INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A super computer discrete ordinates method without observable ray effects or numerical diffusion

Monahan, Shean Patrick, M.S.
The University of Arizona, 1988
A SUPER COMPUTER DISCRETE ORDINATES METHOD WITHOUT
OBSERVABLE RAY EFFECTS OR NUMERICAL DIFFUSION

by

Shean Patrick Monahan

A Thesis Submitted to the Faculty of the
DEPARTMENT OF NUCLEAR AND ENERGY ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN NUCLEAR ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA

1988
STATEMENT BY THE AUTHOR

This thesis 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 or her 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: [Signature]

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

[Signature]

W. L. Filippone
Professor of Nuclear Engineering

Date
To my mother and father
ACKNOWLEDGMENTS

The author would like to express his thanks to Dr. William Filippone and to Dr. Barry D. Ganapol without whom this work would not have been possible. I would also like to acknowledge the great support afforded me by Dr. Robert L. Seale and the entire Department of Nuclear and Energy Engineering at the University of Arizona. Mention must also be made of the efforts of John Garth of the Rome Air Development Center in funding this project, many thanks.

Finally on a more personal note I would like express my sincere gratitude to my good friends who have suffered through the trials and tribulations of my education along with me. They have been and continue to be a constant source of support.
# TABLE OF CONTENTS

LIST OF TABLES ............................................. 7
LIST OF FIGURES ........................................... 8
ABSTRACT .................................................... 10
1. INTRODUCTION ........................................... 11
2. THEORY ................................................... 19
   2.1 The Spencer Lewis Electron Transport Equation .......... 19
   2.2 The $S_N$ Equations ................................ 21
   2.3 The SCS${}_N$ Equations ................................ 23
   2.4 Conventional $S_N$ .................................. 25
   2.5 Determining The Matrix Elements ......................... 28
3. THE ALGORITHM .......................................... 34
   3.1 The Equivalence of Eqs. (3.1) and (2.20) ............... 35
   3.2 Calculational Procedure ................................ 38
   3.3 Code Structure ...................................... 39
4. RESULTS ................................................ 41
   4.1 Ray Mitigation Properties ............................ 41
   4.2 Numerical Dispersion ................................ 43
   4.3 Computational Efficiency ............................. 46
5. CONCLUSIONS ............................................ 72
TABLE OF CONTENTS...continued

APPENDIX A. ........................................... 75
APPENDIX B. ........................................... 77
  B.1  i > i' ........................................... 77
  B.2  i' = i ........................................... 78
REFERENCES ........................................... 80
LIST OF TABLES

4.1 Comparison of Particles Streaming Through a Vacuum For $S_N$ and $SCS_N$  . 44
4.2 Comparison of Particles Streaming Through a Vacuum For SR and $SCS_N$ Methods  . 45
4.3 Total CPU Seconds For Scalar and Vector Processing Computers  . 47
# LIST OF FIGURES

1.1 General Categories of Solution Techniques For The Boltzmann Equation 18

2.1 Angular cone construction for $i > i' + 1$ 32

2.2 Angular cone construction for $i \leq i' + 1$. 33

4.1a $\text{SCS}_N$ Results For an Optically Thin Medium $M = 2$ 48

4.1b DD Results For an Optically Thin Medium $M = 2$ 49

4.2a $\text{SCS}_N$ Results For an Optically Thin Medium $M = 4$ 50

4.2b DD Results For an Optically Thin Medium $M = 4$ 51

4.3a $\text{SCS}_N$ Results For an Optically Thin Medium $M = 6$ 52

4.3b DD Results For an Optically Thin Medium $M = 6$ 53

4.4a $\text{SCS}_N$ Results For an Optically Thin Medium $M = 8$ 54

4.4b DD Results For an Optically Thin Medium $M = 8$ 55

4.5a $\text{SCS}_N$ Results For an Optically Thin Medium $M = 12$ 56

4.5b DD Results For an Optically Thin Medium $M = 12$ 57

4.6a $\text{SCS}_N$ Results for Isotropic Scattering For an Optically Thin Medium, $M=4$ 58

4.6b DD Results for Isotropic Scattering For an Optically Thin Medium, $M=4$ 59

4.7a $\text{SCS}_N$ Results for an Optically Thick Medium $M = 2$ 60

4.7b DD Results For an Optically Thick Medium $M = 2$ 61
### LIST OF FIGURES...continued

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8a</td>
<td>$SCS_N$ Results for an Optically Thick Medium $M = 4$</td>
<td>62</td>
</tr>
<tr>
<td>4.8b</td>
<td>DD Results For an Optically Thick Medium $M = 4$</td>
<td>63</td>
</tr>
<tr>
<td>4.9a</td>
<td>$SCS_N$ Results for an Optically Thick Medium $M = 6$</td>
<td>64</td>
</tr>
<tr>
<td>4.9b</td>
<td>DD Results For an Optically Thick Medium $M = 6$</td>
<td>65</td>
</tr>
<tr>
<td>4.10a</td>
<td>$SCS_N$ Results for an Optically Thick Medium $M = 8$</td>
<td>66</td>
</tr>
<tr>
<td>4.10b</td>
<td>DD Results For an Optically Thick Medium $M = 8$</td>
<td>67</td>
</tr>
<tr>
<td>4.11a</td>
<td>$SCS_N$ Results for an Optically Thick Medium $M = 12$</td>
<td>68</td>
</tr>
<tr>
<td>4.11b</td>
<td>DD Results For an Optically Thick Medium $M = 12$</td>
<td>69</td>
</tr>
<tr>
<td>4.12a</td>
<td>$SCS_N$ Results For a Monodirectional Source Incident on an Optically Thin Medium</td>
<td>70</td>
</tr>
<tr>
<td>4.12b</td>
<td>DD Results For a Monodirectional Source Incident on an Optically Thin Medium</td>
<td>71</td>
</tr>
</tbody>
</table>
ABSTRACT

A new discrete ordinates method designed for use on modern, large memory, vector and/or parallel processing super computers has been developed. The method is similar to conventional $S_N$ techniques in that the medium is divided into spatial mesh cells and that discrete directions are used. However, in place of an approximate differencing scheme, a nearly exact matrix representation of the streaming operator is determined. Although extremely large, this matrix can be stored on today's computers for repeated use in the source iteration. Since the source iteration is cast in matrix form it benefits enormously from vector and/or parallel processing, if available. Several test results are presented demonstrating the reduction in numerical diffusion and elimination of ray effects.
CHAPTER 1.
INTRODUCTION

The distribution of unbound subatomic particles and/or photons within a medium is of fundamental importance in many fields of research today. Although for the most part the primary interest in this area has been in the nuclear industry (radiation shielding, reactor design, etc.), an increasing number of applications are being found in other industries such as the oil, health, electronic, and even agricultural sciences. In most of these research fields, for the media and energies of concern, the particle/photon distribution is governed by the linear Boltzmann transport equation\textsuperscript{1}.

Since the original formulation of this equation many years ago, a myriad of solution techniques have been presented. Each solution technique varies greatly in approach and can be classified according to the particular form of the Boltzmann equation (integrodifferential, integral, or surface integral form) which they purport to solve. Although by no means complete, Figure 1.1 is a diagram of the general categories into which most of these solutions fall. Even though analytic solutions are possible for an ever growing number of cases, the bulk of the commonly used solutions are numerical in nature.

Of all the techniques illustrated in Fig. 1.1, the discrete ordinates\textsuperscript{2}, or $S_N$ method as it is known, is the most widely used method for obtaining deterministic
numerical solutions to the neutral particle transport equation. Relatively recently, it has also found application, with varying degrees of success, to both proton\textsuperscript{3} and electron\textsuperscript{4−8} transport equations. In addition to the details of implementation, the primary difference between the SN method and the others shown lies with the treatment of the angular dependence of the particle/photon distribution. Whereas in most of the other techniques the angular variable is treated as a continuous function, in the SN method angular space is discretized into a finite set of discrete directions. The integrodifferential equation is then evaluated directly but only along these discrete directions.

This approach has two very attractive benefits which account for the methods popularity; the derivation of the SN equations is very straightforward, and the algorithms generated are computationally (especially with regards to computer memory requirements) very efficient. At the time of its original development, the early 1950s, these benefits were of paramount importance, owing to the limited memories and speeds of the scalar computers available at the time. The fact that the method continues to be used and adapted in old and new applications, in this age of modern computing abilities, is a tribute to its power and versatility. However, as with any method which seeks to represent a continuous function in a discrete manner, the SN solutions have not been without problems; the most notable of which are numerical diffusion, and ray effects.

The first of these results primarily from spatial discretization of the streaming term of the transport equation, and varies in severity with the particular differencing scheme used. Although differencing schemes are based on physical arguments they tend to smear the streaming particles into spatial regions which are, in reality,
physically "inaccessible". This smearing can be quite severe near flux discontinuities. The method of streaming rays, a hybrid of the method of characteristics and finite difference $S_N$ techniques, was developed to eliminate this numerical dispersion, however it offers no help in mitigating ray effects.

Ray effects are anomalies that arise from the angular discretization of the streaming term of the transport equation, and are most noticeable in highly absorbing or low scattering media. For these media it is easy to see that the $S_N$ approximation will give incorrect results. In simple terms, the streaming contributions to the angular flux will be overestimated at mesh cells that can be connected to the source by one of the discrete directions, and underestimated for those cells which lie between adjacent discrete directions. The effect is to produce non-physical distortions which appear as "ripples" in the scalar flux. This can be a serious drawback particularly in problems where physical variations are expected and are confused or obscured by these non-physical rays.

To date the most common method of reducing ray effects is also the most obvious. Since the anomaly is the result of the discretization of the angular variable, common sense suggests that simply increasing the number of discrete directions would alleviate the difficulty. However, in many cases, this is unfeasible due to the increased memory requirements and computational times needed for acceptably accurate solutions. The seriousness of the situation is evidenced by the enormous effort expended, since ray effects were first reported, in developing alternative methods which reduce or eliminate this discretization error. A complete listing of all of these methods would be very extensive and not altogether useful. Instead a
brief review of some of the more successful techniques is presented along with their associated disadvantages; disadvantages which we believe justify this work.

In general most of the research in this area may be classified along one of three lines,

1. first collision source methods,

2. reformulation of the $S_N$ equations, or

3. hybrid techniques.

In the first of these approaches the uncollided flux and first collision source are calculated analytically. The collided flux is then determined using the $S_N$ method. The ray effects for the collided flux calculation are usually minor since the first collision source is often not as localized as the original source. Along the same lines it has also been suggested for more complicated cases, that the flux due to the first several collisions be obtained through a higher order approximation of some type (eg. $S_N$ with a large number of discrete directions). The flux thus obtained could then be used as an iteration source to obtain the total flux through a lower order $S_N$ approximation.

Several reformulations$^{11-15}$ of the discrete ordinates equations have also been proposed, which have yielded results with some degree of mitigation. In general the reformulations use some other type of angular approximation to generate discrete ordinate-like equations. In addition to the increased complexity and computational times required, most of these attempts reported only limited mitigation of the ray effect. The most successful of these appears to be the conversion of the $S_N$ equations,
through the use of fictitious sources, to spherical harmonics equations\textsuperscript{16}, which are not subject to the ray effects. The disadvantage of this method is the large increase in computation cost due to the slower convergence rate of the source iterations.

An S\textsubscript{N}-Monte Carlo hybrid method\textsuperscript{17} has also been developed which has been successful in eliminating ray effects without significantly increasing the total computer memory requirements. The mitigation effects of this technique are obtained by randomly selecting the angular directions for streaming at each iteration step, while performing the source integration on a preselected angular quadrature set. The effect is to multiply the number of discrete directions available to a particle by the number of iterations necessary for convergence. As a side effect however, the convergence rate is slowed considerably.

Along the same lines, a second type of S\textsubscript{N}-Monte Carlo hybrid method\textsuperscript{18} using response matrices\textsuperscript{19} has been developed. In this method the spatial area under consideration is subdivided into regions for which one of these techniques is best suited. The solutions obtained by the different methods are then linked at the regional boundaries by the use of the response matrices. Ray effects are kept to a minimum by performing calculations using the Monte Carlo technique in the optically thin regions.

Although many remedies have been suggested over the years, few have been successful in demonstrating a significant reduction, and even fewer in the total elimination of this artificial flux distortion. Those which have had success seem to be related by the common threads of increased complexity and computation time. The purpose of this paper is to present a new approach developed specifically
for today’s large memory, vector and/or parallel processing super computers. This super computer $S_N$ (SCS$_N$) method is also a hybrid, combining elements of multiple collision, integral and conventional $S_N$ methods. As the name implies, the method is based on the conventional $S_N$ technique in that the medium is divided into spatial mesh cells and discrete angles are used. However, instead of relying on conventional differencing schemes, elements of multiple collision and integral transport methods are combined to determine a nearly exact matrix representation of the streaming operator. This matrix is then used directly in the source iteration. The calculation of the matrix elements is facilitated by the introduction of a separate angular grid, comprised of ‘cones’, superimposed over the spatial mesh. The cones extend the space available to streaming particles to include the regions between the discrete directions, thereby eliminating ray effects. In addition, those elements of the matrix that correspond to streaming particles passing from one angular cone to another are easily identified and set equal to zero. This eliminates the numerical diffusion of streaming particles. Of course, some minimal numerical diffusion due to the spatial discretization of the scattering source could still be present. Although the use of cones is not a new idea$^{20}$, here they are used to relate all mesh cells of the discrete ordinates grid, as opposed to linking only those cells adjacent to one another.

Although the discovery and subsequent search for remedies for ray effects and numerical diffusion have, for the most part, been applied to the neutral particle transport equation, these effects are purely a result of the $S_N$ formulation itself, and are therefore present in any problem in which this classical formulation is used. Although our formulation is applied to the Spencer Lewis electron transport equation, the technique should be equally valid in the neutral particle case.
Beginning in Chapter 2 the Spencer-Lewis electron transport equation, along with its particular peculiarities is introduced. The discrete ordinates equations are then derived using a weighted residual method, after which the SCS$_N$ equations are developed, followed by a derivation of the conventional S$_N$ equations for comparison purposes. Chapter 2 concludes with a detailed explanation of the method whereby the elements of the operator are determined. In Chapter 3 the algorithm which takes advantage of the time-like behavior of the Spencer-Lewis equation is presented. Chapter 4 then compares the results of calculations obtained from conventional S$_N$ and SCS$_N$, revealing the elimination of ray effects and decreased numerical dispersion. A comparison of the computation times and efficiencies is also made. Finally Chapter 5 will present a summary of the advantages and disadvantages of the SCS$_N$ method along with suggestions on possible future work.
Figure 1.1 General Categories of Solution Techniques For The Boltzmann Equation
CHAPTER 2.
THEORY

In this chapter the Spencer-Lewis electron transport equation along with the its associated assumptions is introduced. The discrete ordinate formulation corresponding to this equation is then derived using a weighted residual method. The SCS$_N$ equations are then developed, followed by the conventional S$_N$ equations used for comparison. Finally the chapter concludes with a description of the the method for determining the inverse matrix form of the streaming operator, critical to the SCS$_N$ method.

2.1 The Spencer Lewis Electron Transport Equation

For certain situations, the one-dimensional transport of electrons in homogeneous media can be described by the Spencer-Lewis electron transport equation,$^{21,22}$

\[
\left[ \frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \sigma(s) \right] \phi(x, \mu, s) = \int_{-1}^{1} \sigma(s, \mu' \rightarrow \mu) \phi(x, \mu', s) d\mu' + q(x, \mu, s) \ (2.1)
\]
in which

\[ \phi(x, \mu, s) = \text{electron flux at position } x, \text{ path length } s, \]

in direction \( \mu \),

\[ \sigma(s) = \text{total (scattering) cross section}, \]

\[ \sigma(s, \mu' \rightarrow \mu) = \text{differential scattering cross section}, \]

and

\[ q(x, \mu, s) = \text{external electron source}. \]

Inherent in Eq. (1) are the following assumptions:

1. There are no externally applied magnetic or electrical fields

2. Fields created by the electrons themselves are negligible

3. Electron-electron and electron-nucleus scatters may be modelled as localized collisions.

4. The Continuous Slowing Down Approximation (CSDA) is valid.

With these assumptions it is possible to derive Eq. (2.1) from the energy and time dependent electron transport equation\(^2\).

According to the last assumption above there is a one-to-one correspondence between the distance traveled and the energy lost by an electron as it travels through a medium. As a result the path length variable \( s \) actually plays the role of an energy variable. More formally the CSDA models the collision process by decoupling the angular reorientation and energy loss associated with a collision into two separate
processes. Collisions are then only responsible for changing an electron's direction of travel and the energy loss is determined by the stopping power of the medium, $|\frac{dE}{ds}|$, which is a known function of energy. In actuality electrons lose energy in discrete amounts and so the CSDA is not an exact representation. Deviation from this model is known as straggling and is not addressed here.

2.2 The SN Equations

The following derivation is adapted from the generalized derivation presented by Filippone\textsuperscript{24}. It is included here for completeness and to explicitly introduce the assumptions and restrictions behind these equations.

If the flux in Eq. (2.1) is approximated by

$$\phi(x, \mu, s) = \sum_{m' = 1}^{M} \phi^{m'}(x, s)B^{m'}(\mu)$$  \hspace{1cm} (2.2)

where the $B^{m'}$ are a set of $M$ basis functions, the residuals associated with $\phi$ may be defined as

$$R(x, \mu, s) = \left[\frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \sigma(s)\right] \sum_{m' = 1}^{M} \phi^{m'}(x, s)B^{m'}(\mu) - \int_{-1}^{1} \sigma(s, \mu' \rightarrow \mu) \sum_{m' = 1}^{M} \phi^{m'}(x, s)B^{m'}(\mu') - q(x, \mu, s)$$  \hspace{1cm} (2.3)

In order to determine the expansion coefficients $\phi^{m'}$ the residual is forced to be orthogonal to a set of test functions, $[T^m(\mu); m = 1, 2, \ldots, M]$,

$$\int_{-1}^{1} T^{m*}(\mu)R(x, \mu, s)d\mu = 0.$$  \hspace{1cm} (2.4)
Eq. (2.3) may then be rewritten as

\[
\left[ \frac{\partial}{\partial s} + \sigma(s) \right] \sum_{m'=1}^{M} \phi^{m'}(x, s) \int_{-1}^{1} T^{m*}(\mu) B^{m'}(\mu) d\mu \\
+ \sum_{m'=1}^{M} \frac{\partial}{\partial x} \int_{-1}^{1} T^{m*}(\mu) \mu B^{m'}(\mu) d\mu \\
= \sum_{m'=1}^{M} \phi^{m'}(x, s) \int_{-1}^{1} \int_{-1}^{1} T^{m*}(\mu) \sigma(s, \mu' \rightarrow \mu) B^{m'}(\mu') d\mu' d\mu \\
+ \int_{-1}^{1} T^{m*}(\mu) q(x, \mu, s) d\mu
\]

(2.5)

where the asterisks denote complex conjugation.

By requiring the \( T^m \) and \( B^m \) functions to be biorthonormal with weight function \( 1 \) and biorthogonal with weight function \( \mu \) such that

\[
\int_{-1}^{1} T^{m*}(\mu) B^{m'}(\mu) d\mu = \delta_{mm'} \quad (6)
\]

\[
\int_{-1}^{1} T^{m*}(\mu) \mu B^{m'}(\mu) d\mu = \bar{\mu}^{m} \delta_{mm'}, \quad (2.7)
\]

Eq. (2.5) reduces to the desired discrete ordinates equations:

\[
[L_m] \phi^m(x, s) = \sum_{m'=1}^{M} S_{mm'} \phi^{m'}(x, s) + q^m(x, s)
\]

\[
m = 1, 2, ..., M
\]

(2.8)

in which

\[
[L_m] = \left[ \frac{\partial}{\partial s} + \mu^m \frac{\partial}{\partial x} + \sigma^m(s) \right]
\]

\[
S_{mm'} = \int_{-1}^{1} \int_{-1}^{1} T^{m*}(\mu) \sigma(s, \mu' \rightarrow \mu) B^{m'}(\mu') d\mu d\mu', \quad (2.10)
\]

\[
q^m(x, s) = \int_{-1}^{1} T^{m*}(\mu) q(x, \mu, s) d\mu,
\]

(2.11)

and
\[ \mu^m \approx \bar{\mu}^m = \int_{-1}^{1} T^{m*}(\mu) \mu B^m(\mu) d\mu \]  
(2.12)

where \( \mu^m \) is the \( m^{th} \) discrete direction.

To insure that positive values of the flux are obtained from Eq. (2.8) we use SMART scattering theory to model the highly anisotropic electron scattering kernel. With this formulation the \( S_{mm'} \) are replaced by \( \tilde{S}_{mm'} \), the elements of the SMART scattering matrix.

\[ [L_m] \phi^m(x,s) = \sum_{m'=1}^{M} \tilde{S}_{mm'} \phi^{m'}(x,s) + q^m(x,s) \]  
(2.13)

The \( \tilde{S}_{mm'} \) (with the aid of spatially independent analytic solutions and an appropriate choice of test and basis functions) can be chosen positive, and such that, a huge number of miniscule deflections is simulated by a lesser number of larger deflections. As seen below, our algorithm for solving Eq. (2.13) is positive and therefore negative fluxes can not occur. In other words, given a positive valued source, the algorithm structure guarantees that only positive valued fluxes will be calculated.

2.3 The SCSN Equations

Decomposition of the electron flux of Eq. (2.13) into its collided and uncollided components leads to the following representation:

\[ [L_m] \phi^m_0(x,s) = q^m(x,s) \]  
(2.14)

\[ [L_m] \phi^m_c(x,s) = \sum_{m'=1}^{M} \tilde{S}_{mm'} \phi^{m'}(x,s) \]  
(2.15)

where
\[ \phi^m_c(x,s) = \sum_{n=1}^{\infty} \phi^m_n(x,s) \] (2.16)

and \( \phi^m_n \) is the flux of particles having suffered \( n \) collisions. It is apparent from Eq. (2.14) that the inverse operator, \( [L_m]^{-1} \), is a streaming operator, transforming a source into the corresponding uncollided flux. The formulation we present here takes advantage of this knowledge to determine a nearly exact discretized form of this operator. It is this operator, when applied to a discretized form of Eq. (2.13), that yields solutions without observable ray effects or numerical diffusion.

The derivation of the SCS\(_N\) equations is extremely straightforward from this point. Discretization of both the path length variable \( s \), denoted by the subscript \( i \), and the spatial variable \( x \), denoted by the subscript \( j \), reduces Eqs. (2.14) and (2.15) to

\[ [L_m] \tilde{\phi}^m_o = \tilde{q}^m \] (2.17)

\[ [L_m] \tilde{\phi}^m_c = \sum_{m'=1}^{M} \tilde{S}_{mm'} [\tilde{\phi}^m_{c}(x,s) + \tilde{\phi}^m_{o}] \] (2.18)

in which \( \tilde{\phi}^m_r (r = o, c) \) has components \( \phi^m_{r(i,j)} \), \( \tilde{q}^m \) has components \( q^m_{(i,j)} \), and \( L_m \) corresponds to the discretized form of the original operator \( L_m \). Solving both Eqs. (17) and (18) for the uncollided and collided fluxes respectively yields the SCS\(_N\) equations,

\[ \tilde{\phi}^m_o = [L_m]^{-1} \tilde{q}^m \] (2.19)

\[ \tilde{\phi}^m_c(p+1) = [L_m]^{-1} \left[ \sum_{m'=1}^{M} \tilde{S}_{mm'} \tilde{\phi}^{m'}_c(p) + \tilde{Q}^m_1 \right] \] (2.20)

where

\[ \tilde{Q}^m_1 = \sum_{m'=1}^{M} \tilde{S}_{mm'} \tilde{\phi}^m_o \] (2.21)
and the superscript \( (p) \) indicates the \( p^{th} \) iterate. ¹ The basic principle behind the SCS\(_N\) method is to use Eq. (2.19), along with an independently derivable expression for the uncollided flux to determine the elements of the \([L_m]^{-1}\). Once determined the streaming operator is used in Eq. (2.20), which is solved by the standard source iteration method.

Before giving our expression for the \([L_m]^{-1}\) we show how these operators are approximated in conventional S\(_N\) calculations.

### 2.4 Conventional S\(_N\)

Discretization of the space and path length variables in Eq. (2.13), and integration over an arbitrary cell \( i j \), in which \( i \) represents the path length index, and \( j \) the spatial index, leads to,

\[
\frac{1}{\Delta s_i} \left( \phi_{(i+\frac{1}{2},j)}^m - \phi_{(i-\frac{1}{2},j)}^m \right) + \frac{\mu^m}{\Delta x_j} \left( \phi_{(i,j+\frac{1}{2})}^m - \phi_{(i,j-\frac{1}{2})}^m \right) + \sigma_s \phi_{(i,j)}^m = Q_{(ij)}^m \tag{2.22}
\]

where

\[
\phi_{(i\pm\frac{1}{2},j)}^m = \frac{1}{\Delta x_j} \int_{s_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \phi^m(x, s_{i \pm \frac{1}{2}}) ds \tag{2.23}
\]

\[
\phi_{(i,j\pm\frac{1}{2})}^m = \frac{1}{\Delta s_i} \int_{s_{i-\frac{1}{2}}}^{s_{i+\frac{1}{2}}} \phi^m(x_{j \pm \frac{1}{2}}, s) ds \tag{2.24}
\]

¹ For the relationship between the \( p^{th} \) iterate and the \( n^{th} \) collided flux \( \Phi_n^m \) see Appendix A, section A.1
Aside from the inaccuracies in $Q_{(i,j)}^m$, such as those inevitably due to the approximate angular integration of the scattering contribution, Eq. (2.22) is exact. However, assuming the incoming fluxes $\phi_{(i-\frac{1}{2},j)}^m$, and $\phi_{(i,j-\frac{1}{2})}^m$, are both known there is only one equation for the three unknowns $\phi_{(i+\frac{1}{2},j)}^m$, $\phi_{(i,j+\frac{1}{2})}^m$, and $\phi_{(i,j)}$. To obtain a solution, two additional equations are required and these are obtained by introducing a spatial differencing scheme. The simplest and most common of these, known as the diamond differencing approximation, assumes that the edge and center fluxes are related by,

$$\frac{\phi_{(i+\frac{1}{2},j)}^m + \phi_{(i-\frac{1}{2},j)}^m}{2} = \phi_{(i,j)}^m$$

(2.28)

and,

$$\frac{\phi_{(i,j+\frac{1}{2})}^m + \phi_{(i,j-\frac{1}{2})}^m}{2} = \phi_{(i,j)}^m.$$  

(2.29)

Although many other differencing schemes exist, such as linear discontinuous$^{20}$, quadratic$^{21}$, and linear characteristics$^{22}$, to our knowledge all are subjected to numerical diffusion and ray effects to some extent. To preserve as much simplicity as
possible, the above diamond differencing scheme and associated algorithm will be used for comparison to the SCSN formulation.

With this augmented set of equations the $\phi_{(i,j)}^m$ are determined using the standard source iteration technique,

$$\phi_{(i,j)}^{m(p+1)} = [L_m]^{-1} \bar{Q}^m(p)$$

(2.30)

where

$$\bar{Q}^m(p) = \sum_{m' = 1}^{M} \delta_{m m'} \phi_{(i,j)}^{m(p)} + \bar{q}^m,$$

(2.31)

$\phi_{(i,j)}^{m(p+1)}$ has as its components the $\phi_{(i,j)}^{m(p+1)}$, and the matrix elements of $[L_m]^{-1}$ are defined implicitly by Eqs. (2.22), (2.28), and (2.29).

In the conventional S_N algorithm the large matrix $[L_m]^{-1}$ is never explicitly determined. Instead the unknown fluxes $\phi_{(i,j)}^{m}$, $\phi_{(i+\frac{1}{2}, j)}^{m}$, and $\phi_{(i,j+\frac{1}{2})}^{m}$ are determined one cell at a time, sweeping across the spatial mesh grid in the direction of particle travel. This procedure has two important advantages that were essential for implementation on early day computers,

1) There is no need to calculate, nor to store, the large $[L_m]^{-1}$ matrix.

2) Since $[L_m]^{-1}$ represents an streaming operator many of its elements are zero. The matrix multiplication implied by Eq. (2.20), contains very few computations which contribute to the solution. The above formulation by sidestepping this matrix multiplication requires far fewer mathematical operations, an essential time saving feature for codes being executed on scalar machines.

However, there are also disadvantages in using a differencing scheme, such as diamond differencing, that are often all too apparent. The implicit calculation of
the operator \([L_m]^{-1}\), although rapid and very efficient in conserving storage, is the source of both ray effects and the numerical diffusion of streaming particles. Although the elements relating cells which are adjacent to a 'source' cell are determined (implicitly) fairly accurately, those corresponding to distant cells are obtained (implicitly) only after the repeated application of the spatial differencing scheme. This repeated application causes multiplication of discretization errors and many of the elements that are known to be zero, to take on positive or even negative values, dispersing particles into forbidden regions. Furthermore, matrix elements for cells lying along a discrete direction tend to be over-estimated causing ray effects. Since scattering tends to reduce the magnitude of the distant cell matrix elements it also reduces the severity of these two errors.

To reduce these discretization errors we present a method designed specifically with today's large memory, vector and/or parallel processing super computers in mind. As mentioned earlier the basic principles of this SCSN method are simple. Rather than employing approximations to close the set of discrete ordinates equations, we have chosen to implement Eq. (2.20) directly, by first explicitly calculating the elements of the \([L_m]^{-1}\) matrices. For the most part the problem is then reduced to one of determining these matrix elements, to an accuracy consistent with the original discretization scheme.

2.5 Determining The Matrix Elements

In discrete form the element \(\ell^{-1}_{m(i,j,\cdot \cdot \cdot i',j')}\) of \([L_m]^{-1}\) represents the uncollided contribution to cell \(ij\) from a monodirectional unit source of direction \(m\) in cell \(i'j'\). In
theory each of these matrix elements could be evaluated to any degree of accuracy. However, a large computational effort would not be justified considering the level of approximations already made. To preserve as much simplicity as possible without sacrificing accuracy significantly, we use instead a semi-analytic expression in conjunction with a geometric scheme based on physical arguments.

The contribution of uncollided flux from a unit source in the cell $i'j'$ (hereafter referred to as the source cell) to the entire path length step $i$, containing the cell $ij$ (hereafter referred to as the target cell), is first determined from the relation,

$$\phi_{i,o}^m = \left\{ \begin{array}{ll} \frac{1}{\sigma_i^m} - \frac{1.0 - e^{-\sigma_i^m \Delta s_i}}{\Delta s_i (\sigma_i^m)^2}, & \text{for } i' = i, \\ \exp \left[ - \sum_{k=i'+1}^{i-1} \frac{\sigma_k^m \Delta s_k}{\sigma_i^m} \right] \frac{(1.0 - e^{-\sigma_i^m \Delta s_i})}{\sigma_i^m \Delta s_i} & \text{for } i > i', \end{array} \right. \quad (2.32)$$

where $\Delta s_i$ is the size of the $i$th path length step. As Eq. (2.32) indicates $\phi_{i,o}^m$ is independent of the spatial discretization. Both of these expressions are derived by performing a balance of uncollided electrons over the path length $i$. (See Appendix B)

Once determined $\phi_{i,o}^m$, is then distributed to the individual spatial mesh cells by overlaying an angular cone, emanating from the source cell, on the spatial grid. For this angular cone we use one of two forms, depending on the relative distance of the target path length from the path length of the source cell.

Figure 2.1 illustrates the construction of the angular cone for the discrete direction $m$, used for target path length steps when $i > i' + 1$. The angular cone is generated by extending the region of influence of the discrete direction $m$ to include half of
the angular distance to each of the \( m + 1 \) and \( m - 1 \) directions. The area created by the intersection of this angular cone and the \( i \pm \frac{1}{2} \) boundaries of the target path length step defines a region over which \( \phi_{i,m}^n \) is to be distributed. The individual elements of the \([L_m]^{-1}\) are then calculated based on the ratio of area contained within a particular mesh cell to this total area,

\[
\ell_{m(ij-i'j')}^{-1} = \phi_{i,o}^m \left( \frac{\text{area of intersection within cell } ij}{\text{total area of intersection within path length } i} \right).
\]  

(2.33)

If a cell within the target path length is not intercepted by the angular cone the matrix element for that target cell is zero. In fact the majority of these elements are zero, as Fig. 2.1 clearly shows.

Figure 2.2 illustrates the construction of the equivalent angular cone for target path length steps for which \( i \leq i' + 1 \). Whereas the angular cone originated from the center of the source cell for the target path length steps above, it is now constructed so that the \( m \pm \frac{1}{2} \) boundaries of the cone originate from the centers of the spatial edges of the source cell. Once the area is defined, Eq. (2.33) is again used for calculating the individual elements.

The distinction between these two cases is made based on physical arguments. If the previous cell centered scheme were used throughout the discrete ordinates grid, the elements of the \([L_m]^{-1}\) for adjacent cells would be underestimated. In fact, a simple examination of Fig. 2.1 shows that for this scheme, regardless of the discrete direction, the elements of the \([L_m]^{-1}\) for the cells immediately above and below the source cell, \( i'j' \), would be zero. Since this is clearly incorrect, the edge centered scheme is more appropriate for calculating the nearby cell matrix elements. On the other hand, if the above cell edge center scheme (Fig. 2.2) were applied to the entire
grid, the extended regions of adjacent discrete directions would overlap for distant
cells overcompensating for use of discrete directions.

This semi-analytic geometric scheme described above for computing the elements of
the $[L_m]^{-1}$ is the fundamental mechanism whereby both ray effects and numerical
diffusion are removed. The use of discrete ordinates is compensated for by extend-
ing the region of influence of each discrete direction to include a portion of the
previously 'unseen' space between adjacent directions. The numerical diffusion has
been reduced by explicitly setting to zero those elements of $[L_m]^{-1}$ which should be
zero, preventing a streaming contribution to cells which are not within the angular
cone of $\mu^m$. 
Figure 2.1 Angular cone construction for \( i > i' + 1 \)
Figure 2.2 Angular cone construction for $i \leq i' + 1$. 
CHAPTER 3.
THE ALGORITHM

In its present form the direct use of Eqs. (2.19) and (2.20) would require the computation and storage of each of the \([L_m]^{-1}\) elements. For a discretization scheme in which \(il\) and \(jl\) represent the total number of path length steps and spatial mesh cells respectively, each of the \([L_m]^{-1}\) would contain \((il \times jl)^2\) terms. However, as indicated earlier, many of these elements are zero, and since \(s\) increases monotonically as the electrons transport, the calculation for each path length step can be done sequentially. It is therefore possible to develop a scheme whereby many of these zero elements can be excluded from the computation.

Since there can be no upscattering in path length, path length steps which have higher values of \(s\) cannot contribute particles to those with lower values. Those elements of \([L_m]^{-1}\) which correspond to the cells within those “downstream” path lengths are therefore zero. We avoid the unnecessary inclusion of these terms by adopting a path length by path length calculational procedure. For this formulation the path length specific form of Eq. (2.20) becomes,

\[
\bar{\varphi}_{i,c}^{m(p+1)} = [L_m]_{ii}^{-1} \sum_{m' = 1}^{M} \tilde{S}_{im'}^{i} \left( \bar{\varphi}_{i,c}^{m(p)} + \bar{Q}_{ii}^{m} \right)
\]  

(3.1)
where \( Q^m \) is the first collided source, and the path length index \( i \) is explicitly included for clarity. The path length specific operator \([L_m]^{-1}\) requires the computation and storage of only \( j_l \times j_l \) terms, since only those terms relating the spatial mesh cells of path length step \( i \) are necessary.

The elements of the uncollided flux, \( \tilde{\phi}^m \), are determined in a separate calculation by summing the individual contributions from the spatial mesh cells of the preceding path length steps, according to the same geometrical scheme described in Chapter 2. Once the angular cone of a particular source cell is constructed, computation is minimized by identifying and restricting the uncollided calculation to include only those spatial mesh cells within the cone.

3.1 The Equivalence of Eqs. (3.1) and (2.20)

To show the equivalence between the path length step specific form represented by Eq. (3.1), and that implied by Eq. (2.20), it will be necessary to manipulate Eqs. (2.19), (2.20), and (2.21), which are repeated here for convenience.

\[
\tilde{\phi}_o^m = [L_m]^{-1} \bar{q}^m, \quad (2.19)
\]

\[
\tilde{\phi}_c^{m(p+1)} = [L_m]^{-1} \left[ \sum_{m'=1}^M \tilde{S}_{mm'} \tilde{\phi}^{m'(p)}_c + \tilde{Q}_1^m \right], \quad (2.20)
\]

and

\[
\tilde{Q}_1^m = \sum_{m'=1}^M \tilde{S}_{mm'} \tilde{\phi}_c^{m'}, \quad (2.21)
\]
Substitution of Eq. (2.21) into Eq. (2.20), adding Eq. (2.19), and taking the limit as \( p \to \infty \), results in the relation,

\[
\tilde{\phi}^m = [L_m]^{-1} \left[ \sum_{m'=1}^{M} \tilde{S}_{mm'} \tilde{\phi}^{m'} + \tilde{q}^{m'} \right].
\]  

Let

\[
\tilde{\phi}^m = \begin{pmatrix}
\tilde{\phi}_1^m \\
\tilde{\phi}_2^m \\
\vdots \\
\tilde{\phi}_{il}^m
\end{pmatrix},
\]  

\[
\tilde{q}^m = \begin{pmatrix}
\tilde{q}_1^m \\
\tilde{q}_2^m \\
\vdots \\
\tilde{q}_{il}^m
\end{pmatrix},
\]  

\[
[L_m]^{-1} = \begin{pmatrix}
[L_m]^{-1}_{1,1} & [L_m]^{-1}_{1,2} & \cdots & [L_m]^{-1}_{1,il} \\
[L_m]^{-1}_{2,1} & [L_m]^{-1}_{2,2} & \cdots & [L_m]^{-1}_{2,il} \\
\vdots & \vdots & \ddots & \vdots \\
[L_m]^{-1}_{il,1} & [L_m]^{-1}_{il,2} & \cdots & [L_m]^{-1}_{il,il}
\end{pmatrix},
\]  

\[
[\tilde{S}_{mm'}] = \begin{pmatrix}
\tilde{S}_{1mm'} \\
\tilde{S}_{2mm'} \\
\vdots \\
\tilde{S}_{ilmn'}
\end{pmatrix}
\]

where \( il \) denotes the total number of path length steps. Each submatrix represents an \( i^{th} \) component of the total matrix and is composed of terms representing the individual spatial cells contained within the \( i^{th} \) path length step. For example, the submatrix for the \( i^{th} \) flux can also be represented as

\[
\begin{pmatrix}
\tilde{\phi}_1^m \\
\tilde{\phi}_2^m \\
\vdots \\
\tilde{\phi}_{il}^m
\end{pmatrix} = \begin{pmatrix}
\phi_{i,1}^m \\
\phi_{i,2}^m \\
\vdots \\
\phi_{i,il}^m
\end{pmatrix}
\]
where $jl$ is the total number of spatial mesh cells.

With the definitions of Eqs. (3.3) through (3.6) it is possible to write Eq. (3.2) for the $i^{th}$ component of the flux as,

$$\tilde{\phi}_i^m = \sum_{i' = 1}^{il} [L_m]_{ii'}^{-1} \left[ \sum_{m' = 1}^{M} \tilde{S}_{mm'} \tilde{\phi}_{i'}^{m'} + \tilde{q}_{i'}^{m} \right]. \tag{3.8}$$

It is now possible to separate Eq. (3.8) into a contribution from the $i^{th}$ path length step itself, and a contribution from all other path length steps $i' \neq i$,

$$\tilde{\phi}_i^m = [L_m]_{ii}^{-1} \left[ \sum_{m' = 1}^{M} \tilde{S}_{mm'} \tilde{\phi}_{i'}^{m'} + \tilde{q}_{i'}^{m} \right]$$

$$+ \sum_{i' = 1}^{i-1} [L_m]_{ii'}^{-1} \left[ \sum_{m' = 1}^{M} \tilde{S}_{mm'} \tilde{\phi}_{i'}^{m'} + \tilde{q}_{i'}^{m} \right]. \tag{3.9}$$

The summation of the second term of Eq. (3.9) has been restricted to include only path length steps which precede the one of interest, since $[L_m]_{ii'}^{-1} = 0$ for $i' > i$. This reflects the fact that path length variable $s$ increases monotonically as the electrons transport, as previously mentioned. In other words it is impossible for path length steps for which $i' > i$ to contribute to the $i^{th}$ path length step.

It is apparent from Eq. (3.9) that the $i^{th}$ flux may be divided into collided and uncollided terms such that,

$$\tilde{\phi}_i^m = \tilde{\phi}_{i,c}^m + \tilde{\phi}_{i,o}^m \tag{3.10}$$

where

$$\tilde{\phi}_{i,o}^m = [L_m]_{ii}^{-1} \tilde{q}_{i}^{m} + \sum_{i' = 1}^{i-1} [L_m]_{ii'}^{-1} \left[ \sum_{m' = 1}^{M} \tilde{S}_{mm'} \tilde{\phi}_{i'}^{m'} + \tilde{q}_{i'}^{m} \right] \tag{3.11}$$

and
\[ \tilde{\phi}_{i,c}^m = [L_m]_{ii}^{-1} \left[ \sum_{m'=1}^{M} \tilde{\xi}_{mm'}^{i} \tilde{\phi}_{i'}^{m'} \right] \]

\[ = [L_m]_{ii}^{-1} \left[ \sum_{m'=1}^{M} \tilde{\xi}_{mm'}^{i} \left( \tilde{\phi}_{i,c}^{m'} + \phi_{i,o}^{m'} \right) \right] \]

\[ = [L_m]_{ii}^{-1} \left[ \sum_{m'=1}^{M} \tilde{\xi}_{mm'}^{i} \tilde{\phi}_{i,c}^{m'} + \bar{Q}_{i_1}^{m} \right], \quad (3.12) \]

where it is apparent

\[ \bar{Q}_{i_1}^{m} = \sum_{m'=1}^{M} \tilde{\xi}_{mm'}^{i} \phi_{i,o}^{m'} \quad (3.13) \]

Equation (3.12) may now be solved iteratively,

\[ \tilde{\phi}_{i,c}^{m(p+1)} = [L_m]_{ii}^{-1} \sum_{m'=1}^{M} \tilde{\xi}_{mm'}^{i} \left( \tilde{\phi}_{i,c}^{m(p)} + \bar{Q}_{i_1}^{m} \right) \quad (3.1) \]

while \( \tilde{\phi}_{i,o}^{m}, \bar{Q}_{i_1}^{m} \) and the second term of Eq. (3.11) are solved for directly, making it unnecessary to explicitly calculate and store the \([L_m]_{ii}^{-1}\) for \( i' \neq i \). Furthermore since Eq. (3.12) is equivalent to Eq. (3.1) it has been proven that Eqs. (3.1) and (2.20) are also equivalent.

### 3.2 Calculational Procedure

Summarizing, the computational procedure for path length \( i \) is:

1) For each discrete direction \( m \), the \([L_m]_{ii}^{-1}\) are determined from the geometric scheme descibed earlier.

2.) Each of the \( \tilde{\phi}_{i,o}^{m} \) \((m = 1, M)\) are then found through an implicit implimentation of Eq. (3.11).
3.) The first collision source is formed from

$$\tilde{Q}_{i,1}^m = \sum_{m'=1}^{3} \tilde{\sigma}_{m'm}^i \tilde{\phi}_{i,o}^m$$  \hspace{1cm} (3.13)

for each discrete direction and substituted into Eq. (3.1).

4.) Once Eq. (3.1) has converged the total flux is obtained from

$$\tilde{\phi}_i^m = \tilde{\phi}_{i,c}^m + \tilde{\phi}_{i,o}^m$$  \hspace{1cm} (3.10)

The procedure is then repeated for each succeeding path length step until the calculations are complete.

### 3.3 Code Structure

The code is comprised of approximately 1700 lines of FORTRAN 77, of which approximately 40% was inherited from a previously written conventional discrete ordinates code. The inherited portions consisted mainly of memory allocation, input and output statements, and the SMART cross section generation routines. All computer runs were performed on a Digital Electronics Corporation S850, and Scientific Computing Systems SCS 40.

The most important feature of the SCSN code structure is the form of the source iteration. As implied by Eq. (3.1), the source iteration is cast in matrix form. Once the \([L_m]_{ii}^{-1}\) operator has been determined (Subroutine LIN), and the first collision source calculated, the program repeatedly calls the subroutine FLUXES which is the FORTRAN equivalent of Eq. (3.1). Once the new iterate has been
calculated the source is updated in subroutine UPDATE, and compared to the previous iterate. When the process has converged (IC=1) the program merely moves on to the next path length step and repeats the process. The actual coding for the source iteration along with the subroutine FLUXES is shown below.

```
DO 3000 II = 1, ITSMAX
CALL FLUXES(F, Q, GEOLIN, IL, JL, KL)
CALL UPDATE(F, Q, SCAT, Q1, ITS, IL, JL, KL, I)
IF(IC .EQ. 1) GOTO 400
3000 CONTINUE
```

```
SUBROUTINE FLUXES(F, Q, GEOLIN, IL, JL, KL)
DIMENSION GEOLIN(JL, JL, KL), F(JL, KL), Q(JL, KL)
DO 8100 K = 1, KL
DO 8100 J = 1, JL
SUM = 0.0
DO 4000 JO = 1, JL
4000 SUM = SUM + GEOLIN(J, JO, K) * Q(JO, K)
8100 F(J, K) = SUM
RETURN
END
```

The above code segment has been included only to illustrate the enormous benefit which can be obtained by execution on a vector and/or parallel processing machine.
CHAPTER 4.

RESULTS

In this chapter the performance of the SCSN algorithm is analyzed and compared to results obtained from a conventional SN solution using a diamond differencing (DD) scheme. The superiority of the new method is demonstrated with regards to both ray effects and numerical diffusion, and the computational speed and efficiency are also compared.

4.1 Ray Mitigation Properties

In order to determine the ray mitigation properties of the SCSN method it was necessary to test the algorithm in a situation for which the conventional SN technique is clearly subject to ray effects. To this end, five test cases were executed for an isotropic, 200 KeV electron source incident on the spatial center of an Aluminum slab. The number of discrete directions used was the only parameter which varied between cases, taking on values of 2, 4, 6, 8, and 12. The results of the SCSN calculations are shown in Figs. 4.1a-4.5a, and those of the DD method in Figs. 4.1b-4.5b. The mesh cell sizes corresponded to 3.7 mean-free-path squares, an optically thin medium.
The results of the DD calculations were as expected. Significant ray effects are noticeable in each of the low order quadrature cases and persist to some extent even into the $M = 12$ (Fig. 4.5b) case. An examination of Figs. (4.1a-4.5a) reveals that for the most part the SCSN method is not subject to these same effects. The physical variations which are present appear to be the result of scattering. This is born out by Figs. (4.6a) and (4.6b), which show results for a test case identical to that described above ($M=4$) with the exception that isotropic scattering was also assumed. The ray effects in the DD case (Fig. 4.6a) are still very prevalent, however the physical variations present in Fig. (4.6b) have disappeared.

To insure that this new method was capable of reproducing results for which the conventional $S_N$ can be considered reliable, the above test cases were repeated for an optically thick medium (37 mfp squares). Figures (4.7a-4.11a) and (4.7b-4.11b) illustrate the results obtained once again for angular quadratures of 2, 4, 6, 8, and 12. The remarkably persistent nature of the ray effects is illustrated by the $M = 2$ and 4 (Figs. 4.7b and 4.8b) of the conventional results. The SCSN counterparts to these results (Figs. 4.7a and 4.8a) show no such effects, although there does seem to be a scattering ray effect present in the $M = 4$ (Fig. 4.8a) case. For higher orders of quadrature the DD solution appears to remain constant, indicating that an accurate solution has been obtained. Comparison of Figs. (4.9a-4.11a) and (4.9b-4.11b) clearly shows that the SCSN technique is capable of reproducing results for which the DD solution can be considered accurate.
4.2 Numerical Dispersion

As explained earlier numerical diffusion is the artificial propagation of particles which results from the spatial discretization of the streaming term of the transport equation. The effect is most noticeable near flux discontinuities, such as along a wavefront, or along the edges of a beam transporting through a vacuum. In the next example we consider a monodirectional source ($\mu = .30$) incident on the $s = 0$ surface of a perfect vacuum. The purpose of this problem is to demonstrate the reduced numerical diffusion obtained with the (SCSN) method, as compared to the conventional technique. For ease of comparison the results of the conventional $S_N$ method, with and without a negative flux fixup (Table 4.1), along with the results obtained from the method of streaming rays (SR), with and without a ray effect mitigation routine (Table 4.2), are shown. (The SR method is included in the section as a standard, having been conceived precisely to eliminate numerical diffusion.)
Table 4.1
Comparison of Particles Streaming
Through a Vacuum For $S_N$ and SCSN Methods

<table>
<thead>
<tr>
<th></th>
<th>DD</th>
<th>-1.9</th>
<th>-0.8</th>
<th>4.5</th>
<th>3.3</th>
<th>-1.9</th>
<th>-5.8</th>
<th>-6.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DD</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
<th>6.0</th>
<th>7.0</th>
<th>8.0</th>
<th>9.0</th>
<th>10.0</th>
<th>11.0</th>
<th>12.0</th>
<th>13.0</th>
<th>14.0</th>
<th>15.0</th>
<th>16.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>-3.8</td>
<td>-2.9</td>
<td>5.7</td>
<td>11.2</td>
<td>12.7</td>
<td>11.6</td>
<td>9.5</td>
<td>7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>0.0</td>
<td>0.8</td>
<td>3.8</td>
<td>7.9</td>
<td>10.1</td>
<td>9.8</td>
<td>8.3</td>
<td>6.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DD</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>7.1</td>
<td>14.7</td>
</tr>
<tr>
<td>SC</td>
<td>9.2</td>
<td>19.0</td>
</tr>
</tbody>
</table>

The SCSN solution is clearly superior to both of the conventional $S_N$ solutions in representing the discontinuity along the edges of the beam. In fact the results of the conventional method without a negative flux fixup are completely worthless and are included only to emphasize the severity of the problem. Although conventional techniques using negative-flux-fixups do considerably better, they are nonlinear, and as is apparent in Table 4.1, not completely effective.
If the source particles are interpreted as a monodirectional beam then the (SR) results, without ray mitigation, represents the true solution, and all particles should remain within the dotted lines shown in Table 4.2. On the other hand, if the source particles are assumed to be distributed between $\mu = 0.2$, and $\mu = 0.4$, the edges of the cone for $\mu = 0.30$, then the flux should be contained within the solid lines of Tables 4.1 and 4.2, and the results are best represented by the $(SCS_N)$ method. For this interpretation of the source the (SR) method does better with its ray effect mitigation routine turned on. However, it is evident that this routine is only partially successful as the flux still peaks along the source direction $\mu = 0.30$.

**Table 4.2**

Comparison of Particles Streaming Through a Vacuum For SR and $SCS_N$ Methods

<table>
<thead>
<tr>
<th></th>
<th>SRM</th>
<th>SR</th>
<th>SC</th>
<th>SRM</th>
<th>SR</th>
<th>SC</th>
<th>SRM</th>
<th>SR</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SRM</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SRM</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SRM</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SRM</td>
<td>3.0</td>
<td>9.0</td>
<td>14.2</td>
<td>18.3</td>
<td>12.9</td>
<td>7.0</td>
<td>18.3</td>
<td>14.9</td>
<td></td>
</tr>
<tr>
<td>SR</td>
<td>3.0</td>
<td>9.0</td>
<td>14.2</td>
<td>15.7</td>
<td>12.3</td>
<td>7.0</td>
<td>13.2</td>
<td>12.6</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>3.0</td>
<td>9.2</td>
<td>19.0</td>
<td>19.6</td>
<td>13.3</td>
<td>7.3</td>
<td>13.8</td>
<td>13.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.9</td>
<td>11.0</td>
<td>5.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SRM</td>
<td>16.9</td>
<td>11.0</td>
<td>5.0</td>
<td>1.60</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SR</td>
<td>16.4</td>
<td>10.8</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>16.4</td>
<td>10.8</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16.9</td>
<td>11.0</td>
<td>5.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SRM</td>
<td>16.9</td>
<td>11.0</td>
<td>5.0</td>
<td>1.60</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SR</td>
<td>16.4</td>
<td>10.8</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>16.4</td>
<td>10.8</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
As a final example each code was again tested for cases identical to those discussed in the ray mitigation section. Once again the optically thin medium (3.7 mfp squares) was considered. However, the source was restricted to a single quadrature direction. Figures 4.12a and 4.12b are the results of this final comparison. It is interesting to note that for the DD solution both numerical diffusion and ray effects are clearly visible. The superiority of the SCSN solution is evident in the sharply defined beam edges and in the electron distribution within the confines of those edges.

4.3 Computational Efficiency

Table 4.3 is a comparison of the CPU (Central Processing Unit) seconds necessary to complete calculations for the DD and SCSN codes. Included with each quadrature direction is the time required on both scalar (DEC 8650) and vector (SCS 40) processing computers. In order to compare the two systems the ratio of times must be considered since each operates at a different processing frequency. The times listed correspond to the optically thin cases discussed earlier in regards to ray mitigation (Figs. 4.1a-4.5a and 4.1b-4.5b).

As expected the SCSN code was slower than its conventional counterpart. The vast discrepancy in total CPU time for the scalar computer can be attributed to two factors, the original calculation of the $[L_m]^{-1}$ elements, and the inefficient matrix multiplications within the source iteration. In fact for each of the cases shown in Table 4.3 there were 101 spatial mesh cells, requiring each of the $L_m^{-1}$ matrices to contain over 10,000 elements. The source iteration, in turn, required over $10,000 \times M$ mathematical multiplications each time it was performed.
Since the source iteration is ideally suited for vectorization, the SCS\textsubscript{N} method was expected to benefit significantly by execution on a vector machine. In fact the results of the vector computer calculations showed this to be the case. For the identical problems performed above, the ratio of CPU times required by each of the codes was considerably reduced.

Table 4.3
Total CPU Seconds For Scalar and Vector Processing Computers

<table>
<thead>
<tr>
<th>M</th>
<th>S\textsubscript{N}</th>
<th>SCS\textsubscript{N}</th>
<th>Ratio</th>
<th>S\textsubscript{N}</th>
<th>SCS\textsubscript{N}</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10.11</td>
<td>152.23</td>
<td>15.06</td>
<td>36.03</td>
<td>122.88</td>
<td>3.410</td>
</tr>
<tr>
<td>4</td>
<td>18.47</td>
<td>216.52</td>
<td>11.72</td>
<td>115.79</td>
<td>236.93</td>
<td>2.046</td>
</tr>
<tr>
<td>6</td>
<td>30.71</td>
<td>308.60</td>
<td>10.05</td>
<td>246.10</td>
<td>399.26</td>
<td>1.622</td>
</tr>
<tr>
<td>8</td>
<td>47.03</td>
<td>393.53</td>
<td>8.37</td>
<td>426.88</td>
<td>617.77</td>
<td>1.447</td>
</tr>
<tr>
<td>12</td>
<td>93.48</td>
<td>588.66</td>
<td>6.30</td>
<td>939.99</td>
<td>1200.8</td>
<td>1.277</td>
</tr>
</tbody>
</table>

Although the SCS\textsubscript{N} algorithm required more time to complete calculations, it was the more efficient of the two. As Figs. (4.1a-4.5a) and (4.1b-4.5b) illustrate, the higher accuracy of the streaming term approximation allows the SCS\textsubscript{N} method to yield better results, for a much lower order angular quadrature, than are obtainable by the conventional S\textsubscript{N} technique.
Figure 4.1a  $SCS_N$ Results For an Optically Thin Medium $M = 2$
Figure 4.1b  DD Results For an Optically Thin Medium $M = 2$
Figure 4.2a  SCS\textsubscript{N} Results For an Optically Thin Medium \( M = 4 \)
Figure 4.2b  DD Results For an Optically Thin Medium $M = 4$
Figure 4.3a  SCS$_N$ Results For an Optically Thin Medium $M = 6$
Figure 4.3b  DD Results For an Optically Thin Medium $M = 6$
Figure 4.4a  SCS$_N$ Results For an Optically Thin Medium $M = 8$
Figure 4.4b  DD Results For an Optically Thin Medium $M = 8$
Figure 4.5a  SCS\(_N\) Results For an Optically Thin Medium \(M = 12\)
Figure 4.5b  DD Results For an Optically Thin Medium $M = 12$
Figure 4.6a  SC$S_N$ Results for Isotropic Scattering For an Optically Thin Medium, $M=4$
Figure 4.6b  DD Results for Isotropic Scattering For an Optically Thin Medium, $M=4$
Figure 4.7a  $SCS_N$ Results for an Optically Thick Medium $M = 2$
Figure 4.7b  DD Results For an Optically Thick Medium $M = 2$
Figure 4.8a  $SCS_N$ Results for an Optically Thick Medium $M = 4$
Figure 4.8b  DD Results For an Optically Thick Medium $M = 4$
Figure 4.9a  SCS\textsubscript{N} Results for an Optically Thick Medium $M = 6$
Figure 4.9b  DD Results For an Optically Thick Medium $M = 6$
Figure 4.10a  SCS$_N$ Results for an Optically Thick Medium $M = 8$
Figure 4.10b  DD Results For an Optically Thick Medium $M = 8$
Figure 4.11a  $\text{SCS}_N$ Results for an Optically Thick Medium $M = 12$
Figure 4.12a  $SCS_N$ Results For a Monodirectional Source Incident on an Optically Thin Medium
Figure 4.12b  DD Results For a Monodirectional Source Incident on an Optically Thin Medium
CHAPTER 5.
CONCLUSIONS

As originally developed, the discrete ordinates method was optimized for use on scalar computers. Computer technology has improved to the point where it now seems feasible to design deterministic computer algorithms that yield results free of ray effect and numerical diffusion errors. The SCS$_N$ algorithm was designed to take advantage of several of the increased capabilities of today's super computers. The large memory capacity now available allows for the storage of the $[L_m]^{-1}$ matrices. These large matrices are calculated a single time, and to a high degree of accuracy, and then used repeatedly in the source iterations (see Eq. 3.1). As in most discrete ordinate transport calculations, the source iteration accounts for a major portion of the algorithm's computational requirements. The fact that this source iteration is cast in matrix form allows for a significant increase in computational speed as the result of vector processing.

Although to date no tests have been performed, the algorithm seems ideal for parallel processing. As the algorithm now exists, determining the elements of the $[L_m]^{-1}$ requires a major portion of the total computation time. However, since these elements are independent of one another, in principle each could be determined by a separate processor (assuming that enough were available). In addition the matrix
multiplication required by the source iteration could also benefit from execution on a parallel processing machine.

The $\text{SCS}_N$ method has several desireable features which include:

1) The elimination of the ray effect problem.

2) Negligible numerical diffusion.

3) A positive and linear computational algorithm (spatial differencing schemes using negative flux fixups are nonlinear).

4) A source iteration well suited for vector and parallel processing.

In addition, although applied here to simple geometries, the method is completely general, and it appears that extensions to more complicated systems are possible. Many of the transport solvers in use today have the further abilities of yielding the diffusion limit\textsuperscript{28} in optically thick media, and interacting well with acceleration schemes such as Diffusion Synthetic Acceleration\textsuperscript{29} (DSA). It is not known at this time if solvers of the $\text{SCS}_N$ type are also adaptable in this manner.
The two main disadvantages of this method are clearly,

1) The large memory requirements,

and

2) The large number of time consuming, nonproductive multiplications within the source iteration.

When considered with respect to the abilities of the modern super computing machines available today, these disadvantages become much less significant.
APPENDIX A

RELATION BETWEEN \( \bar{\phi}_N^M \) AND THE \( P^{TH} \) ITERATE

By definition,

\[
[L_m] \bar{\phi}_n^m = \sum_{m'=1}^{M} \tilde{S}_{mm'} \bar{\phi}_{n-1}^{m'}
\]  
(A.1)

where \( \bar{\phi}_n^m \) is the flux of particles having suffered \( n \) collisions. The total collided flux is obtained by summing both sides of Eq. (A.1) over all possible \( n \), that is,

\[
[L_m] \sum_{n=1}^{\infty} \bar{\phi}_n^m = \sum_{m'=1}^{M} \tilde{S}_{mm'} \left( \sum_{n=1}^{\infty} \bar{\phi}_{n-1}^{m'} \right).
\]  
(A.2)

Truncation of the infinite series at \( (p + 1) \) terms for numerical evaluation yields

\[
[L_m] \sum_{n=1}^{p+1} \bar{\phi}_n^m = \sum_{m'=1}^{M} \tilde{S}_{mm'} \left( \sum_{n=1}^{p+1} \bar{\phi}_{n-1}^{m'} \right).
\]  
(A.3)

Letting \( (n - 1) \rightarrow n \) on the right side of Eq. (A.3) so that \( (p + 1) \rightarrow p \) leads to,

\[
[L_m] \sum_{n=1}^{p+1} \bar{\phi}_n^m = \sum_{m'=1}^{M} \tilde{S}_{mm'} \left( \sum_{n=0}^{p} \bar{\phi}_{n}^{m'} \right),
\]  
(A.4)

or

\[
[L_m] \sum_{n=1}^{p+1} \bar{\phi}_n^m = \sum_{m'=1}^{M} \tilde{S}_{mm'} \left( \sum_{n=1}^{p} \bar{\phi}_{n}^{m'} \right) + \bar{Q}_1^m,
\]  
(A.5)
where

\[ \bar{Q}^m_1 = \sum_{m'=1}^{M} \phi^m_{0}. \]  

(A.6)

A comparison of Eq. (A.5) and Eq. (2.20),

\[ \bar{\phi}^{m(p+1)} = [L_m]^{-1} \left[ \sum_{m'=1}^{M} \bar{S}_{mm'} \bar{\phi}^{m(p)}_c + \bar{Q}^m_1 \right], \]  

(2.20)

clearly shows that the \( p^{th} \) iterate of Eq. (2.20) is actually equal to the sum given by,

\[ \bar{\phi}^{m(p)}_c = \sum_{n=1}^{p} \bar{\phi}^m_n. \]  

(A.7)

The iteration process implied by Eq. (2.20) is then said to converge when the contribution of the \( n^{th} \) collided flux does not significantly change the sum of the previous fluxes.
APPENDIX B
DERIVING EQUATION (2.32)

Equation (2.32)

\[
\phi_{i,o}^m = \begin{cases} 
\frac{1}{\sigma_{i}^m} - \frac{1.0-e^{-\sigma_{i}^m \Delta s_i}}{\Delta s_i (\sigma_{i}^m)^2}, & \text{for } i' = i, \\
\exp \left[ -\sum_{k=i'+1}^{i-1} \sigma_{k}^m \Delta s_k \right] \left( \frac{1.0-e^{-\sigma_{i'}^m \Delta s_{i'}}}{\sigma_{i'}^m \Delta s_{i'}} \right) \left( \frac{1.0-e^{-\sigma_{i}^m \Delta s_i}}{\sigma_{i}^m \Delta s_i} \right) & \text{for } i > i', 
\end{cases}
\]

is an expression for the uncollided flux in cell \( i \) due to a unit source located in cell \( i'j' \). Although two derivations, one for \( i > i' \), and another for \( i = i' \) are necessary, both are obtained by performing an electron balance on path length step \( i \).

7.1 \( i > i' \)

Consider a point source of unit magnitude emitting electrons of path length \( s \) located in cell \( i'j' \) of path length step \( i' \). The number of uncollided electrons entering path length step \( i \) from this source is

\[
\text{# Entering } i = \exp \left[ - \left( \sum_{k=i'+1}^{i-1} \sigma_{k}^m \Delta s_k - s\sigma_{i}'^m \right) \right]. \tag{B.1}
\]
Similarly the number of uncollided electrons exiting path length step \( i \) is

\[
\# \text{ Exiting } i = \exp \left[ - \left( \sum_{k=i'+1}^{i} \sigma_k^m \Delta s_k - s_i^m \right) \right]. \tag{B.2}
\]

From these two relations the number reacting in path length step \( i \) due to a distributed source in cell \( i'j' \) then must be

\[
\text{Total } \# \text{ Reacting in } i = \exp \left[ - \sum_{k=i'+1}^{i-1} \sigma_k^m \Delta s_k \right] \int_{\Delta s_i} \int_{\Delta x_{i'}} dx \exp [s_i^m] - \exp \left[ - \sum_{k=i'+1}^{i} \sigma_k^m \Delta s_k \right] \int_{\Delta s_i} \int_{\Delta x_{i'}} dx \exp [s_i^m] \tag{B.3}
\]

\[
= \exp \left[ - \sum_{k=i'+1}^{i-1} \sigma_k^m \Delta s_k \right] \frac{\Delta x_{i'}}{\sigma_i^{m'}} \left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right) - \exp \left[ - \sum_{k=i'+1}^{i} \sigma_k^m \Delta s_k \right] \frac{\Delta x_{i'}}{\sigma_i^{m'}} \left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right) \tag{B.4}
\]

\[
= \Delta x_{i'} \exp \left[ - \sum_{k=i'+1}^{i-1} \sigma_k^m \Delta s_k \right] \frac{\left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right)}{\sigma_i^{m'}} \left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right) \tag{B.5}
\]

Since the number reacting is also given by \( \phi_{i,o}^m \Delta x_j \Delta s_i \sigma_i^m \), solving for the flux yields the desired relation,

\[
\phi_{i,o}^m = \exp \left[ - \sum_{k=i'+1}^{i-1} \sigma_k^m \Delta s_k \right] \frac{\left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right)}{\sigma_i^{m'}} \left( 1.0 - e^{-\sigma_i^m \Delta s_i} \right) \tag{B.6}
\]

assuming \( \Delta x_{i'} = \Delta x_j \).
7.2 \( i' = i \)

For this case the expressions are even simpler. The total number of particles appearing in path length step \( i \) as the result of a unit source in cell \( ij \) is

\[
\# \text{Appearing in } i = \int_{\Delta x_j} dx \int_{\Delta s_i} ds = \Delta x_j \Delta s_i
\]

(B.7)

The number exiting now becomes:

\[
\# \text{Exiting } i = \exp \left[ -\Delta s_i \sigma_i^m \right] \int_{\Delta x_j} dx \int_{\Delta s_i} ds e^{\sigma_i^m} = \frac{1.0 - e^{\sigma_i^m \Delta s_i}}{\sigma_i^m}
\]

(B.8)

Therefore the total number reacting in path length step \( i \) is

\[
\phi_{i,o} \Delta x_j \Delta s_i \sigma_i^m = \Delta x_j \Delta s_i - \Delta x_j \frac{1.0 - e^{\sigma_i^m \Delta s_i}}{\sigma_i^m}
\]

or solving for the flux

\[
\phi_{i,o}^m = \frac{1}{\sigma_i^m} \frac{1.0 - e^{\sigma_i^m \Delta s_i}}{\Delta s_i (\sigma_i^m)^2}
\]

(B.9)
REFERENCES


