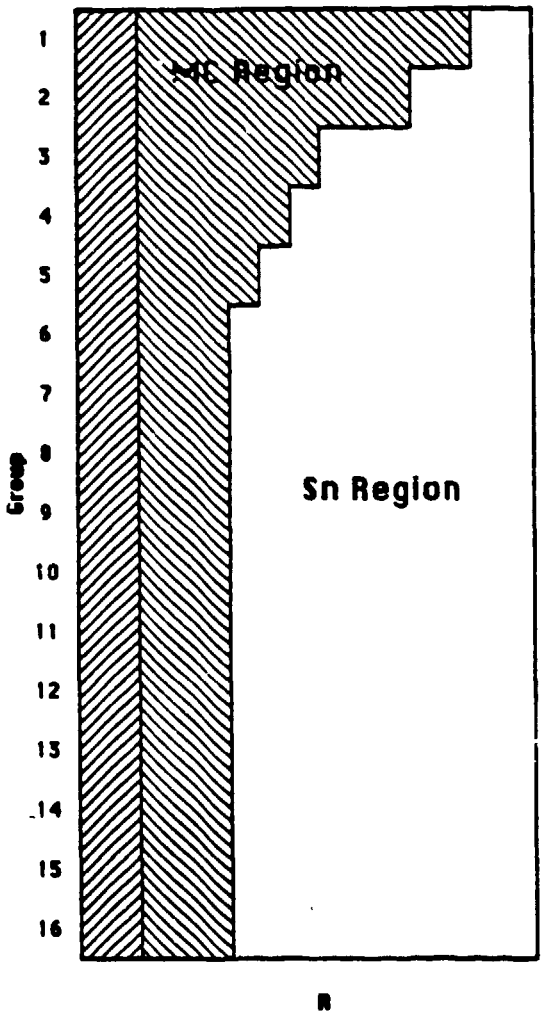

Figure 11-7 Monte Carlo/S_N Structure, Uranium Rod Problem

confirmed by comparing the fission source (FS) resulting from the sampling of fixed source (6.79×10^{-2}) with the total fission source (7.67×10^{-2}).

The hybrid results were compared with three different S_N calculations. The computation times, number of outer iterations required for convergence, and calculated fission sources are shown below at Table 11-4. The particle leakage as a function of energy group is shown at Table 11-5, while the total reaction rates along the centerline of the cylinder ($r = .5$ cm) are compared at Figs. 11-7 and 11-8 below for groups 1 and 16, respectively.

Table 11-4
Computational Times and Fission Sources, Uranium Rod Problem

Type	CPU Time	# Outers	FS
$S_6/1 \times 2$	8.9 sec	3	7.97e-2
$S_{12}/2 \times 4$	54.6 sec	3	7.70e-2
$S_{12}/4 \times 8$	127.0 sec	3	7.69e-2
MC/S_6	152.8 sec	2	7.67e-2

Table 11-5
Particle Leakage, Uranium Rod Problem

Group	$S_6/1 \times 2$	$S_{12}/2 \times 4$	MC/S_6
1	9.26e-2	9.37e-2	9.36e-2
2	2.56e-1	2.58e-1	2.57e-1
3	1.05e-1	1.05e-1	1.06e-1
4	1.61e-1	1.61e-1	1.60e-1
5	1.73e-1	1.72e-1	1.72e-1
6	1.09e-1	1.08e-1	1.08e-1
7	5.96e-2	5.86e-2	5.86e-2
8	3.45e-2	3.38e-2	3.37e-2
9	1.86e-2	1.82e-2	1.82e-2
10	7.98e-3	7.77e-3	7.80e-3
11	4.49e-3	4.36e-3	4.38e-3
12	2.84e-3	2.75e-3	2.77e-3
13	1.63e-3	1.58e-3	1.59e-3
14	9.09e-4	8.79e-4	8.85e-4
15	7.81e-4	7.55e-4	7.61e-4
16	9.87e-4	9.50e-4	9.61e-4

From Table 11-5, we see that while the hybrid MC/S_6 and S_{12} particle leakages are in good agreement with each other, the standard S_6 calculation differs somewhat, not only in the first group, which is not surprising, but in lower energy groups as well. Thus, the advantage of using the more accurate Monte Carlo method in the uranium rod region carries through for all energy groups. Examining Figs. 11-8 and 11-9, we see that the hybrid S_6 calculation, despite some statistical variation since the centerline region is contained entirely within the Monte Carlo region, agrees well with the S_{12} calculations, while the the S_6 calculation is clearly off. If we compare the reaction rates at the outer edge of the cylinder (Fig. 11-10), we see that the hybrid results have been smoothed by the S_N operator, and are in good agreement with the S_{12} calculation, while the standard S_6 results are slightly larger.

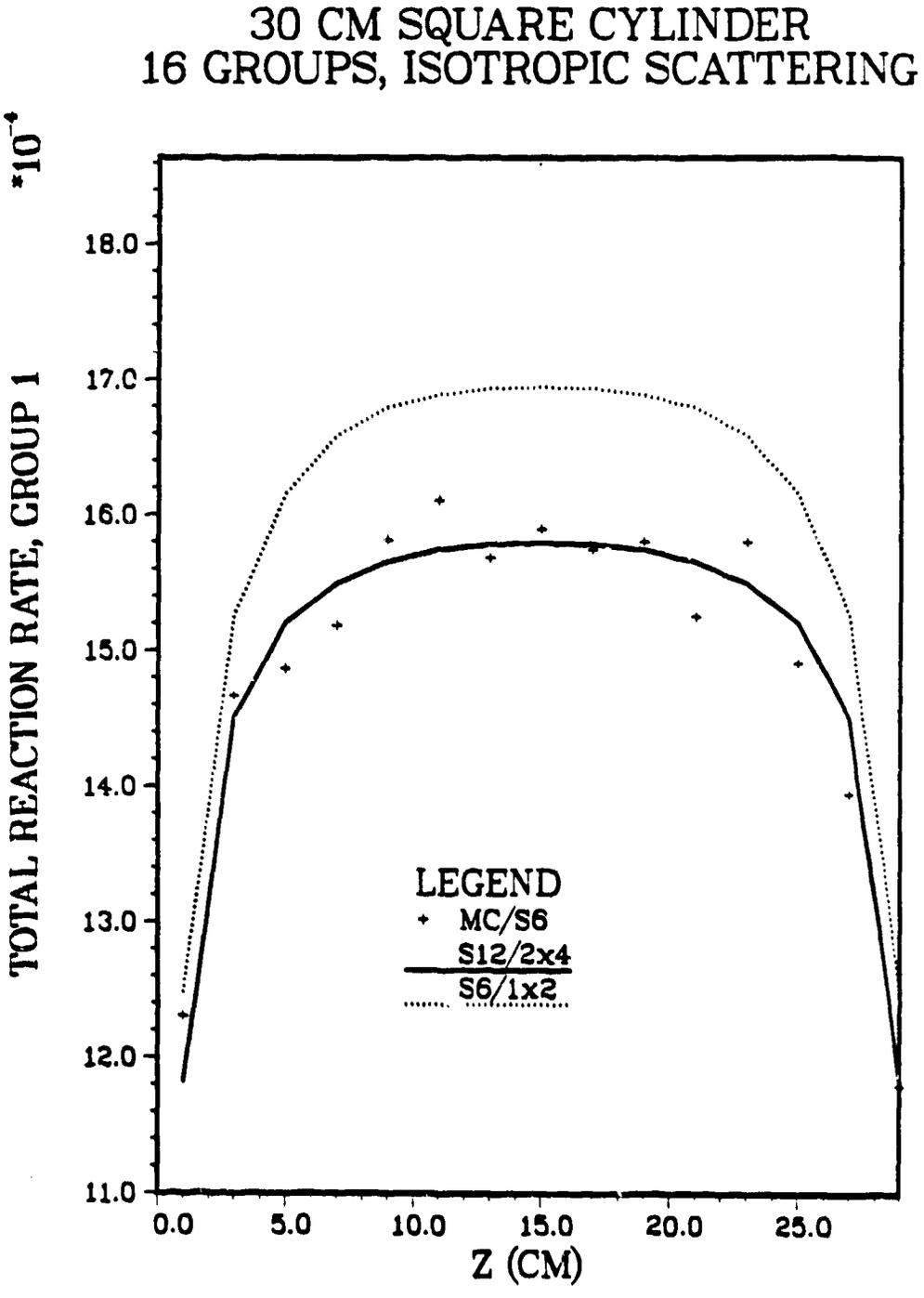

Figure 11-8 Total Reaction Rates, r = .5 Cm, Group 1, Uranium Rod Problem

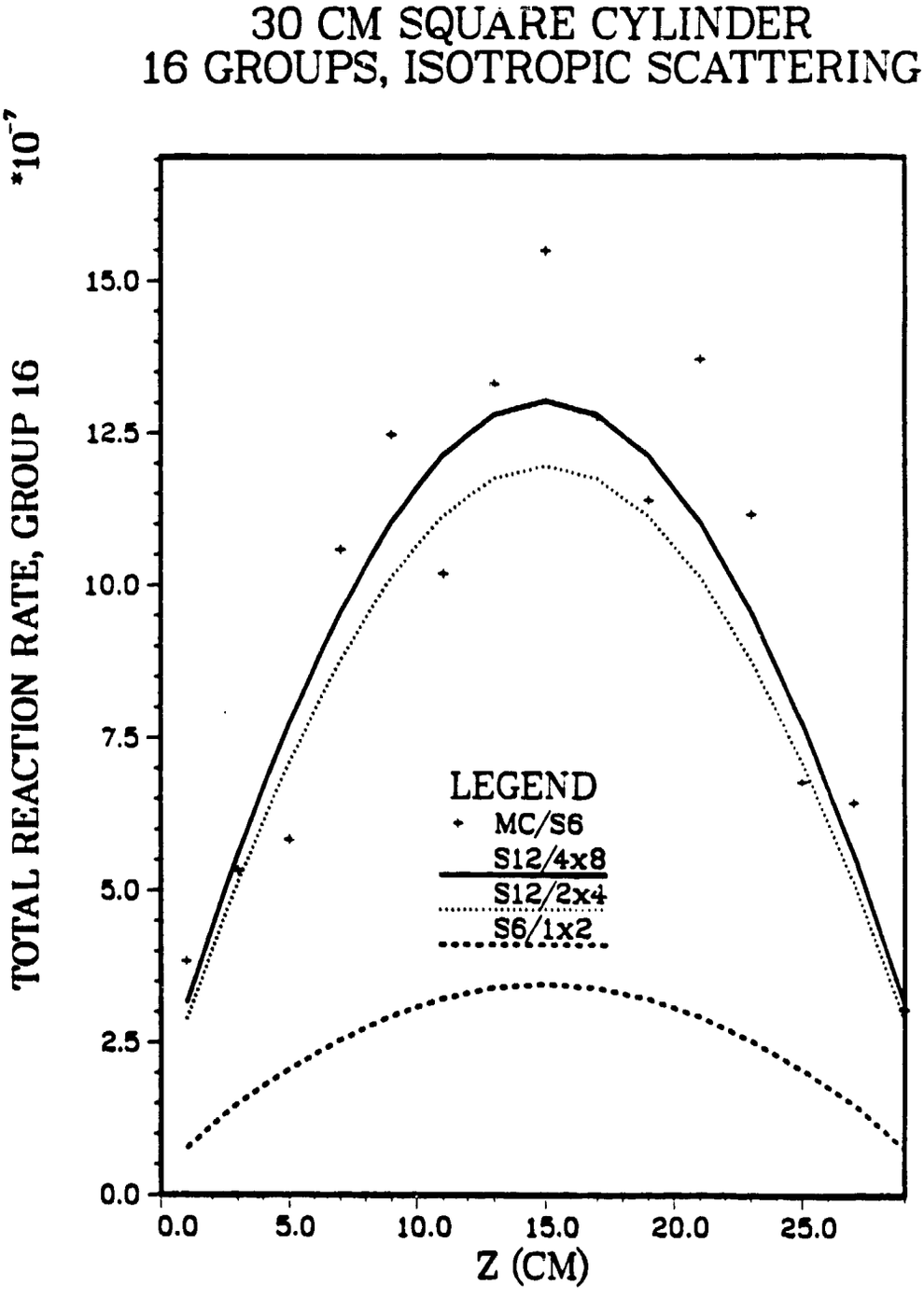

Figure 11-9 Total Reaction Rates, r = .5 Cm, Group 16, Uranium Rod Problem

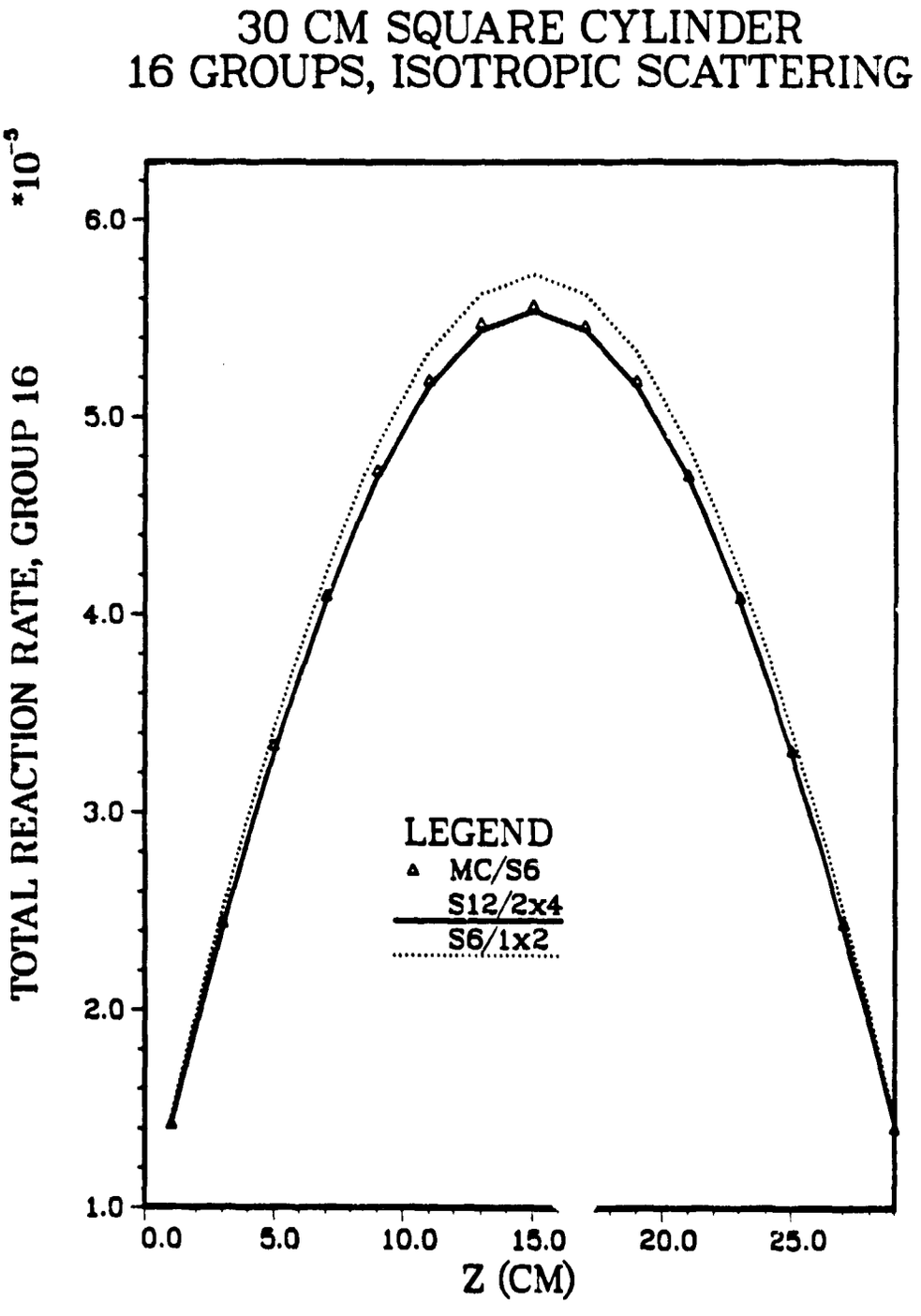

Figure 11-10 Total Reaction Rates, r = 14.5 Cm, Uranium Rod Problem

The hybrid Monte Carlo/S_6 calculation clearly provides more accurate results for all regions of the problem than a standard S_6 calculation, even for this simple geometry. This is also reflected in the respective fission sources computed by the respective methods.

11.3 The Al/U235 Block Problem

We now examine a more difficult problem for which the hybrid method is better suited. The problem geometry consists of an aluminum plate, 10 cm long and 3 cm high, with a density of 3 gm/cm^3, sandwiched between two uranium (UO_2) blocks, each 10 cm long and 5 cm high. The uranium blocks are 100% enriched in U235, and have a density of 10 gm/cm^3. The source consists of an isotropic boundary source on the left edge, located between $y = 6$ to $y = 7$ cm, in group 1 (reference Fig. 11-11). The mean free path in the aluminum plate ranges from $\lambda_8 = 12.2$ cm to $\lambda_5 = 5.9$ cm, while that in the uranium blocks ranges from $\lambda_2 = 6.3$ cm to $\lambda_{16} = .2$ cm. The problem is divided into coarse meshes of size 1 cm by 1 cm. Since the length of the aluminum plate is approximately 1 mfp for the source neutrons, an accurate determination of the fluxes at the right-hand edge should present problems for the standard S_N method.

The aluminum plate is designated as the Monte Carlo region, and a 1 mfp boundary layer is once again specified, resulting in the Monte Carlo/S_N structure shown at Fig. 11-12 below, where the Monte Carlo region comprises the entire problem geometry for the first two groups. Since the flux decreases rapidly after

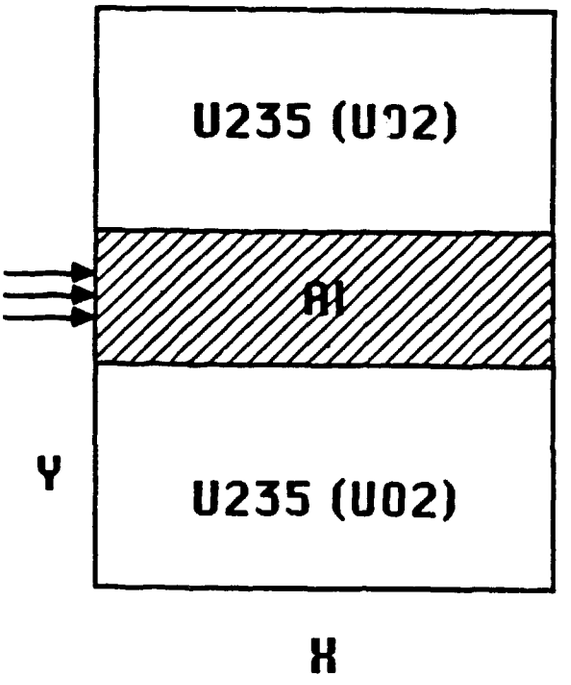

Figure 11-10 The Al/U235 Problem Geometry

group 7, the calculations for groups 8 through 16 are done entirely by S_N. Approximately 530,000 histories are used to sample the fixed source, resulting in relative errors in the coarse mesh tracklengths of less than 2% in group 1, 11% in group 6, and 67% in group 7. 103,680 histories apiece are used to sample each response matrix in groups 3 through 7. The particle balances for the Monte Carlo/S_N groups were all less than 5×10^{-4}, while the R_{ERR} values were between .0058 and .0073. The group sources and number of histories run are shown below at Table 11-6 for group 4, where OITNO is the outer iteration number, L_{SN} represents the fission source from S_N regions to the group 4 Monte Carlo region, and T_{SN} is the number of histories used to sample the S_N fission source. Note that T_{SN} is a constant, and T_{BF} goes to a constant, because their respective sources start at or reach a level where only one history is used to sample each state. Although some of the sampled residuals for later outer iterations contain negative elements, their magnitudes are small and do not adversely affect the calculation.

Table 11-6
Source Strengths and Numbers of Histories, Group 4, Al/U235 Problem

OITNO	L_{SN}	T_{SN}	L_{BF}	T_{BF}
1	-	-	3.3999e-2	119,767
2	3.3919e-3	6,720	9.2667e-3	32,470
3	3.6550e-4	6,720	1.4379e-3	4,861
4	5.2505e-5	6,720	2.1485e-4	617
5	7.2478e-6	6,720	2.9681e-5	480
6	1.0413e-6	6,720	4.6696e-6	480
7	1.9434e-7	6,720	9.2425e-7	480

Table 11-7 below lists the run times, number of outer iterations required for convergence, and the calculated fission sources for the hybrid method, various S_N runs,

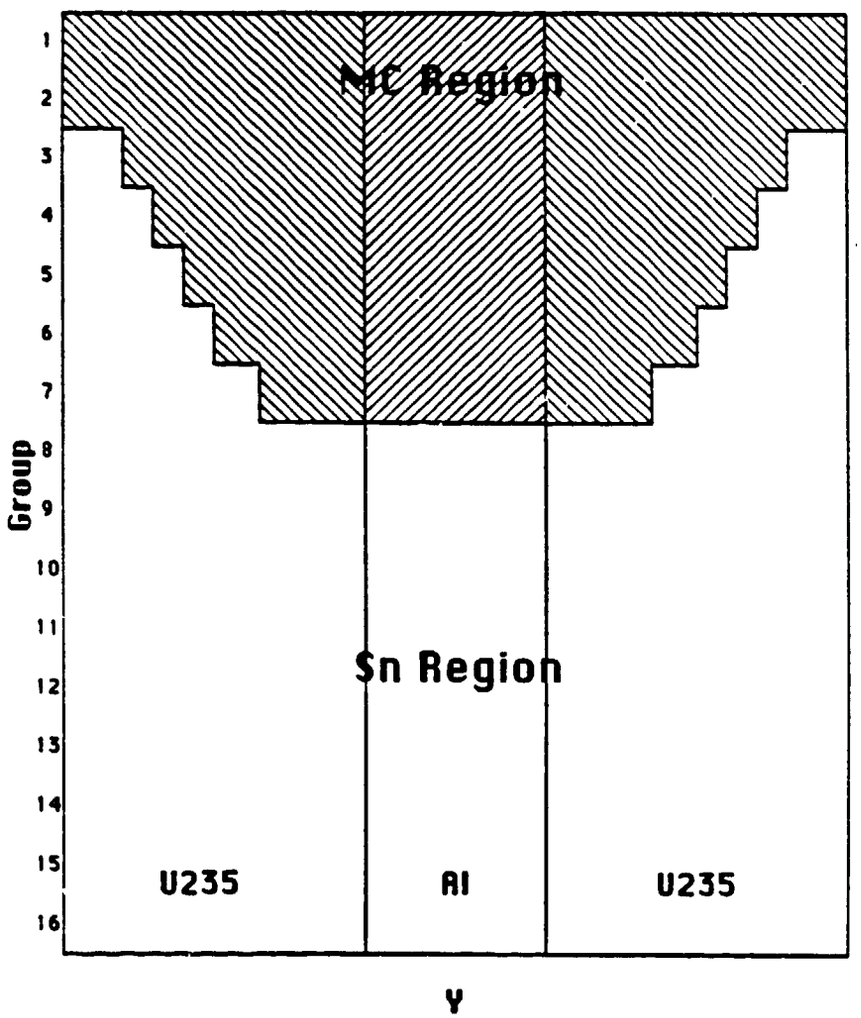

Figure 11-12 Monte Carlo/S_N Structure, Al/U235 Problem

and a Monte Carlo run. The Monte Carlo run was performed by defining the entire problem geometry as the Monte Carlo region, and increasing the number of histories used to sample the fixed source to approximately 2.1 million. Figure 11-13 presents a comparison of particle leakage for the S_6 case and the hybrid method. Figs. 11-14 and 11-15 show the group 1 total reaction rates for the left and right edges, respectivley, while Fig. 11-16 shows the left edge total reaction rates for group 6, and Fig. 11-17 the same for group 7.

Table 11-7
Computational Times and Fission Sources, Al/U235 Problem

Type	CPU Time	# Outers	FS
MC/S_6	144.7 sec	7	.9426
MC	155.6 sec	-	.9440
$S_6/2 \times 2$	15.6 sec	7	.9679
$S_{12}/2 \times 2$	35.3 sec	5	.9444
$S_{20}/2 \times 2$	95.9 sec	6	.9544
$S_{30}/2 \times 2$	206.8 sec	6	.9560
$S_{30}/4 \times 4$	458.5 sec	6	.9462

Although there are 18 orders of magnitude difference between the particle leakages between group 1 and group 16 (Fig. 11-13), both the hybrid method and the S_N (all S_N runs showed similar leakages) are in good agreement over the entire spectrum. In contrast, the pure MC calculation (not shown), with an approximately equivalent amount of computational effort, shows zero leakages past group 8 since, for analog Monte Carlo, the probability of particles reaching group 8 is vanishingly small for a reasonable number of histories. Thus, the hybrid Monte Carlo/S_N method is clearly preferrable to a standard analog Monte Carlo calculation.

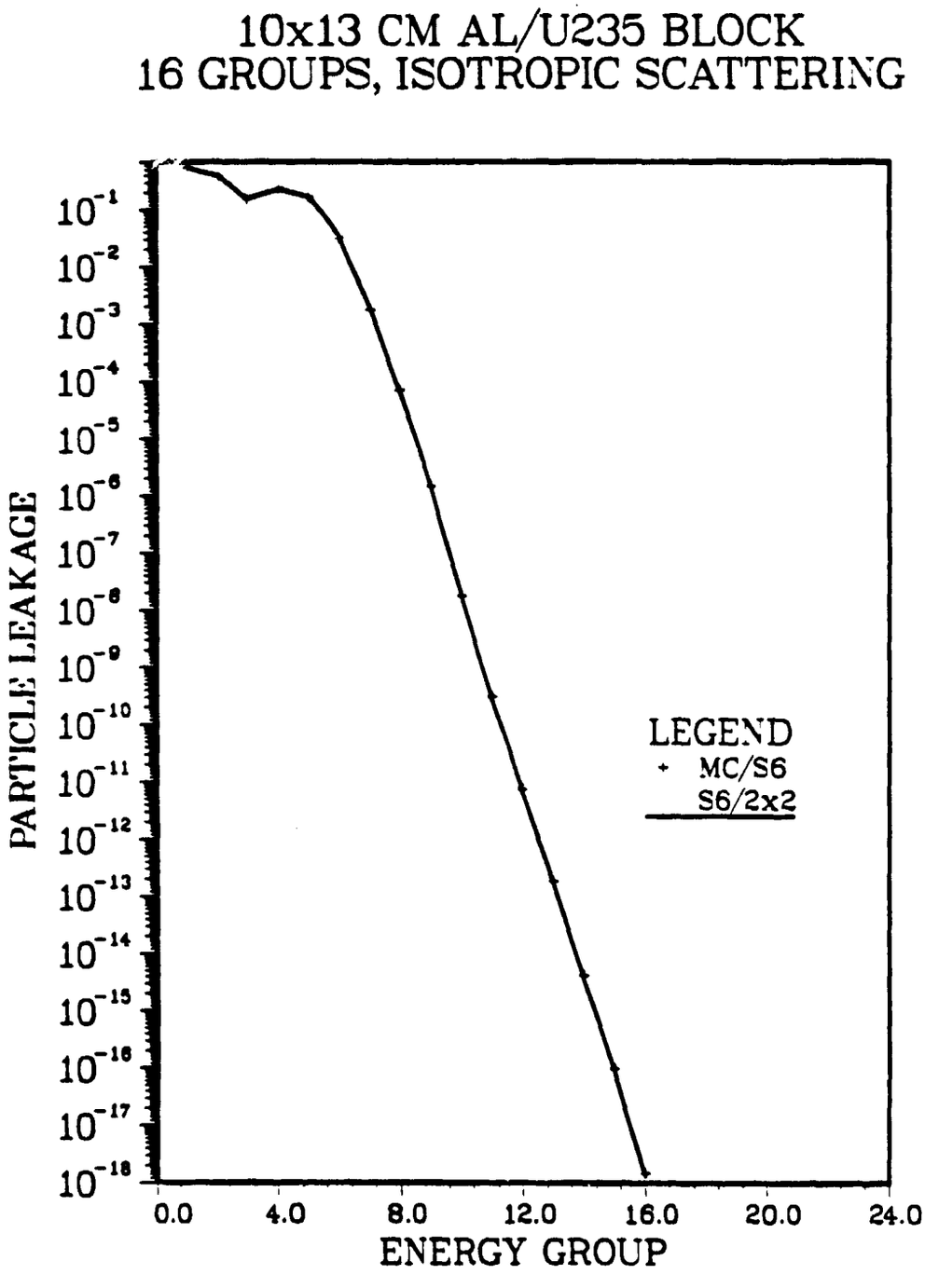

Figure 11-13 Particle Leakage, Al/U235 Problem

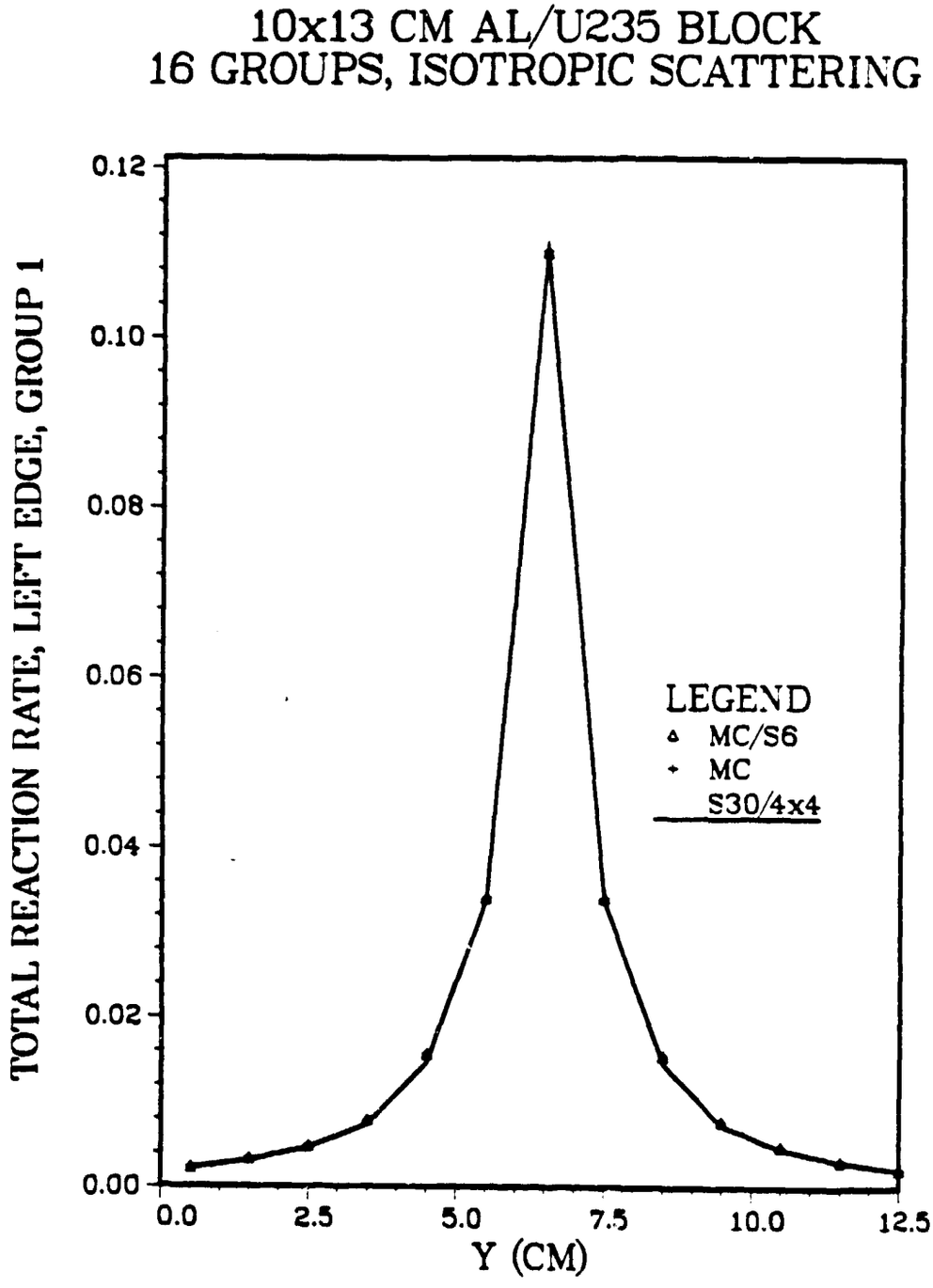

Figure 11-14 Total Reaction Rates, x = .5 Cm. Group 1

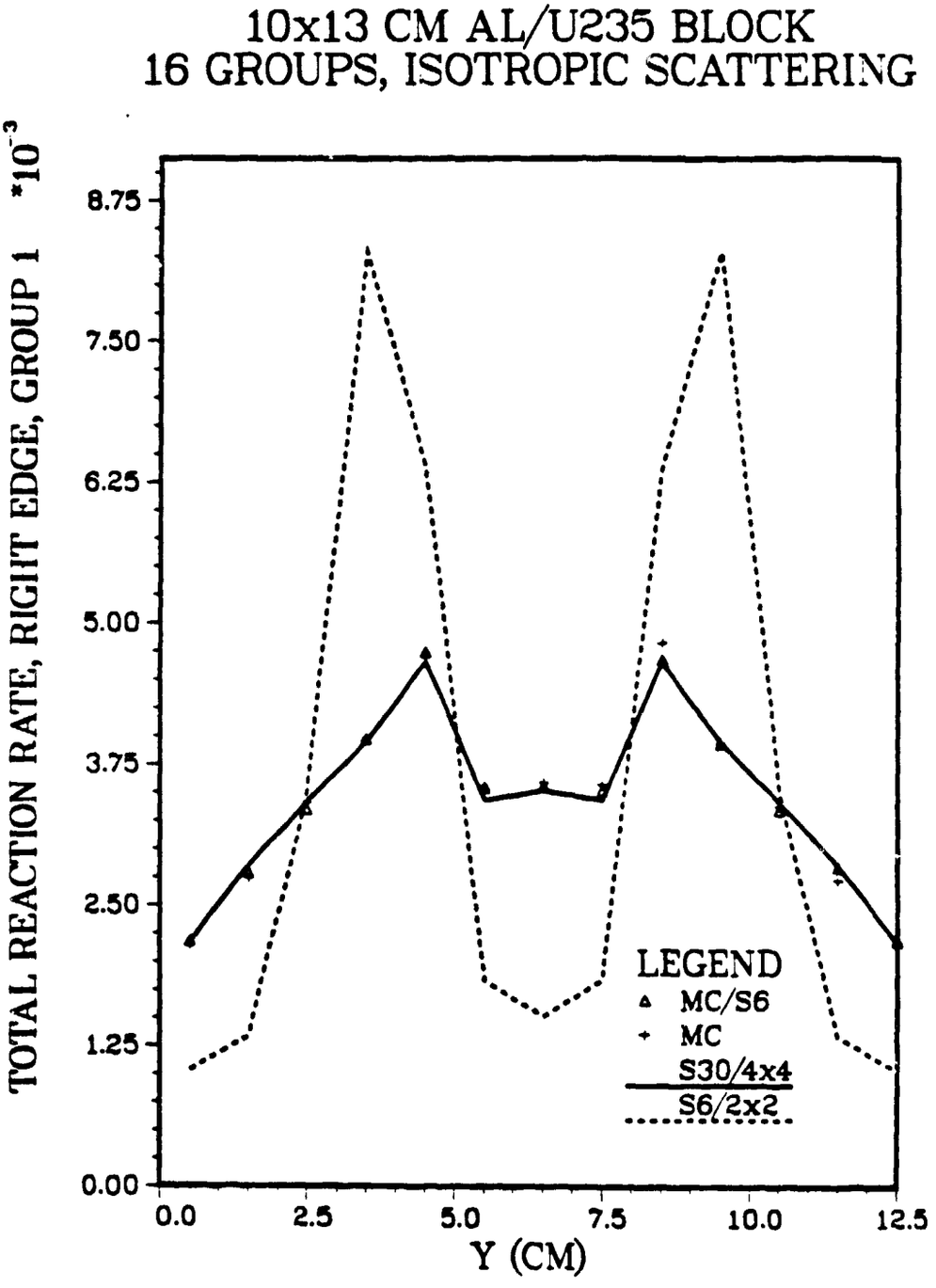

Figure 11-15 Total Reaction Rates, $x = 9.5$ Cm, Group 1

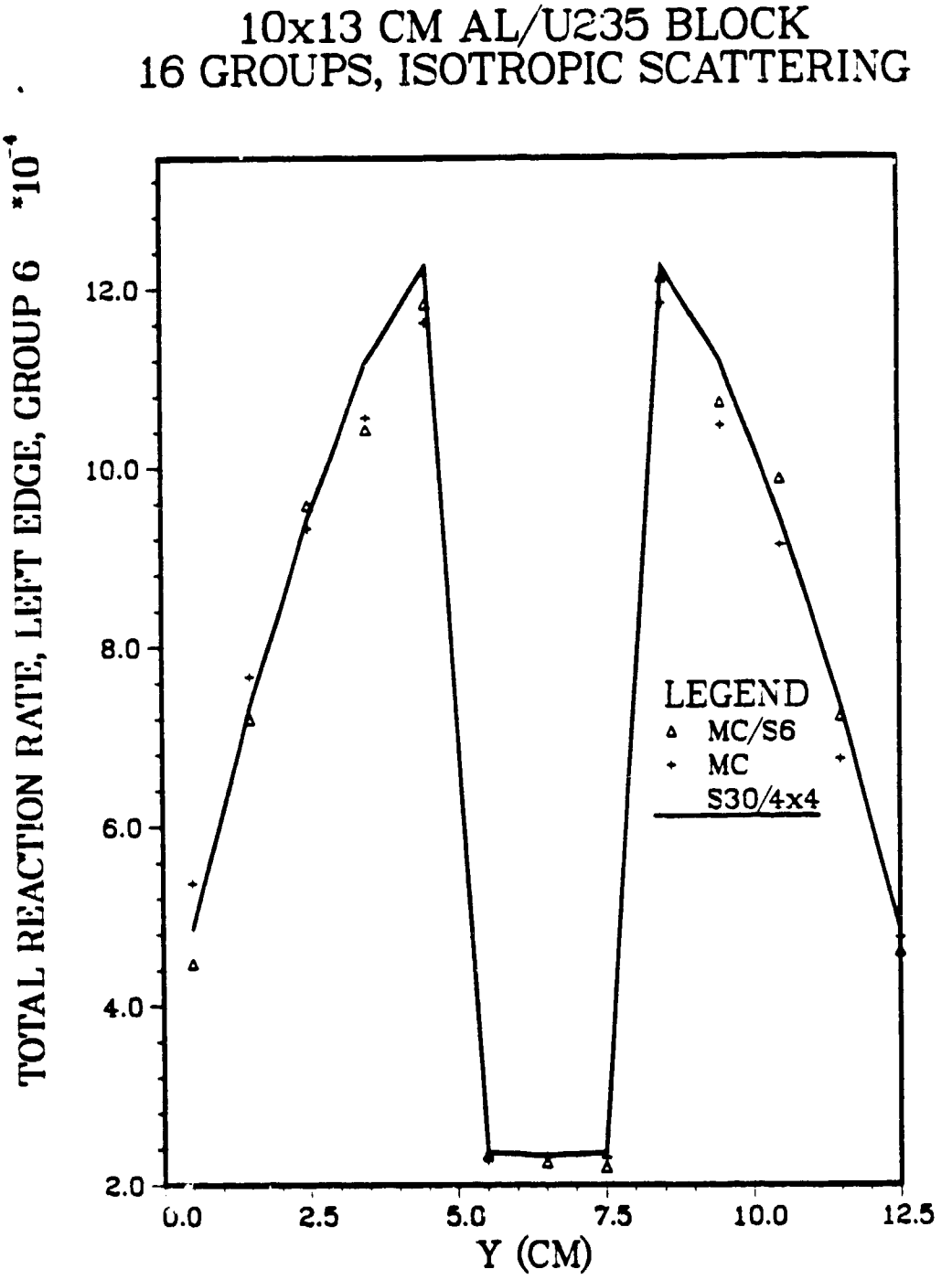

Figure 11-16 Total Reaction Rates, x = .5 Cm, Group 6

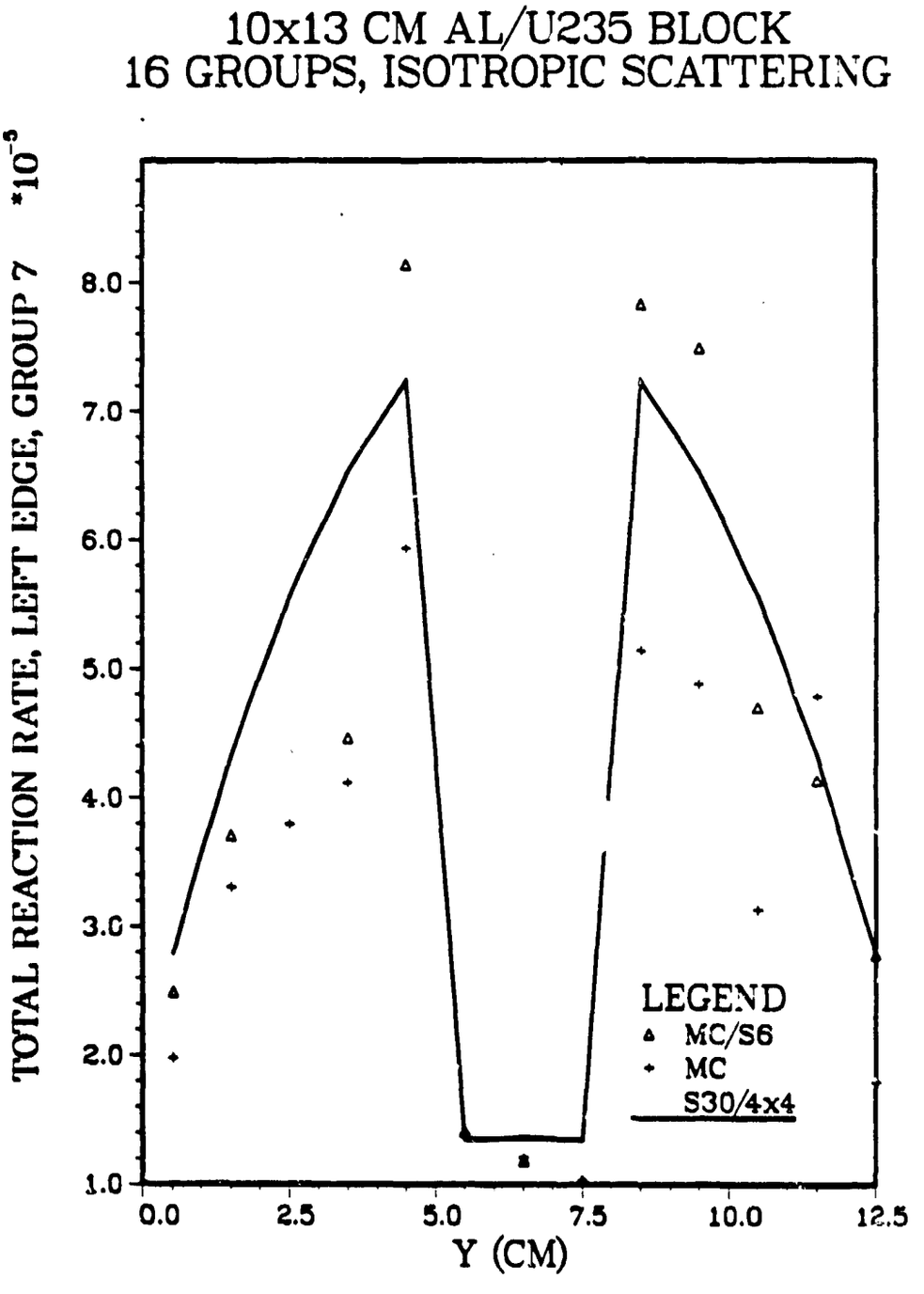

Figure 11-17 Total Reaction Rates, $x = .5$ Cm, Group 7

Surprisingly, all the S_N calculations were within about 15% of one another when determining the left edge total reaction rates for group 1, although the best agreement with the Monte Carlo and hybrid Monte Carlo/S_6 runs was obtained with the $S_{30}/4 \times 4$ calculation (Fig. 11-14). As expected, however, the S_N calculations exhibited severe ray effects when trying to determine the reaction rates at the right edge (Fig. 11-15), and a high quadrature order and small mesh size was required to match the hybrid results. The hybrid method mitigates any ray effects, of course, since the entire calculation for group 1 is performed by the Monte Carlo method.

By group 6, the S_N no longer suffers from ray effects, and all calculations, including the Monte Carlo and hybrid method, are in good agreement (Fig. 11-16). For group 7, however (Fig. 11-17), it is evident that statistical variations are beginning to severly affect the accuracy of both the analog Monte Carlo and hybrid Monte Carlo/S_N methods. It is interesting to note, however, that while both the Monte Carlo and the hybrid Monte Carlo/S_N calculations required approximately the same amount of computational time (reference Table 11-7 above), the hybrid method appears to match the S_N results better. Of couse, if a biased Monte Carlo method that increased the number of samples in lower energy groups was employed, it might reduce the variance of the results for a given computational time.

The advantage of the hybrid Monte Carlo/S_N method is most clearly shown at Table 11-7, where an S_{30} calculation with reduced mesh size is required to match the results of a hybrid S_6 calculation in determining the fission source. Note that the standard S_N method requires over three times the amount of compuatational time to match the results of the hybrid Monte Carlo/S_N method.

CHAPTER 12.
CONCLUSIONS AND FUTURE EFFORTS

The hypothesis of the hybrid Monte Carlo/S_N method is that, by coupling the two methods together in the same problem, we can solve certain types of problems more efficiently than either the Monte Carlo method or the S_N method can by themselves. While Alcouffe and Filippone demonstrated the feasibility of the hybrid approach for one-group problems in simplified $X-Y$ geometries with fictitous cross sections,[10,11] the ability of the hybrid method to handle the complexities posed by more realistic problems remained undetermined. The goal of this dissertation was to extend the capabilities of the hybrid method and examine its feasibility when applied to problems containing such complexities.

To allow the hybrid method to solve more complex problems, the following significant additions were made:

a. The ability to solve multigroup problems (including upscatter and fission) using response matrices with reasonable storage requirements and variable thickness boundary layers.

b. Improved performance of the diffusion synthetic accelerator within the S_N portion of the method.

c. Approximately a five-fold increase in the speed of the Monte Carlo calculations through vectorization.

d. Developing Monte Carlo tracking algorithms which allow the hybrid method to solve any problem geometry and material composition which can be input into TWODANT for $X - Y$ or $R - Z$ geometries.

e. Generalizing the hybrid Monte Carlo method to allow use of the multigroup S_N cross sections.

f. Incorporating anisotropic scattering into the Monte Carlo collision routines, and

g. Developing limited estimates of the errors in the Monte Carlo calculations.

With these additional capabilities, we were able to apply the hybrid Monte Carlo/S_N method to the benchmark problems of Chapter 11. For the first two benchmarks (Graphite Block and Uranium Rod), even though not ideal cases for the hybrid method, we saw that the basic premise of the method held. That is, we were able to obtain results with a low order S_N solver in the hybrid method that required a higher order (i.e., larger quadrature order and smaller mesh size) standard S_N solver. With the third benchmark (Aluminum-U235 Block), which was better suited for the hybrid method since it combined both optically thick and thin regions, we saw that the hybrid method clearly outperformed the S_N method.

Of the hybrid method enhancements listed above, the most important is the ability to handle multigroup problems. We have shown that the hybrid method

is capable of solving not just multigroup problems involving downscatter only, but those with upscatter and/or fission as well, and in reasonable amounts of time. With improved Monte Carlo methods (i.e., the use of non-analog Monte Carlo), we can expect orders of magnitude reductions in the computational time required by the Monte Carlo portion of the hybrid method. Since the Monte Carlo method usually requires the large majority of the computational time, this will further enhance the practicality of the hybrid method.

The second most important addition to the hybrid method is that we are able to implement it within TWODANT without adversely affecting the acceleration of the S_N inner iterations. Thus, the efficiency of the S_N methods in TWODANT are retained for pure downscatter and non-highly multiplying problems. While this may seem unimportant in relation to the Monte Carlo computational times, experience has shown that the expense of unaccelerated S_N calculations rapidly becomes prohibitive for complex problems in two dimensions. However, for highly multiplying problems, some means of accelerating the outer iterations in the hybrid method is still required, if it is desirable to apply the hybrid method to these types of problems.

It would also be desirable to use the hybrid method in problems involving both geometrically complicated and geometrically simple regions, where the Monte Carlo method would be used in the geometrically complicated region, and the S_N method in the geometrically simple. To include geometrically complicated (i.e., three dimensional) regions in the hybrid method, additional input to define the geometry and more extensive tracking routines would be required, in addition to some averaging algorithm for the transition between the three dimensional Monte

Carlo region and the two dimensional S_N. In addition, since the Monte Carlo region is no longer necessarily optically thin, more extensive variance reduction measures would be required. None of the above items would appear to require any substantial revisions in the hybrid method's theory, while the ability to handle geometricaly complicated problems method would greatly increase the number of "real" problems the hybrid method could be applied to.

Before the hybrid method could be applied to any "real" problems, however, the ability to accurately process cross sections including anisotropic scattering must be added. As previously pointed out, however, there are already several relatively straightforward methods in existence for doing so, any one of which could be employed with the hybrid method. It might also be noted that use of anisotropic scattering would enhance the desirability of the hybrid method, since the computation time for higher order S_N quadrature sets increases with the degree of anisotropy due to the increased number of moments that must be computed.

Perhaps the one unresolved issue in the application of the hybrid method to more complex problems is the ability to provide a rigorous error estimate in the quantities of interest, as all standard Monte Carlo codes currently do. As previously explained, standard variance estimation techniques are not practical to implement in the hybrid technique, and it is currently unclear as to whether some other means of estimating the variance is even possible. A more fruitful approach may be that of enhancing and expanding the error estimates already provided which, although not rigorous, at least provide some indication as to the precision of the Monte Carlo calculation.

Although work in the areas mentioned above is required before the hybrid method can be applied to more complex problems for which it is better suited, we have shown that the hybrid method is capable of solving multigroup problems in $X - Y$ or $R - Z$ geometries using the S_N method while mitigating standard S_N deficiencies such as ray effects and streaming effects. Those areas requiring additional work have been delineated and, with the possible exception of error estimates, it seems clear that there are no significant remaining obstacles towards development of a practical hybrid Monte Carlo/S_N code. Since, even with basically analog Monte Carlo techniques, isotropic scattering, and simplified test problems, the hybrid method generates run times of the same order as S_N calculations for comparable accuracy, we believe that further development of the hybrid Monte Carlo/S_N method is warranted, and that it will furnish an attractive alternative to existing solution methods for certain types of problems.

REFERENCES

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

[2] V. R. Cain, "Application of S_N Adjoint Flux Calculations to Monte Carlo Biasing", *Trans. Am. Nucl. Soc.*, **10**, 649 (1967).

[3] M. B. Emmett, C. W. Burgart, and T. J. Hoffman, "Domino, A General Purpose Code for Coupling Discrete Ordinates and Monte Carlo Radiation Transport Calculations", ORNL-4853 (1973).

[4] M. B. Emmett, "Domino-II, A General Purpose Code for Coupling DOT-IV Discrete Ordinates and Monte Carlo Radiation Transport Calculations", ORNL TM-7771 (1981).

[5] T. J. Seed, W. T. Urban, and D. J. Dudziak, "Coupled Monte Carlo/Discrete Ordinates Calculations for ETF Vacuum-Pumping Duct Shielding", *Trans. Am. Nucl. Soc.*, **38**, 649 (1981).

[6] H. Nakashima, H. Hasegawa, and Y. Kanda, "Neutron Streaming Analysis for Helical Geometry System, Heliotron-H Fusion Power Reactor", *Nucl. Sci. Tech.*, **23**, 287 (1986).

[7] S. N. Cramer and C. O. Slater, "Investigation of Radiation Effects in Hiroshima and Nagasaki Using a General Monte Carlo-Discrete Ordinates Coupling Scheme", *Trans. Am. Nucl. Soc.*, **61**, 138 (1990).

[8] R. E. Alcouffe, "A First Collision Source for Coupling Monte Carlo and Discrete Ordinates", Springer-Verlag Lecture Notes in Physics Vol. 240, 352 (1985).

[9] W. L. Filippone and B. D. Ganapol, "An S_N-Monte Carlo Hybrid Transport Method", *Trans. Am. Nucl. Soc.*, **41**, 649 (1982).

[10] W. L. Filippone and R. E. Alcouffe, "Linking Monte Carlo and S_N Via Response Matrices", LANL LA-10690-PR, 36 (1985).

[11] W. L. Filippone and R. E. Alcouffe, "The S_N/Monte Carlo Response Matrix Hybrid Method", *Nucl. Sci. Eng.*, **100**, 209 (1988).

[12] R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimensional, Diffusion Acclerated, Neutral-Particle Transport", LANL LA-10049-M, Rev. 1 (1984).

[13] R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analyses: Theory and Guidelines for Effective Use of Transport Codes", LANL LA-10983-MS (1987).

[14] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, John Wiley & Sons, Inc., New York (1984).

[15] B. G. Carlson and K. D. Lathrop, "Transport Theory - The Method of Discrete Ordinates", *Computing Methods in Reactor Physics*, Gordon and Breach Science Publishers, Inc., New York (1968).

[16] L. L. Carter and E. D. Cashwell, "Particle Transport Simulation with the Monte Carlo Method", ERDA TID-26607 (1975).

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

[18] Los Alamos Radiation Transport Group, "MCNP - A General Purpose Monte Carlo Code for Neutron and Photon Transport", LANL LA-7396-M, Rev. 2 (1986).

[19] W. L. Filippone, "Orders-of-Scattering Expansion Functions", *Trans. Am. Nucl. Soc.*, **34**, 649 (1980).

[20] W. L. Filippone, *Computational Methods of Engineering Science*, Unpublished, Tucson, Az (1986).

[1] F. W. Bobrowicz, J. E. Lynch, K. J. Fisher, and J. E. Tabor, "Vectorized Monte Carlo Photon Transport", LANL LA-9752-MS (1983).

[22] F. W. Bobrowicz, K. J. Fisher, and R. G. Brickner, "Vectorized Monte Carlo Neutron Transport", LANL LA-UR-84-1269 (1984).

[23] K. J. Fisher, "Vectorized Monte Carlo Radiation Transport", LANL LA-UR-86-3737 (1986).

[24] J. R. Lamarsh, *Nuclear Reactor Theory*, Addison-Wesley, Reading, MA (1966).

[25] R. E. Seamon, "Comparing Cross Sections for MCNP and TWODANT", LANL internal memorandum X-6:RES-88-445 to Distribution (October 27, 1988).

[26] L. L. Carter and C. A. Forest, "Transfer Matrix Treatments for Multigroup Monte Carlo Calculations - The Elimination of Ray Effects", *Nucl. Sci. Eng.*, **59**, 27 (1976).

[27] M. J. Berger, "Monte Carlo Calculations of the Penetration and Diffusion of Fast Charged Particles", *Methods of Computational Physics*, Academic Press, New York (1963).

[28] K. A. Brownlee, "Statistical Theory and Methodology in Science and Engineering", Robert E. Krieger Publishing Co., Inc., Malabar, FL (1984).

[29] A. Dubi, "On the Analysis of the Variance in Monte Carlo Calculations", *Nucl. Sci. Eng.*, **72**, 108 (1979).

[30] I. Lux, "On Efficient Estimation of Variances", *Nucl. Sci. Eng.*, **92**, 108 (1986).

[31] E. W. Larsen, "Diffusion-Synthetic Acceleration Methods for Discrete-Ordinates Problems", *Trans. Theory and Stat. Physics*, **13**, 107 (1984).

[32] E. W. Larsen, "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations. Part I: Theory", *Nucl. Sci. Eng.*, **82**, 47 (1982).

[33] R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations", *Nucl. Sci. Eng.*, **64**, 344 (1977).

[34] R. D. O'Dell, F. W. Brinkley, D. R. Marr, and R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport", LANL LA-9184-M, Rev (1989).

[35] R. E. Alcouffe, Los Alamos National Laboratory, Private Comunication (1989).