ADVANCED QUADRATURE SETS, ACCELERATION AND PRECONDITIONING TECHNIQUES FOR THE DISCRETE ORDINATES METHOD IN PARALLEL COMPUTING ENVIRONMENTS

By

GIANLUCA LONGONI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2004

Copyright 2004

by

GIANLUCA LONGONI

I dedicate this research work to Rossana and I thank her for the support and affection demonstrated to me during these years in college. This work is dedicated also to my family, and especially to my father Giancarlo, who always shared my dreams and encouraged me in pursuing them.

"Only he who can see the invisible, can do the impossible."
By Frank Gaines

ACKNOWLEDGMENTS

The accomplishments achieved in this research work would have not been possible without the guidance of a mentor such as Prof. Alireza Haghighat; he has been my inspiration for achieving what nobody else has done before in the radiation transport area.

I wish to thank Prof. Glenn E. Sjoden for his endless help and moral support in my formation as a scientist. I also express my gratitude to Dr. Alan D. George, for providing me with the excellent computational facility at the High-Performance Computing and Simulation Research Lab. I am thankful to Prof. Edward T. Dugan for his useful suggestions and comments, as well as to Dr. Ray G. Gamino from Lockheed Martin – KAPL and Dr. Joseph Glover, for being part of my Ph.D. committee. I also thank UFTTG for the interesting conversations regarding radiation transport physics and computer science.

I am also grateful to the U.S. DOE Nuclear Education Engineering Research (NEER) program, the College of Engineering, and the Nuclear and Radiological Engineering Department at the University of Florida, for supporting the development of this research work.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

ADVANCED QUADRATURE SETS, ACCELERATION AND PRECONDITIONING TECHNIQUES FOR THE DISCRETE ORDINATES METHOD IN PARALLEL COMPUTING ENVIRONMENTS

By

Gianluca Longoni

December 2004

Chair: Alireza Haghighat
Major Department: Nuclear and Radiological Engineering

In the nuclear science and engineering field, radiation transport calculations play a key-role in the design and optimization of nuclear devices. The linear Boltzmann equation describes the angular, energy and spatial variations of the particle or radiation distribution. The discrete ordinates method (S_N) is the most widely used technique for solving the linear Boltzmann equation. However, for realistic problems, the memory and computing time require the use of supercomputers. This research is devoted to the development of new formulations for the S_N method, especially for highly angular dependent problems, in parallel environments. The present research work addresses two main issues affecting the accuracy and performance of S_N transport theory methods: quadrature sets and acceleration techniques.

New advanced quadrature techniques which allow for large numbers of angles with a capability for local angular refinement have been developed. These techniques have

been integrated into the 3-D S_N PENTRAN (Parallel Environment Neutral-particle TRANsport) code and applied to highly angular dependent problems, such as CT-Scan devices, that are widely used to obtain detailed 3-D images for industrial/medical applications.

In addition, the accurate simulation of core physics and shielding problems with strong heterogeneities and transport effects requires the numerical solution of the transport equation. In general, the convergence rate of the solution methods for the transport equation is reduced for large problems with optically thick regions and scattering ratios approaching unity. To remedy this situation, new acceleration algorithms based on the Even-Parity Simplified S_N (EP-SS_N) method have been developed. A new stand-alone code system, PENSSn (Parallel Environment Neutral-particle Simplified Sn), has been developed based on the EP-SS_N method. The code is designed for parallel computing environments with spatial, angular and hybrid (spatial/angular) domain decomposition strategies. The accuracy and performance of PENSSn has been tested for both criticality eigenvalue and fixed source problems.

PENSSn has been coupled as a preconditioner and accelerator for the S_N method using the PENTRAN code. This work has culminated in the development of the Flux Acceleration Simplified Transport (FAST©) preconditioning algorithm, which constitutes a completely automated system for preconditioning radiation transport calculations in parallel computing environments.

CHAPTER 1
INTRODUCTION

1.1 Overview

In the nuclear engineering field, particle transport calculations play a key-role in the design and optimization of nuclear devices. The Linear Boltzmann Equation (LBE) is used to describe the angular, energy and spatial variations of the particle distribution, i.e., the particle angular flux.[1] Due to the integro-differential nature of this equation, an analytical solution cannot be obtained, except for very simple problems. For real applications, the LBE must be solved numerically via an iterative process. To solve large, real-world problems, significant memory and computational requirements can be handled using parallel computing environments, enabling memory partitioning and multitasking.

The objective of this dissertation is to investigate new techniques for improving the efficiency of the of the S_N method for solving problems with highly angular dependent sources and fluxes in parallel environments. In order to achieve this goal, I have investigated two major areas:

1. Advanced quadrature sets for problems characterized with highly angular dependent fluxes and/or sources.
2. Advanced acceleration/preconditioning algorithms.

1.2 The Linear Boltzmann Equation

The LBE is an integro-partial differential equation, which describes the behavior of neutral particle transport. The Boltzmann equation, together with the appropriate boundary conditions, constitutes a mathematically well-posed problem having a unique solution. The solution is the distribution of particles throughout the phase space, i.e.,

space, energy, and angle. The distribution of particles is, in general, a function of seven independent variables: three spatial, two angular, one energy, and one time variable. The time-independent LBE in its general integro-differential form[1] is given by Eq. 1.1.

$$\begin{aligned}\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},E,\hat{\Omega})+\sigma_t(\vec{r},E)\psi(\vec{r},E,\hat{\Omega}) &= q_{ext}(r,E,\hat{\Omega})\\ +\int_0^\infty dE'\int_{4\pi} d\Omega'\sigma_s(\vec{r},E'\to E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},E',\hat{\Omega}') &\\ +\frac{\chi(E)}{4\pi}\int_0^\infty dE'\int_{4\pi} d\Omega'\nu\sigma_f(\vec{r},E')\psi(\vec{r},E',\hat{\Omega}'). &\end{aligned} \tag{1.1}$$

In Eq. 1.1, I have defined the following quantities:

ψ : Angular flux [particles/cm^2/MeV/sterad/sec] .
$\vec{r}$: Particle position in a 3-D space [cm].
E: Particle energy [MeV].
$\hat{\Omega}$: Particle direction unit vector.
σ_t : Macroscopic total cross-section [1/cm/MeV].
q_{ext} : External independent source [particles/cm^3/MeV/sterad/sec] .
σ_s : Macroscopic double-differential scattering cross-section [1/cm/sterad/MeV].
χ : Fission spectrum [1/MeV].
ν : Average number of neutrons generated per fission.
σ_f : Macroscopic fission cross-section [1/cm/MeV].

1.3 Advanced Angular Quadrature Sets for the Discrete Ordinates Method

The discrete ordinates method (S_N) is widely used in nuclear engineering to obtain a numerical solution of the integro-differential form of the Boltzmann transport equation. The method solves the LBE for a discrete number of ordinates (directions) with associated weights.[2] The combination of discrete directions and weights is referred to as quadrature set.[3] The major drawback of the S_N method is the limited number of directions involved, which, in certain situations, may lead to the so called *ray-effects*, which appear as unphysical flux oscillations. In general, this behavior occurs for problems with highly

angular dependent sources and/or fluxes, or when the source is localized in a small region of space, in low density or highly absorbent media.

In the past, several remedies for ray-effects have been proposed; the most obvious one is to increase the number of directions of the quadrature set, or equivalently the S_N order. However, this approach can lead to significant memory requirements and longer computational times. Carlson and Lathrop proposed a number of remedies[4-5] for ray-effects based on specialized quadrature sets, which satisfy higher order moments of the direction cosines. Remedies based on first-collision approximations have also been investigated.[3] The source of particles generated from first-collision processes is often less localized than the true source; hence, the flux due to this source is usually less prone to ray-effects than the flux from the original source. If the true source is simple enough, analytic expressions can be obtained for the uncollided flux and used to produce a first collision source; however, for general sources and deep-penetration problems, this method can be very time-consuming. An alternative approach[6] is to expand the angular flux in terms of spherical harmonics (P_N). The P_N method does not suffer from ray-effects, because the angular dependency in the angular flux is treated using continuous polynomial functions. However, the P_N method has found limited applicability due to its computational intensive requirements.

One of the most widely used techniques for generating quadrature sets is the level-symmetric[3] (LQ_N) method; however, the LQ_N method yields negative weights beyond order S_{20}. In problems with large regions of air or highly absorbent materials, higher order (>20) quadrature sets are needed in order to mitigate ray-effects; therefore, it is necessary to develop techniques which allow for higher order or biased quadrature sets.

In the past, different techniques have been investigated to remedy this issue. The equal weight quadrature set (EQ_N), developed by Carlson,[7] is characterized by positive weights for any S_N order. Other quadrature sets have been derived, by relaxing the constraints imposed by the LQ_N method;[2] for this purpose, the Gauss quadrature formula and Chebyshev polynomials have been used for one-dimensional cylindrical or two-dimensional rectangular geometries.[4] In a recent study, the uniform positive-weight quadrature sets[8] (UE_N and UG_N) have been derived. The UE_N quadrature set is derived by uniformly partitioning the unit sphere into the number of directions defined by the S_N order while the UG_N quadrature set selects the ordinates along the *z*-axis as roots of Legendre polynomials.

A new biasing technique, named *Ordinate Splitting* (OS), has been developed[9] for the Equal Weight (EW) quadrature set; the OS technique is a method which is suitable to solve problems in which the particle angular flux and/or source are peaked along certain directions of the unit sphere. The idea is to select a direction of flight of the neutron and split it into a certain number of directions of equal weights, while conserving the original weight. This new biasing technique has been implemented in the PENTRAN code[10] and it has been proven very effective for medical physics applications such as CT-Scan devices.[11-15]

In this research work, I have developed new quadrature sets[11-12] based on Legendre (P_N) and Chebyshev (T_N) polynomials. The Legendre-Chebyshev (P_N-T_N) quadrature set is derived by setting the polar angles equal to the roots of the Legendre polynomial of order *N*, and the azimuthal angles are calculated by finding the roots of the Chebyshev polynomials.

The Legendre Equal-Weight (P_N-EW) quadrature set is derived by choosing the polar angles as the roots of the Legendre polynomial of order *N*, while the azimuthal angles are calculated by equally partitioning a 90 degree angle. The set of directions is then arranged on the octant of the unit sphere similar to the level-symmetric triangular pattern. The main advantage of these new quadrature sets is the absence of negative weights for any S_N order, and their superior accuracy compared to other positive schemes such as equal weight quadrature sets.

Moreover, I have developed a new refinement technique, referred to as Regional Angular Refinement (RAR), which leads to a biased angular quadrature set.[13-15] The RAR technique consists of fitting an auxiliary quadrature set in a region of the unit sphere, where refinement is needed. These quadrature sets have been applied successfully to large problems, such as a CT-Scan device used for medical/industrial imaging[9] and a Time-of-Flight (TOF) experiment simulation.[16] The benefit of using biased quadrature sets is to achieve accurate solutions for highly angular dependent problems with reduced computational time.

1.4 Advanced Acceleration Algorithms for the S_N Method on Parallel Computing Environments

Radiation transport calculations for realistic systems involve the solution of the discretized S_N equations. A typical 3-D transport model requires the discretization of the S_N equations in ~300,000 spatial meshes, 47 energy groups, and considering an S_8 calculation, a total of 80 directions on the unit sphere. These figures yield approximately 1.13 billion unknowns. In terms of computer memory, this number translates into ~ 9 GBytes of RAM for storage (in single precision) of only the angular fluxes. It is clear that an efficient solution for such a problem is out of the scope of a regular workstation

available with current technology. Therefore, it is necessary to develop new algorithms capable of harnessing the computational capabilities of supercomputers. Based on this philosophy, in the late 1990s G. Sjoden and A. Haghighat have developed a new 3-D parallel radiation transport code: PENTRAN (Parallel Environment Neutral-particle Transport).[10]

However, besides the size and complexity of the problem being solved, other aspects come into play, especially when dealing with criticality eigenvalue problems. Because of the physics of these problems, the convergence rate of the currently used iterative methods is quite poor. For realistic problems, such as whole-core reactor calculations performed in a 3-D geometry, the solution of the S_N equations may become impractical if proper acceleration methods[17] are not employed.

The main philosophy behind the novel acceleration algorithms developed in this work is to employ a simplified mathematical model which closely approximates the S_N equations, yet can be solved efficiently.

The new acceleration/preconditioning algorithms have been developed during the course of this research in three major phases:

1. Development of the PENSSn code based on the Even-Parity Simplified S_N (EP-SS_N) method.
2. Investigation of a new synthetic acceleration algorithm based on the EP-SS_N method.
3. Development of an automated acceleration system for the S_N method on parallel environments: FAST© (Flux Acceleration Simplified Transport).

1.5 The Even-Parity Simplified S_N Method

The Even-Parity Simplified S_N (EP-SS_N) method is developed based on the philosophy considered in the P_N and Simplified Spherical Harmonics (SP_N) methods.[18]

The spherical harmonics (P_N) approximation to the transport equation is obtained by expanding the angular flux using spherical harmonics functions truncated to order *N*, where *N* is an odd number; these functions form a complete basis in the limit of the truncation error. In the limit of $N \to \infty$, the exact transport solution is recovered.[1] In 3-D geometries, the number of P_N equations grows as $(N+1)^2$. The P_N equations can be reformulated in a second-order form, cast as $(N+1)^2/2$ diffusion-like equations, characterized by an elliptic operator. However, the number of these equations is very large, and the coupling involves not only angular moments, but also mixed spatial derivatives of these moments.[6]

Because of these issues, to reduce the computing time in the early 1960s, Gelbard et al. proposed the Simplified Spherical Harmonics[18] or SP_N method. The Gelbard procedure consists of extending the spatial variable to 3-D by substituting the second order derivatives in the 1-D P_N equations with the 3-D Laplacian operator. As a result, coupling of spatial derivatives is eliminated, yielding only (*N*+1) equations as compared to $(N+1)^2$. Further, since the SP_N equations can be reformulated as second-order elliptic equations, effective iterative techniques such as Krylov subspace[19-20] methods, can be employed.

The main disadvantage of the SP_N equations is that the exact solution to the transport equation is not recovered as $N \to \infty$, due to terms that are inherently omitted in replacing a 1-D leakage term with a simplified 3-D formulation. However, for idealized systems characterized by homogeneous materials and isotropic scattering, the SP_N and the S_N equations yield the same solution, given the same quadrature set and spatial discretization is used for both methods.

Despite this fact, the SP_N equations yield improved solutions[21-22] compared to the currently used diffusion equation. The theoretical basis for the SP_N equations has been provided by many authors using variational methods and asymptotic analysis.[22] It has been shown that these equations are higher-order asymptotic solutions of the transport equation. Moreover, Pomraning has demonstrated that the SP_N equations, for odd *N*, are a variational approximation to the one-group even-parity transport equation with isotropic scattering in an infinite homogeneous medium.

Recently, the SP_N formulation has received renewed interest, especially in reactor physics applications. For applications such as the MOX fuel assemblies[22-23] or for reactor problems with strong transport effects,[24] diffusion theory does not provide accurate results, while the SP_N equations improve the accuracy of the solution within a reasonable amount of computation time.

Initially, I derived the SP_3 equations starting from the 1-D P_3 equations;[25] however, for developing a general order algorithm, I derived a general formulation using the even-parity form of the S_N transport equation.[26] The Even-Parity Simplified S_N (EP-SS_N) formulation has some interesting properties that make it suitable to develop algorithms of any arbitrary order. Chapter 4 is dedicated to this issue. To make the method more effective, the convergence properties of the EP-SS_N equations were improved by modifying the scattering term; it will be shown that this improved derivation is problem dependent but can reduce the number of iterations significantly.

I have developed a general 3-D parallel code,[23] PENSSn (Parallel Environment Neutral-particle Simplified S_N), based on the EP-SS_N equations. The EP-SS_N equations are discretized with a finite-volume approach, and the spatial domain is partitioned into

coarse meshes with variable fine grid density in each coarse mesh emulating the PENTRAN grid structure. A projection algorithm is used to couple the coarse meshes based on the values of the even-parity angular fluxes on the interfaces. PENSSn uses iterative solvers based on Krylov subspace[19] methods: the Conjugate Gradient (CG) and the Bi-Conjugate Gradient (Bi-CG) solvers. In addition, I have developed a preconditioner based on the Incomplete Cholesky factorization[20] for the CG method. The finite-volume discretization of the EP-SS_N equations in a 3-D Cartesian geometry yields sparse matrices with a 7-diagonal sparse structure. Therefore, I optimized the memory management of PENSSn by using a Compressed Diagonal Storage (CDS) approach, where only the non-zero elements on the diagonals are stored in memory.

The PENSSn code is designed for parallel computing environments with angular, spatial and hybrid (angular/spatial) domain decomposition algorithms.[23] The space decomposition algorithm partitions the 3-D Cartesian space into coarse meshes which are then distributed among the processors while the angular decomposition algorithm partitions the directions among the processors. The hybrid decomposition algorithm is a combination of the two algorithms discussed above. Note that the hybrid decomposition combines the benefits of memory partitioning offered by the spatial decomposition algorithm, and the speed-up offered by the angular decomposition algorithm. The code is written in Fortran 90, and for seamless parallelization, the MPI (Message Passing Interface) library[27] is used.

I have tested the PENSSn code for problems characterized by strong transport effects, and have shown that the improvements over the diffusion equation can be significant.[26] The solutions obtained with the EP-SS_N method are in good agreement with

S_N and Monte Carlo methods; however, the computation time is significantly lower. Hence, these results indicate that the EP-SS_N method is an ideal candidate for the development of an effective acceleration or preconditioning algorithm for radiation transport calculations.

1.6 A New Synthetic Acceleration Algorithm Based on the EP-SS_N Method

As mentioned earlier in this chapter, the solution of the linear Boltzmann equation is obtained numerically via an iterative process. The most widely used technique to iteratively solve the transport equation is the Source Iteration (SI) method[3] or Richardson iteration. The convergence properties of this method are related to the spectral radius of the transport operator. It can be shown that for an infinite, homogeneous medium, the spectral radius of the transport operator is dominated by the scattering ratio *c*, given by

$$c = \frac{\sigma_s}{\sigma_t}, \tag{1.2}$$

where σ_s is the macroscopic scattering cross-section and is the macroscopic total cross-section. Note that Eq. 1.2 can be obtained by Fourier analysis in an infinite homogeneous medium. The asymptotic convergence rate v_∞ is defined as

$$v_\infty \to -\log(c). \tag{1.3}$$

Eq. 1.3 indicates that for problems with scattering ratios close to unity, the unaccelerated SI method is ineffective, because the asymptotic convergence rate tends toward zero. Hence, the use of an acceleration scheme is necessary.

In the past, many acceleration techniques have been proposed[25] for solving the steady-state transport equation. The synthetic methods have emerged as effective techniques to speed-up the convergence of the SI iterative process.[28] In the synthetic

acceleration process a lower-order approximation of the transport equation (e.g., diffusion theory) is corrected using the transport equation at each iteration. In this way the spectral radius of the accelerated transport operator is reduced with consequent speed-up of the iteration process.

Two categories of synthetic methods have been investigated so far, the Diffusion Synthetic Acceleration (DSA) and the Transport Synthetic Acceleration (TSA).[29-30] The first method has been proven to be very effective for 1-D problems and for certain classes of multi-dimensional problems. However, recently it has been shown that for multi-dimensional problems with significant material heterogeneities, the DSA method fails to converge efficiently.[31-32] The same behavior, along with possible divergence, has been reported also for TSA.[30]

I have developed and tested a new synthetic acceleration algorithm[33] based on the simplified form of the Even-Parity transport equations (EP-SS_N). I tested the EP-SS_N algorithm for simple 3-D problems and I concluded that it is affected by instability problems. These instabilities are similar in nature to what has been reported for DSA by Warsa, Wareing, and Morel.[31-32] The main problems affecting the stability of the synthetic methods are material heterogeneities and the inconsistent discretization of the lower-order operator with the transport operator, which leads to divergence if the mesh size is greater than ~1 mean free path.

Moreover, because the synthetic acceleration method has been implemented into the PENTRAN code, another consideration comes into play. The spatial differencing in the PENTRAN code system is based on an Adaptive Differencing Strategy[10] (ADS); with this method the code automatically selects the most appropriate differencing scheme

based on the physics of the problem. Hence, the discretization of the lower-order operator should be consistent with every differencing scheme present in the code. This task is feasible if we consider only the linear-diamond (LD) differencing scheme; however, the complexity increases if we consider the Directional Theta Weighted (DTW) or the Exponential Directional Weighted (EDW) differencing schemes.[34] Moreover, it has been shown that even a fully consistent discretization of the lower-order operator does not guarantee the convergence of the synthetic method.

Due to the issues discussed above, I have developed a system that utilizes the EP-SS_N method as preconditioner for the S_N method.

1.7 An Automatic Preconditioning Algorithm for the S_N Method: FAST© (Flux Acceleration Simplified Transport)

The last phase of the development of an effective acceleration algorithm for the S_N method has culminated in the development of the FAST© system (Flux Acceleration Simplified Transport). The FAST© algorithm is based on the kernel of the PENSSn code. The main philosophy followed in the development of the system is completely antithetic to the synthetic acceleration approach. The idea is to quickly obtain a relatively accurate solution with the EP-SS_N method and to use it a preconditioned initial guess for the S_N transport calculation. This approach has the main advantage of decoupling the two methods, hence avoiding all the stability issues discussed above.

The FAST© system is currently implemented into the PENTRAN-SSn Code System for distributed memory environments, and it is completely automated from a user point of view. Currently, the system has successfully accelerated large 3-D criticality eigenvalue problems, speeding-up the calculations by a factor of 3 to 5 times, and hence reducing significantly the spectral radius.

1.8 Outline

The remainder of this dissertation is organized as follows. Chapter 2 provides the theory for the discrete ordinates method. The discretization of the phase space variables in the transport equation will be discussed, along with the proper boundary conditions. Chapter 3 discusses the theoretical development of the advanced and biased quadrature sets for the discrete ordinates method. It also presents the application of the new quadrature sets for the simulation of a CT-Scan device and for the Kobayashi benchmark problem 3. Chapter 4 discusses the derivation of the EP-SS_N equations. Chapter 5 discusses the numerical methods for the solution of the EP-SS_N equations, along with the iterative solvers based on Krylov subspace methods. Chapter 6 addresses the numerics and accuracy of the EP-SS_N equations. Chapter 7 presents the parallel algorithms implemented in the PENSSn code for distributed memory architectures. Chapter 8 describes the development of a new synthetic acceleration algorithm based on the EP-SS_N method and its limitations. Chapter 9 focuses on the development of the FAST© preconditioner; the performance of the algorithm is measured with two test problems and a large 3-D whole-core criticality eigenvalue calculation. Chapter 10 will draw the conclusions on the objectives accomplished and it will point out some aspects for future development.

CHAPTER 2
THE DISCRETE ORDINATES METHOD

In this chapter, the Discrete Ordinates Method (S_N) will be discussed in detail. The discretized form of the transport equation is formulated in a 3-D Cartesian geometry. I also address the iterative techniques and acceleration methods used to solve the S_N transport equations.

2.1 Discrete Ordinates Method (S_N)

The Discrete Ordinates Method (S_N) is widely used to obtain numerical solutions of the linear Boltzmann equation. In the S_N method, all of the independent variables (space, energy and angle) are discretized as discussed below.

2.1.1 Discretization of the Energy Variable

The energy variable of the transport equation is discretized using the multigroup approach.[3] The energy domain is partitioned into a number of discrete intervals (g=1…G), starting with the highest energy particles (g=1), and ending with the lowest (g=G). The particles in energy group g are those with energies between E_{g-1} and E_g. The multigroup cross-sections for a generic reaction process x are defined as

$$\sigma_{x,g}(\vec{r}) = \frac{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \sigma_x(\vec{r},E)\psi(\vec{r},E,\hat{\Omega})}{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \psi(\vec{r},E,\hat{\Omega})}. \tag{2.1}$$

Based on the definition given in Eq. 2.1, the group constants are defined in Eqs. 2.2, 2.3, and 2.4, for the “total,” “fission” and “scattering” processes, respectively.

$$\sigma_{t,g}(\vec{r}) = \frac{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \sigma_t(\vec{r},E)\psi(\vec{r},E,\hat{\Omega})}{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \psi(\vec{r},E,\hat{\Omega})}, \tag{2.2}$$

$$\sigma_{f,g}(\vec{r}) = \frac{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \sigma_f(\vec{r},E)\psi(\vec{r},E,\hat{\Omega})}{\int_{E_g}^{E_{g-1}} dE \int_{4\pi} d\Omega \psi(\vec{r},E,\hat{\Omega})}, \tag{2.3}$$

$$\sigma_{s,gg'}(\vec{r}) = \frac{\int_{E_{g'}}^{E_{g'-1}} dE' \int_{4\pi} d\Omega' \sigma_s(\vec{r},E' \to E, \hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},E,\hat{\Omega}')}{\int_{E_{g'}}^{E_{g'-1}} dE' \int_{4\pi} d\hat{\Omega} \psi(\vec{r},E,\hat{\Omega})}. \tag{2.4}$$

With the group constants defined above, the multigroup formulation of the transport equation is written as

$$\begin{aligned}\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega}) + \sigma_g(\vec{r})\psi_g(\vec{r},\hat{\Omega}) &= \sum_{g'=1}^{G} \int_{4\pi} d\Omega' \sigma_{gg'}(\vec{r},\hat{\Omega}'\cdot\hat{\Omega})\psi_{g'}(\vec{r},\hat{\Omega}') \\ &+ \frac{1}{k}\chi_g \sum_{g'=1}^{G} \nu\sigma_{f,g'}(\vec{r})\phi_{g'}(\vec{r}) + q_g^e(\vec{r},\hat{\Omega}),\end{aligned} \tag{2.5}$$

for g=1, G,

where the angular flux in group g is defined as

$$\psi_g(\vec{r},\hat{\Omega}) = \int_{E_g}^{E_{g-1}} dE \psi(\vec{r},E,\hat{\Omega}). \tag{2.6}$$

In Eq. 2.5, $q_g^e(\vec{r},\hat{\Omega})$ is the angular dependent fixed source; in general, for criticality eigenvalue problems, this term is set to zero. The scalar flux in Eq. 2.5 is defined as

$$\phi_g(\vec{r}) = \int_{4\pi} d\Omega \psi_g(\vec{r},\hat{\Omega}). \tag{2.7}$$

In a 3-D Cartesian geometry, the "streaming" term can be expressed as

$$\hat{\Omega}\cdot\vec{\nabla} = \mu\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y} + \xi\frac{\partial}{\partial z}, \tag{2.8}$$

where the direction cosines are defined as

$$\mu = \hat{\Omega} \cdot \hat{i}\,, \qquad \eta = \hat{\Omega} \cdot \hat{j}\,, \qquad \xi = \hat{\Omega} \cdot \hat{k}\,. \tag{2.9}$$

Figure 2-1 shows the Cartesian space-angle system of coordinates in three dimensions.

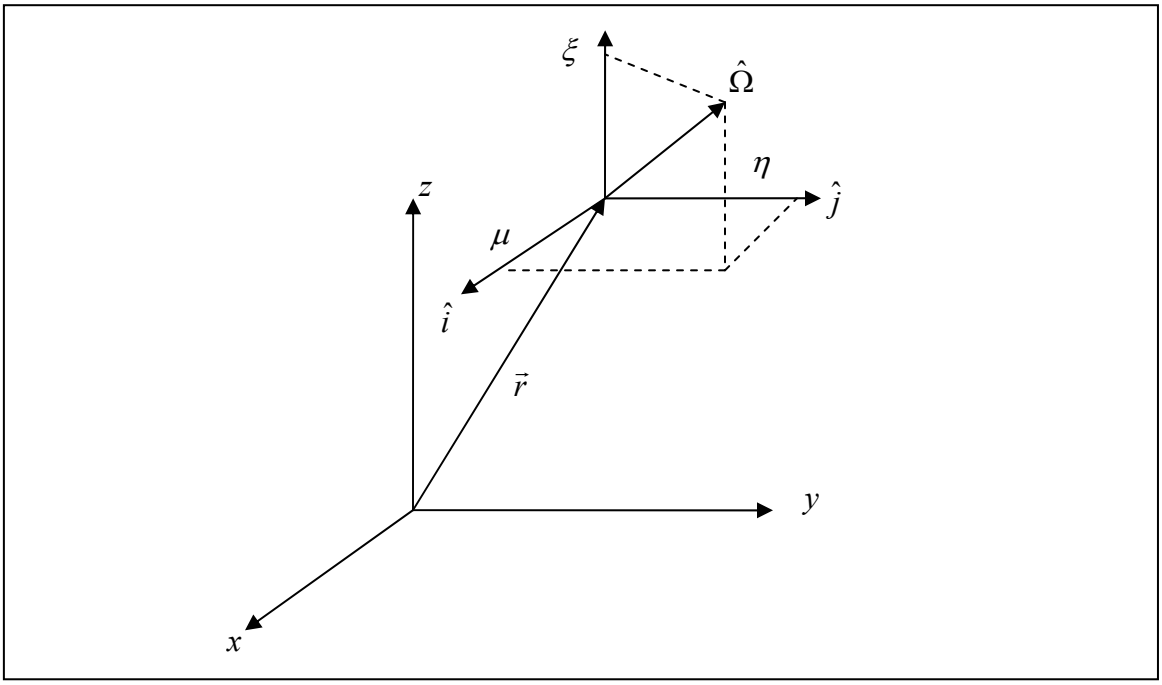

Figure 2-1. Cartesian space-angle coordinates system in 3-D geometry.

The multigroup transport equation, with the scattering kernel expanded in terms of Legendre polynomials and the angular flux in terms of spherical harmonics is given by Eq. 2.10. The complete derivation of the scattering kernel expansion in spherical harmonics shown in Eq. 2.10 is given in Appendix A.

$$\left[\left(\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\xi\frac{\partial}{\partial z}\right)+\sigma_g(x,y,z)\right]\psi_g(x,y,z,\mu,\varphi)=$$
$$\sum_{g'=1}^{G}\sum_{l=0}^{L}(2l+1)\sigma_{sl,g'\to g}(x,y,z)\{P_l(\mu)\phi_{l,g'}(x,y,z)+$$
$$2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)[\phi_{Cl,g'}^k(x,y,z)\cos(k\varphi)+\phi_{Sl,g'}^k(x,y,z)\sin(k\varphi)]\}$$
$$+\frac{\chi_g}{k}\sum_{g'=1}^{G}\nu\sigma_{f,g'}(x,y,z)\phi_{0,g'}(x,y,z)+q_g^e(\vec{r},\hat{\Omega}), \tag{2.10}$$

where

μ : direction cosine along the *x*-axis
η : direction cosine along the *y*-axis
ξ : direction cosine along the *z*-axis
σ_g : total macroscopic cross-section
φ : azimuthal angle, i.e. $\arctan\left(\frac{\xi}{\mu}\right)$
$\psi_g(x,y,z,\mu,\varphi)$: angular flux in energy group *g*
$\sigma_{sl,g'\to g}$: l^{th} moment of the macroscopic transfer cross-section
$P_l(\mu)$: l^{th} Legendre Polynomial
$\phi_{l,g}(\mu)$: l^{th} flux moment
$P_l^k(\mu)$: associated l^{th}, k^{th} Legendre Polynomial
$\phi_{Cl,g}^k(\mu)$: cosine associated l^{th}, k^{th} flux moment
$\phi_{Sl,g}^k(\mu)$: sine associated l^{th}, k^{th} flux moment
χ_g : group fission spectrum
k : criticality eigenvalue
$\nu\sigma_{f,g}$: fission neutron generation cross-section

2.1.2 Discretization of the Angular Variable

The angular variable, $\hat{\Omega}$, in the transport equation is discretized by considering a finite number of directions, and the angular flux is evaluated only along these directions. Each discrete direction can be visualized as a point on the surface of a unit sphere with an associated surface area which mathematically corresponds to the weight of the integration scheme. The combination of the discrete directions and the corresponding weights is

referred to as quadrature set. In general, quadrature sets should satisfy the following properties:[3]

- The associated weights must be positive and normalized to a constant, usually chosen to be one

$$\sum_{m=1}^{M} w_m = 1.0 \,. \tag{2.11}$$

- The quadrature set is usually chosen to be symmetric over the unit sphere, so the solution will be invariant with respect to a 90-degree axis rotation and reflection. This condition results in the odd-moments of the direction cosines having the following property

$$\sum_{m=1}^{M} w_m \mu_m^n = \sum_{m=1}^{M} w_m \eta_m^n = \sum_{m=1}^{M} w_m \xi_m^n = 0.0 \,, \text{ for } n \text{ odd.} \tag{2.12}$$

- The quadrature set must lead to accurate values for moments of the angular flux (i.e., scalar flux, currents); this requirement is satisfied by the following conditions on the even-moments of the direction cosines as follows

$$\sum_{m=1}^{M} w_m \mu_m^n = \sum_{m=1}^{M} w_m \eta_m^n = \sum_{m=1}^{M} w_m \xi_m^n = \frac{1}{n+1} \,, \text{ for } n \text{ even.} \tag{2.13}$$

A widely used method for generating a quadrature set is the level-symmetric technique (LQ_N). In this technique, the directions are ordered on each octant of the unit sphere along the *z*-axis (ξ) on *N*/2 distinct levels. The number of directions on each level is equal to $\frac{N}{2} - i + 1$, for *i*=1, *N*/2. It is worth noting that in 3-D geometries, the total number of directions is $M=N(N+2)$, where *N* is the order of the S_N method.

Considering $\mu_i^2 + \eta_j^2 + \xi_k^2 = 1$ and $i + j + k = \frac{N}{2} + 2$, where *N* refers to the number of levels and *i, j, k* are the indices of the direction cosines, we derive a formulation for determining the directions as follows

$$\mu_i^2 = \mu_1^2 + (i-1)\Delta \,, \tag{2.14}$$

where

$$\Delta = \frac{2(1-3\mu_1^2)}{(N-2)}, \text{ for } 2 \le i \le \frac{N}{2}, \text{ and } 0 < \mu_1^2 \le \frac{1}{3}. \tag{2.15}$$

In Eq. 2.14 the choice of μ_1 determines the distribution of directions on the octant. If the value of μ_1 is small, the ordinates will be clustered near the poles of the sphere; alternatively, if the value of μ_1 is large, the ordinates will be placed far from the poles.

The weight associated to each direction, called a point weight, is then evaluated with another set of equations. For example, in the case of an S_8 level-symmetric quadrature set, this condition can be formulated as follows

$$2p_1 + 2p_2 = w_1, \tag{2.16a}$$

$$2p_2 + p_3 = w_2, \tag{2.16b}$$

$$2p_2 = w_3, \tag{2.16c}$$

$$1p_1 = w_4, \tag{2.16d}$$

where p_1, p_2 and p_3 are point weights associated with each direction, and w_1, w_2, w_3, w_4 are the weights associated with the levels, as shown in Figure 2-2.

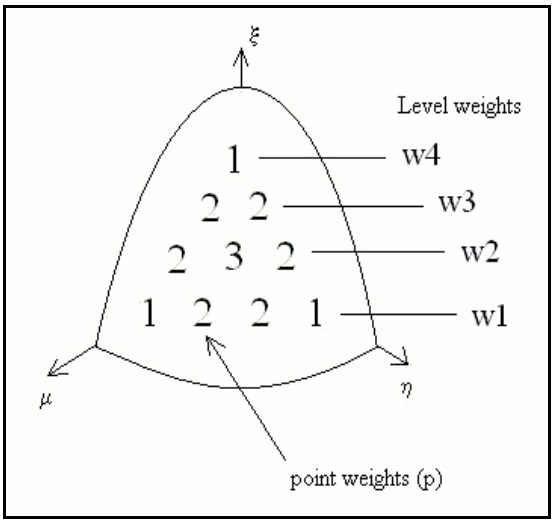

Figure 2-2. Point weight arrangement for a S_8 level-symmetric quadrature set.

As an example, Figure 2-3 shows the S_{20} LQ_N quadrature set for one octant of the unit sphere.

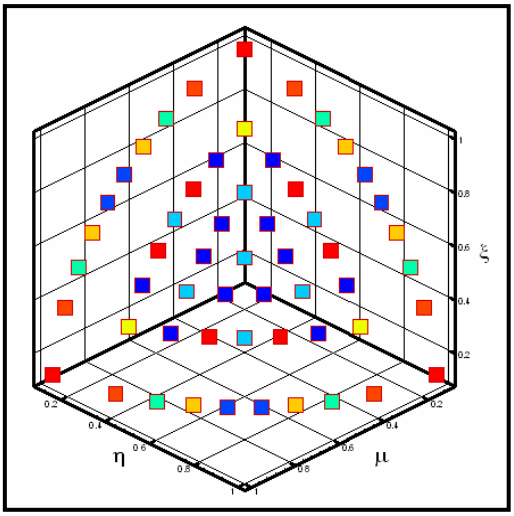

Figure 2-3. S_{20} LQ_N quadrature set.

Note that, this quadrature set is limited by unphysical negative weights beyond order S_{20}. Therefore, if a higher order quadrature set is needed beyond S_{20}, another formulation has to be developed, which satisfies the even- and odd-moments conditions. To address this issue, I have developed new quadrature techniques based on the Gauss-Legendre quadrature formula and on the Chebyshev polynomials.

2.1.3 Discretization of the Spatial Variable

The linear Boltzmann equation, given in Eq. 2.10, can be rewritten in an abbreviated form as

$$\left[\left(\mu_m \frac{\partial}{\partial x} + \eta_m \frac{\partial}{\partial y} + \xi_m \frac{\partial}{\partial z}\right) + \sigma_g(\vec{r})\right]\psi_{m,g}(\vec{r}) = Q_{m,g}(\vec{r}), \qquad (2.17)$$

$$\text{for } m = 1, M \text{ and } g = 1, G.$$

The angle and energy dependence are denoted by the indices *m* and *g*, respectively. The right hand side of Eq. 2.17 represents the sum of the scattering, fission and external sources.

The spatial domain is partitioned into computational cells, bounded by $x_{1/2}, x_{3/2}, \ldots, x_{I+1/2}; y_{1/2}, y_{3/2}, \ldots, y_{J+1/2}; z_{1/2}, z_{3/2}, \ldots, z_{K+1/2}$. The cross-sections are assumed to be constant within each cell and they are denoted by $\sigma_{i,j,k}$. Eq. 2.17 is then integrated over the cell volume $V_{i,j,k} = \Delta x_i \Delta y_j \Delta z_k$, and then divided by the cell volume to obtain the volume- and surface-averaged fluxes, therefore reducing to

$$\begin{aligned} &\frac{\mu_m}{\Delta x_i}\left(\Psi_{i+1/2,j,k,m,g} - \Psi_{i-1/2,j,k,m,g}\right) + \frac{\eta_m}{\Delta y_j}\left(\Psi_{i,j+1/2,k,m,g} - \Psi_{i,j-1/2,k,m,g}\right) \\ &+ \frac{\xi_m}{\Delta z_k}\left(\Psi_{i,j,k+1/2,m,g} - \Psi_{i,j,k-1/2,m,g}\right) + \sigma_{i,j,k}\Psi_{i,j,k,m,g} = Q_{i,j,k,m,g}. \end{aligned} \tag{2.18}$$

In Eq. 2.18, the indices *i, j, k* represent the cell-center values, while $i\pm1/2$, $j\pm1/2$, $k\pm1/2$ refer to the surface values.

2.1.4 Differencing Schemes

For the S_N method, different classes of differencing schemes are available. *Low-order differencing schemes* require only the angular fluxes, and the average values at the cell boundaries to be related at the cell average value. Various forms of Weighted Difference (WD) schemes belong to this class. *High-order differencing schemes* require higher order moments, and may be linear or non-linear. Discontinuous, characteristic, and exponential schemes are examples of high-order differencing schemes.[35]

The solution of the S_N equations is obtained by marching along the discrete directions generated in each octant of the unit sphere; this process is usually referred to as a *transport sweep*.[3] For each computational cell, the angular fluxes on the three incoming surfaces are already known, from a previous calculation or boundary conditions. The cell-center fluxes and the fluxes on the three outgoing surfaces must be calculated, hence

additional relationships are needed. The additional relationships are referred to as the "*differencing schemes"*. The general form of WD schemes can be expressed as

$$\Psi_{i,j,k,m,g} = a_{i,j,k,m,g}\Psi_{i+1/2,j,k,m,g} + (1 - a_{i,j,k,m,g})\Psi_{i-1/2,j,k,m,g}, \tag{2.19a}$$

$$\Psi_{i,j,k,m,g} = b_{i,j,k,m,g}\Psi_{i,j+1/2,k,m,g} + (1 - b_{i,j,k,m,g})\Psi_{i,j-1/2,k,m,g}, \tag{2.19b}$$

$$\Psi_{i,j,k,m,g} = c_{i,j,k,m,g}\Psi_{i,j,k+1/2,m,g} + (1 - c_{i,j,k,m,g})\Psi_{i,j,k-1/2,m,g}. \tag{2.19c}$$

The values $a_{i,j,k,m,g}$, $b_{i,j,k,m,g}$, and $c_{i,j,k,m,g}$ are determined based on the type of weighted scheme employed.

2.1.4.1 Linear-Diamond Scheme (LD)

In the LD scheme, the cell-average flux is an arithmetic average of any two opposite boundary fluxes; hence the weights are set to constant values $a=b=c=1/2$.

$$\begin{aligned}\Psi_{i,j,k,m,g} &= \frac{1}{2}\left(\Psi_{i+1/2,j,k,m,g} + \Psi_{i-1/2,j,k,m,g}\right) = \\ &= \frac{1}{2}\left(\Psi_{i,j+1/2,k,m,g} + \Psi_{i,j-1/2,k,m,g}\right) = \\ &= \frac{1}{2}\left(\Psi_{i,j,k+1/2,m,g} + \Psi_{i,j,k-1/2,m,g}\right)\end{aligned} \tag{2.20}$$

For example, in the direction $\mu_m > 0, \eta_m > 0, \xi_m > 0$ the outgoing fluxes are obtained as follows

$$\Psi_{i+1/2,j,k,m,g} = 2\Psi_{i,j,k,m,g} - \Psi_{i-1/2,j,k,m,g}, \tag{2.21a}$$

$$\Psi_{i,j+1/2,k,m,g} = 2\Psi_{i,j,k,m,g} - \Psi_{i,j-1/2,k,m,g}, \tag{2.21b}$$

$$\Psi_{i,j,k+1/2,m,g} = 2\Psi_{i,j,k,m,g} - \Psi_{i,j,k-1/2,m,g}. \tag{2.21c}$$

We can then eliminate the fluxes on the outgoing surfaces in Eq. 2.18 and obtain the center-cell angular flux

$$\Psi_{i,j,k,m,g} = \frac{\frac{2\mu_m}{\Delta x_i}\Psi_{i-1/2,j,k,m,g} + \frac{2\eta_m}{\Delta y_j}\Psi_{i,j-1/2,k,m,g} + \frac{2\xi_m}{\Delta z_k}\Psi_{i,j,k-1/2,m,g} + Q_{i,j,k}}{\sigma_{i,j,k} + \frac{2\mu_m}{\Delta x_i} + \frac{2\eta_m}{\Delta y_j} + \frac{2\xi_m}{\Delta z_k}}. \quad (2.22)$$

The outgoing fluxes are then evaluated using Eqs. 2.21a, b, and c.

The LD differencing scheme may yield negative angular fluxes in regions where the flux gradient is large, even if the incoming fluxes and scattering source are positive. In this case, one approach is to set negative fluxes equal to zero, and then the cell-average flux is recalculated to preserve the balance of particles. This approach is referred to as *negative flux fix-up* (DZ). The DZ scheme performs better than the LD in practical applications, but the linearity and accuracy of the LD equations is not preserved.

2.1.4.2 Directional Theta-Weighted Scheme (DTW)

This scheme uses a direction-based parameter to obtain the weighting factors *a*, *b*, and *c* which are restricted to the range 0.5 and 1.0. The DTW scheme uses a direction-based parameter to obtain an angular flux weighting factor, which ensures positivity of the angular flux and removes the oscillations due to the spatial and angular discretization.[36] The DTW average cell angular flux is given by

$$\psi_{i,j,k,m,g} = \frac{q_{i,j,k} + \frac{|\mu_m|}{a_{i,j,k,m,g}\Delta x}\psi_{in,x} + \frac{|\eta_m|}{b_{i,j,k,m,g}\Delta y}\psi_{in,y} + \frac{|\xi_m|}{c_{i,j,k,m,g}\Delta z}\psi_{in,z}}{\frac{|\mu_m|}{a_{i,j,k,m,g}\Delta x} + \frac{|\eta_m|}{b_{i,j,k,m,g}\Delta y} + \frac{|\xi_m|}{c_{i,j,k,m,g}\Delta z} + \sigma}. \quad (2.23)$$

In the DTW scheme, the weights ($a_{i,j,k,m,g}$, $b_{i,j,k,m,g}$, $c_{i,j,k,m,g}$) are restricted to the range between 0.5 and 1.0, approaching second order accuracy when all weights are equal to 0.5, which is in this case is equivalent to the LD scheme.

2.1.4.3 Exponential Directional-Weighted Scheme (EDW)

The Exponential Directional Weighted (EDW) differencing scheme[34], implemented in PENTRAN, is a predictor-corrector scheme, which utilizes the DTW to predict a solution that is then corrected using an exponential fit. The EDW is an inherently positive scheme, and the auxiliary equations derived for this method are given in Eq. 2.24.

$$\psi_m(x,y,z) = a\exp\left(\lambda_i P_1(x)/|\mu_m|\right)\exp\left(\lambda_j P_1(y)/|\eta_m|\right)\exp\left(\lambda_k P_1(z)/|\xi_m|\right). \tag{2.24}$$

The DTW scheme is used to calculate the angular fluxes ($\tilde{\psi}$) needed for the estimation of the coefficients λ_i, λ_j, and λ_k given in Eqs. 2.25a, b, and c, respectively.

$$\lambda_i \approx \frac{\left(\tilde{\psi}_{out,x} - \tilde{\psi}_{in,x}\right)|\mu_m|}{2\tilde{\psi}_A}, \tag{2.25a}$$

$$\lambda_j \approx \frac{\left(\tilde{\psi}_{out,y} - \tilde{\psi}_{in,y}\right)|\eta_m|}{2\tilde{\psi}_A}, \tag{2.25b}$$

$$\lambda_k \approx \frac{\left(\tilde{\psi}_{out,z} - \tilde{\psi}_{in,z}\right)|\xi_m|}{2\tilde{\psi}_A}, \tag{2.25c}$$

where the subscripts *in* and *out* refers to the incoming and outgoing surface averaged angular fluxes. The cell-average angular flux formulated with the EDW scheme is given in Eq. 2.26.

$$\begin{aligned}\psi_A = &\left(\exp\left(\frac{2\lambda_i}{|\mu_m|}\right)-1\right)\left(\exp\left(\frac{2\lambda_j}{|\eta_m|}\right)-1\right)\left(\exp\left(\frac{2\lambda_k}{|\xi_m|}\right)-1\right)\cdot \\ &\cdot\frac{1}{\beta}\left(q_A + \frac{|\mu_m|}{\Delta x}\psi_{in,x} + \frac{|\eta_m|}{\Delta y}\psi_{in,y} + \frac{|\xi_m|}{\Delta z}\psi_{in,z}\right),\end{aligned} \tag{2.26}$$

where β is calculated using the coefficients given in Eqs. 2.25a, b, and c.[37]

2.1.5 The Flux Moments

The flux moments are obtained from the angular fluxes using the following formulation

$$\phi_{i,j,k,l} = \sum_{m=1}^{M} w_m \Psi_{i,j,k} P_l(\mu_m). \tag{2.27}$$

Note that Eq. 2.27 for l=0 yields the scalar flux. The Associated Legendre moments are calculated using Eqs. 2.28 and 2.29.

$$\phi^n_{i,j,k,C,l} = \sum_{m=1}^{M} w_m \Psi_{i,j,k} P^n_l(\mu_m)\cos(n\mu_m), \tag{2.28}$$

$$\phi^n_{i,j,k,S,l} = \sum_{m=1}^{M} w_m \Psi_{i,j,k} P^n_l(\mu_m)\sin(n\mu_m). \tag{2.29}$$

2.1.6 Boundary Conditions

Three major boundary conditions can be expressed with a general formula as

$$\Psi\left(\vec{r}_s, \hat{\Omega}_m\right) = a\Psi\left(\vec{r}_s, \hat{\Omega}'_m\right), \tag{2.30}$$

where $\hat{\Omega}'_m \cdot \hat{n} = -\hat{\Omega}_m \cdot \hat{n}$.

Depending on the value of coefficient a, the three boundary conditions are:

a=1, reflective boundary condition.

a=0, vacuum or non-reentrant boundary condition.

a=β, albedo boundary condition.

2.2 Source Iteration Method

Due to the integro-differential nature of the transport equation, the solution of the multigroup discrete ordinates equations is obtained by means of an iterative process, named the *source iteration*.[3] The source iteration method consists of guessing a source (i.e., in-group scattering source), then sweeping through the angular, spatial and energy

domains of the discretized system with the appropriate boundary conditions. When the sweep is completed, integral quantities such as scalar flux and flux moments are obtained from the angular fluxes, and then a new in-group scattering source is calculated, and the iteration process continues until a convergence criterion, shown in Eq. 2.31, is satisfied. Typical tolerances for fixed source calculations range in the order of 1.0e-3 to 1.0e-4.

$$\frac{\left|\psi^{i}-\psi^{i-1}\right|}{\psi^{i-1}}<\varepsilon. \tag{2.31}$$

If fission and/or up-scattering processes are present, outer iterations are performed on the fission and transfer scattering sources. Acceleration techniques may be applied between source iterations to speed-up the convergence rate by determining a better guess for the flux moments and the source.

2.3 Power Iteration Method

Criticality eigenvalue problems are solved using the method of power iteration.[38] For this method, it is assumed that the eigenvalue problem has a largest positive eigenvalue, $k>0$, with an associated fission distribution $F(\vec{r})$ that is nonnegative. Hence by considering $k^0>0$ and $F^0(\vec{r})>0$ as initial guesses, we calculate the eigenvalue at iteration i as follows

$$k^{i+1}=k^{i}\frac{\int dE\int d\vec{r}F^{i+1}(\vec{r})}{\int dE\int d\vec{r}F^{i}(\vec{r})}, \tag{2.32}$$

where $F^{i}(\vec{r})=\nu\sigma_{f}(\vec{r},E)\phi^{i}(\vec{r},E)$ is the fission source distribution at iteration i. The iterative process is continued until the desired convergence is reached, as shown in Eq. 2.33.

$$\frac{\left|k^{i}-k^{i-1}\right|}{k^{i-1}}<\varepsilon_{k}. \tag{2.33}$$

Generally, the tolerance required for criticality calculation is *1.0e-4* to 1.0e-6.

2.4 Acceleration Algorithms for the S_N Method

Many acceleration methods have been proposed to speed-up the convergence of the iterative methods used to solve the steady-state transport equation.[17] There are a number of problems where standard non-accelerated iterative methods converge too slowly to be practical. Most of these problems are characterized by optically thick regions with scattering ratio near unity.

The three major acceleration approaches are the Coarse Mesh Rebalance (CMR), Multigrid, and Synthetic methods. The CMR approach is based on the fact that the converged solution must satisfy the particle balance equation.[3] By imposing this balance condition on the unconverged solution over coarse regions of the problem domain, it is possible to obtain an iteration procedure that usually converges more rapidly to the correct solution. However, this method is highly susceptible to the choice of the coarse mesh structure and can be unstable.

The multigrid approach has been used to accelerate the S_N calculations; the basic principle of the method is to solve the equations on a coarse grid and to project the solution onto a finer grid. Different types of multigrid approaches exist, such as "/" Slash-cycle, V-Cycle and W-Cycle, and the Simplified Angular Multigrid[39] (SAM). In Chapter 8, I will compare the results obtained with the EP-SS_N synthetic acceleration and the SAM method.

The synthetic acceleration approach is based on using a lower-order operator, generally diffusion theory, as a means to accelerate the numerical solution of the

transport equation.[17] In the late 1960s, Gelbard and Hageman developed a synthetic acceleration method based on the diffusion and the S_4 equations.[28] Later, Reed independently derived a similar synthetic acceleration scheme[40] and pointed out some limitations of the method derived by Gelbard and Hageman. The synthetic method developed by Reed has the advantage of being very effective for small mesh sizes, but it is unstable for mesh sizes greater than ~1 mfp (mean free path). Later, Alcouffe independently derived the Diffusion Synthetic Acceleration (DSA) method.[29] He addressed the issue of stability of the method and derived an unconditionally stable DSA algorithm. Alcouffe pointed out that in order to obtain an unconditionally stable method, the diffusion equation must be derived consistently from the discretized version of the transport equation. In this way, the consistency between the two operators is preserved.

Recently, the DSA method has been found to be ineffective[30-32] for multidimensional problems with strong heterogeneities, even when a consistent discretization of the lower order and transport operators is performed.

CHAPTER 3
ADVANCED QUADRATURE SETS FOR THE S_N METHOD

This chapter covers the development of advanced quadrature sets for solving the neutron transport equation via the Discrete Ordinates (S_N) method. The level-symmetric (LQ_N) quadrature set is the standard quadrature set for S_N calculations; although, as discussed in the previous chapter, this quadrature set is limited to order 20. The Equal Weight (EW) quadrature set has been proposed to resolve the issue of negative weights; this quadrature set is generated by partitioning the unit sphere into *M* directions, where $M = N(N+2)$ and by assigning an equal weight to each direction $w_i = \frac{1}{M}$. The EW quadrature set yields positive weights for any S_N order; however, it does not completely satisfy the even-moment conditions given in Eqs. 2-13.

I have developed and tested new quadrature sets based on the Legendre (P_N) and Chebyshev (T_N) polynomials. In this chapter, I discuss the Legendre Equal-Weight (P_N-EW), the Legendre-Chebyshev (P_N-T_N) quadrature sets, and the Regional Angular Refinement (RAR) technique for local angular refinements. The P_N-EW and P_N-T_N have no limitations on the number of directions, and the RAR technique is an alternative to the Ordinate Splitting (OS) technique.[11] The OS technique has been developed to refine the directions of a standard quadrature set using equal-spaced and equal-weight directions, while the RAR utilizes the P_N-T_N quadrature set over a subset of ordinates. The main difference between the OS and RAR techniques is in the refining methodology; the OS technique focuses on the refinement of each single direction, while the RAR considers a

sector of the unit sphere. Also the biased quadrature set generated with RAR satisfies the conditions on the odd- and even-moments of the direction cosines (Eq. 2-12 and 2-13).

To examine the effectiveness of the new techniques for angular quadrature generation, each technique has been implemented into the PENTRAN code[10] (Parallel Environment Neutral-Particle TRANsport), and utilized for a number of problems of practical interest.

3.1 Legendre Equal-Weight (P_N-EW) Quadrature Set

In order to develop a quadrature set which is not limited to order S_{20}, I have investigated the Gauss-Legendre quadrature technique.[4] This quadrature set is characterized by the same arrangement of directions as the LQ_N, but the directions and weights are evaluated differently. Given the S_N order for the discrete set of directions, we apply the Gauss-Legendre quadrature formula using the following recursive formulation

$$(j+1)P_{j+1} = (2j+1)\xi P_j - jP_{j-1}, \text{ for } j = 0, N, \tag{3.1}$$

where

$$-1 < \xi < 1, \; P_{-1}(\xi) = 0, \text{ and } \; P_0(\xi) = 1. \tag{3.2}$$

The ξ-levels or polar angles, along the z-axis are set equal to the roots of Eq. 3.1. The ξ values represent the *levels* of the quadrature set. Once we have evaluated the ordinates along the z-axis, we obtain the weights associated with each level using the following recursive formulation

$$w_i = \frac{2}{(1-\xi_i^2)\left[\left(\frac{dP_N}{d\xi}\right)_{\xi_i}\right]^2}, \text{ for } i = 1, \frac{N}{2}. \tag{3.3}$$

In order to complete the definition of each discrete direction, the azimuthal angle is evaluated on each level by equally partitioning a 90 degree angle into $\frac{N}{2}-i+2$ angular intervals, where i=1, N/2. Hence, the weight associated with each direction is given by

$$p_{i,j}=\frac{w_i}{j}, \text{ for } i=1\ldots\frac{N}{2}. \tag{3.4}$$

In Eq. 3.4, $j=1\ldots\frac{N}{2}-i+1$ is the number of directions with equal weights on the i^{th} level. Figure 3-1 shows the directions and the associated weights for an S_{28} P_N-EW quadrature set on one octant of the unit sphere.

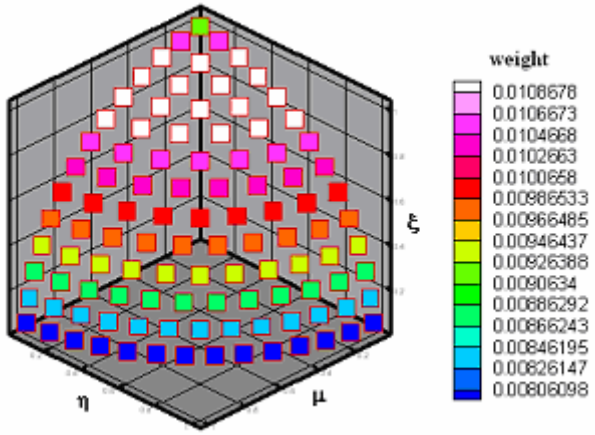

Figure 3-1. S_{28} P_N-EW quadrature set.

Note that in Figure 3-1, all directions on the same ξ-level have the same weight, as indicated by the color.

3.2 Legendre-Chebyshev (P_N-T_N) Quadrature Set

In the P_N-T_N quadrature set, similar to P_N-EW, we choose the ξ-levels on the z-axis equal to the roots of the Gauss-Legendre quadrature formula given in Eqs. 3.1 and 3.2; however, the azimuthal angles on each level are set equal to the roots of the Chebyshev

(T_N) polynomials of the first kind. Chebyshev polynomials of the first kind are formulated as follows

$$T_l\left[\cos(\omega)\right] \equiv \cos(l\omega). \tag{3.5}$$

The Chebyshev polynomials are orthogonal and satisfy the following conditions

$$\int_{-1}^{1} dy T_l(y) T_k(y)(1-y^2)^{-1/2} = \begin{cases} 0, l \neq k \\ \pi, l = k = 0 \\ \pi/2, l = k \neq 0 \end{cases}$$
$$y = \cos(\omega) \tag{3.6}$$

Again, using the ordering of the LQ_N quadrature set, we define the azimuthal angles on each level using the following formulation

$$\omega_{i,j} = \left(\frac{2i-2j+1}{2i}\right)\frac{\pi}{2}, \tag{3.7}$$

where $\omega_{i,j} \in \left(0, \frac{\pi}{2}\right)$ and i=1, N/2.

In Eq. 3.7, i is the level number and $j = 1 \ldots \frac{N}{2} - i + 1$. The level and point weights are generated in the same way as for the P_N-EW. Note that both P_N-EW and P_N-T_N quadrature sets do not present negative weights for S_N orders higher than 20.

Figure 3-2 shows the directions and the associated weights for an S_{28} P_N-T_N quadrature set on one octant of the unit sphere.

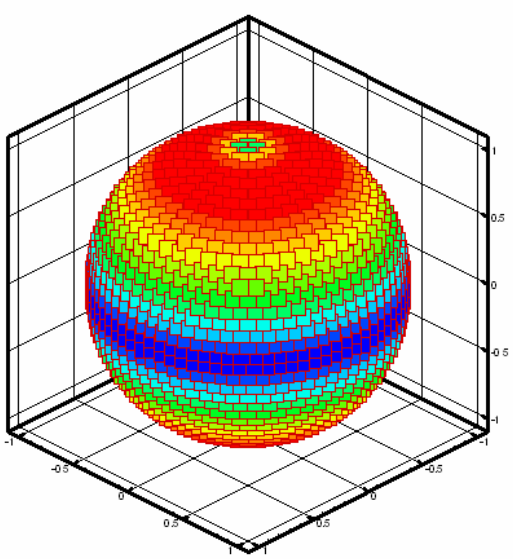

Figure 3-2. S_{40} P_N-T_N quadrature set over the unit sphere.

3.3 The Regional Angular Refinement (RAR) Technique

The RAR method is developed for solving problems with highly peaked angular fluxes and/or sources. The approach consists of two steps. In the first step, we derive a P_N-T_N quadrature set of arbitrary order on one octant of the unit sphere, as described in Section 3.2.

In the second step, we define the area (angular segment) of the octant to be refined along with the order (N') of the $P_{N'}$-$T_{N'}$ quadrature set to be used in this region. The area to be refined is characterized by the polar range (ξ_{min}, ξ_{max}) and the azimuthal range (φ_{min}, φ_{max}). Generally, the biased region is selected based on the physical properties of the model. For example, if a directional source is forward peaked along the x-axis, the quadrature set will be refined on the pole along the x-axis.

The ξ-levels of the $P_{N'}$-$T_{N'}$ quadrature set are calculated using Eqs. 3.1 and 3.2; therefore, they are mapped onto the (ξ_{min}, ξ_{max}) sub-domain using the following formula:

$$\tilde{\xi}_i = \left(\frac{\xi_{max} - \xi_{min}}{2}\right)\xi_i + \left(\frac{\xi_{max} + \xi_{min}}{2}\right), \text{ for } i=1, N'/2. \tag{3.8}$$

Hence, the azimuthal angles are evaluated using the Chebyshev polynomials as follows

$$\omega_{i,j} = \frac{\pi}{2}\left(\frac{N'-2j+1}{N'}\right)\left(\frac{\varphi_{\max}-\varphi_{\min}}{2}\right)+\left(\frac{\varphi_{\max}+\varphi_{\min}}{2}\right), \tag{3.9}$$

where $\omega_{i,j} \in \left(0, \frac{\pi}{2}\right)$, i=1, N'/2 and j=1, N'/2.

The number of directions in the refined region is equal to $\left(\frac{N'}{2}\right)^2$. The weights in the refined region are renormalized to preserve the overall normalization on the unit sphere.

Figure 3-3 shows the RAR technique applied to an S_{16} P_N-T_N quadrature set. In the biased region, which extends from ξ=0.0 to ξ=0.2 along the z-axis, and from φ=0º to φ=10º on the azimuthal plane, an S_{10} P_N-T_N quadrature set is fitted.

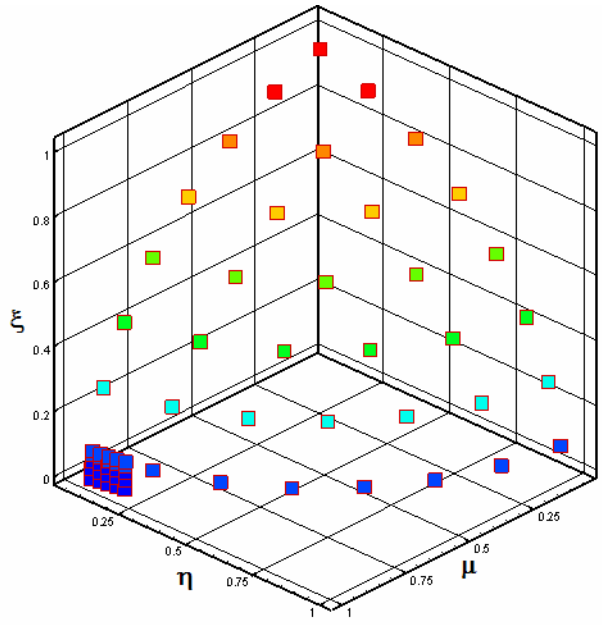

Figure 3-3. P_N-T_N quadrature set (S_{16}) refined with the RAR technique.

3.4 Analysis of the Accuracy of the P_N-EW and P_N-T_N Quadrature Sets

The main advantage of the LQ_N technique is the fact that it preserves moments of the direction cosines, thereby leading to an accurate solution. The P_N-EW and P_N-T_N

quadrature sets attempt to preserve these quantities independently along the μ-, η-, and ξ-axes.

Therefore, I verified the capability of the new quadrature sets in preserving the even moments of the direction cosines, which are directly related to the accuracy of the quadrature set. Table 3-1 compares the even-moments of the direction cosines calculated with an S_{30} P_N-EW quadrature set with the exact value.

Table 3-1. Even-moments obtained with a P_N-EW S_{30} quadrature set.

Moment Order (n even)	$\sum_{i=1}^{M} w_i \mu_i^n$	$\sum_{i=1}^{M} w_i \eta_i^n$	$\sum_{i=1}^{M} w_i \xi_i^n$	Exact value $\left(\frac{1}{1+n}\right)$
2	0.333333333	0.333333333	0.333333333	0.333333333
4	0.194318587	0.194318587	0.2	0.2
6	0.135907964	0.135907964	0.142857143	0.142857143
8	0.103874123	0.103874123	0.111111111	0.111111111
10	0.083691837	0.083691837	0.090909091	0.090909091
12	0.06983611	0.06983611	0.076923077	0.076923077
14	0.059749561	0.059749561	0.066666667	0.066666667
16	0.052087133	0.052087133	0.058823529	0.058823529
18	0.046074419	0.046074419	0.052631579	0.052631579
20	0.041234348	0.041234348	0.047619048	0.047619048
22	0.037257143	0.037257143	0.043478261	0.043478261
24	0.033932989	0.033932989	0.043478261	0.043478261
26	0.031114781	0.031114781	0.037037037	0.037037037
28	0.028696358	0.028696358	0.034482759	0.034482759
30	0.026599209	0.026599209	0.032258065	0.032258065

As expected, the P_N-EW preserves exactly the even-moments conditions along the ξ-axis, while on the μ-, η-axes these conditions are only partially preserved; the maximum relative difference between the even-moments calculated with P_N-EW and the exact solution is 17.0%.

Table 3-2 shows the comparison of the even-moments evaluated with an S_{30} P_N-T_N quadrature set and the exact value.

Table 3-2. Even-moments obtained with a P_N-T_N S_{30} quadrature set.

Moment Order (n even)	$\sum_{i=1}^{M} w_i \mu_i^n$	$\sum_{i=1}^{M} w_i \eta_i^n$	$\sum_{i=1}^{M} w_i \xi_i^n$	Exact value $\left(\frac{1}{1+n}\right)$
2	0.333333333	0.333333333	0.333333333	0.333333333
4	0.199999962	0.199999962	0.2	0.2
6	0.142857143	0.142857143	0.142857143	0.142857143
8	0.111111111	0.111111111	0.111111111	0.111111111
10	0.090909091	0.090909091	0.090909091	0.090909091
12	0.076923077	0.076923077	0.076923077	0.076923077
14	0.066666667	0.066666667	0.066666667	0.066666667
16	0.058823529	0.058823529	0.058823529	0.058823529
18	0.052631579	0.052631579	0.052631579	0.052631579
20	0.047619048	0.047619048	0.047619048	0.047619048
22	0.043478261	0.043478261	0.043478261	0.043478261
24	0.043478261	0.043478261	0.043478261	0.043478261
26	0.037037037	0.037037037	0.037037037	0.037037037
28	0.034482759	0.034482759	0.034482759	0.034482759
30	0.032258065	0.032258065	0.032258065	0.032258065

It is clear from Table 3-2 that the P_N-T_N quadrature set completely satisfies the even-moment conditions. This is possible because both roots of Legendre and Chebyshev polynomials satisfy the even-moment conditions given by Eqs. 2.13.

In order to further verify the accuracy of the P_N-EW and P_N-T_N quadrature sets, and to check their accuracy, I used a simple test problem, consisting of a homogeneous parallelepiped, where an isotropic source is placed in its lower left corner as shown in Figure 3-4.

Due to the symmetry of the problem, it is expected that the particle currents flowing out of regions A and B (see Figure 3-4) have the same value.

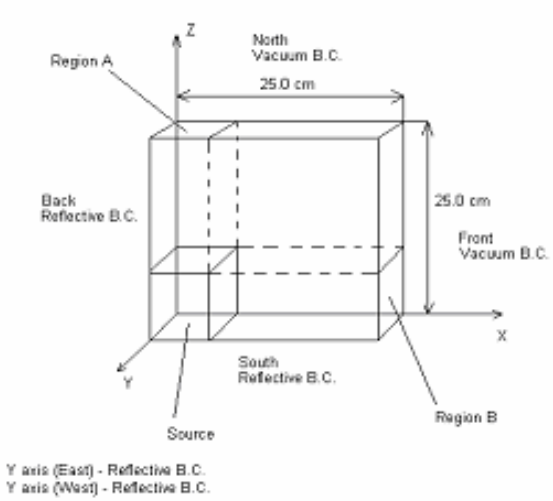

Figure 3-4. Configuration of the test problem for the validation of the quadrature sets.

Figure 3-5 shows the relative difference between the particle current in regions A and B for the EW, P_N-EW, and P_N-T_N as compared to the LQ_N technique.

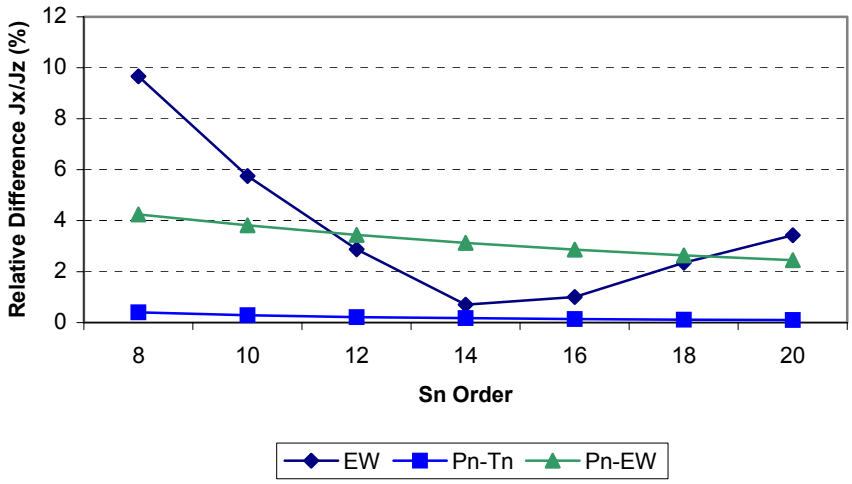

Figure 3-5. Relative difference between the currents J_x and J_z for the test problem.

Level-symmetric is considered as the reference because it preserves moments of both azimuthal and polar direction cosines. The P_N-T_N yields almost perfect symmetry, while the P_N-EW and Equal Weight show maximum relative differences of 4% and 10%, respectively. It is worth noting that the loss in accuracy of the EW quadrature set is attributed to the fact that the even-moment conditions are not satisfied. The P_N-EW yields

higher accuracy compared to EW, because the even-moment conditions are satisfied along the *z*-axis.

3.5 Testing the Effectiveness of the New Quadrature Sets

In this section, the effectiveness of the new quadrature sets is examined by simulating two test problems: Kobayashi benchmark problem 3 and a CT-Scan device for industrial/medical imaging applications.

3.5.1 Kobayashi Benchmark Problem 3

To examine the effectiveness of the new quadrature sets, I have used the first axial slice of the Kobayashi[43] 3-D benchmark problem 3 with pure absorber. Figures 3-6 show two different mesh distributions: Figure 3-6A is obtained from a previous study[42], where an appropriate variable mesh was developed; Figure 3-6B shows a uniform mesh distribution that I have developed for the current study. The uniform mesh is used in order to separate the effects of the angular discretization from the spatial discretization. The reference semi-analytical solutions are evaluated in two spatial zones shown in Figures 3-6A and 3-6B (zone 1 ≡ along *y*-axis, at every 10.0 cm intervals between 5.0 and 95.0 cm; zone 2 ≡ along *x*-axis, *y* = 55.0 cm, every 10.0 cm, between 5.0 cm and 55.0 cm).

A B

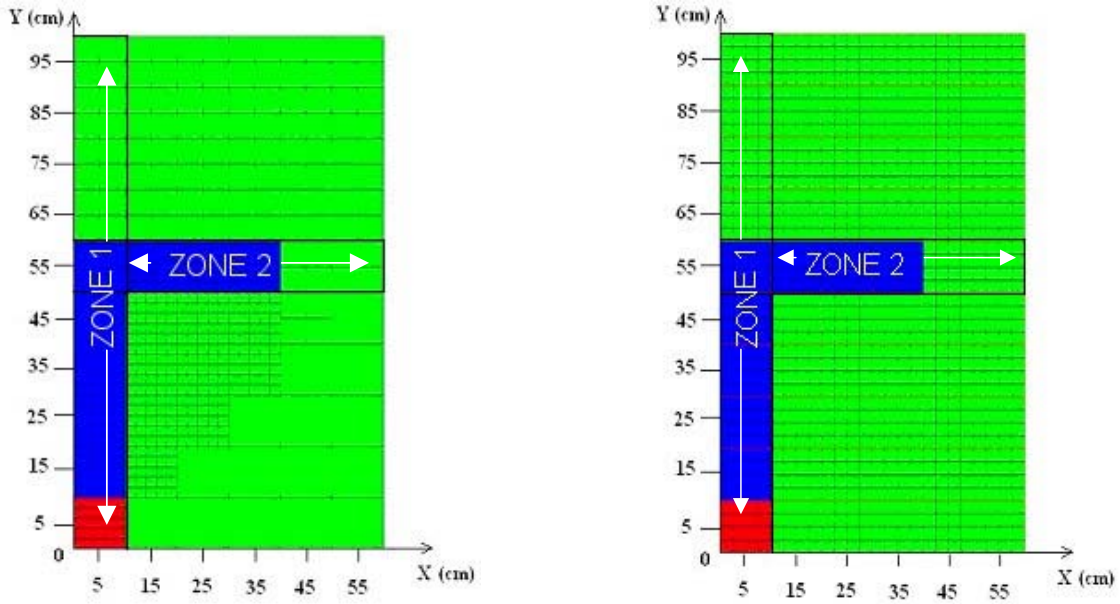

Figure 3.6. Mesh distribution for the Kobayashi benchmark problem 3: A) Variable mesh distribution; B) Uniform mesh distribution.

Figure 3-7 shows the ratio of the calculated to the exact solution (C/E) for the level-symmetric, P_N-EW and P_N-T_N quadrature sets of order 20 for zone 1. Also, in this figure, I present a solution[34] obtained in a previous study which uses the variable mesh distribution shown in Figure 3-6a. In the previous study, by taking advantage of the variable mesh distribution, the solution obtained with the level-symmetric quadrature set presented a maximum relative error of ~6% in zone 1. In the current study, the solution obtained with the level-symmetric quadrature set and uniform spatial mesh yields a maximum relative error of ~10% in zone 1.

In zone 1, the P_N-EW and P_N-T_N quadrature sets underestimate the scalar flux by ~51.9% and ~8.5%, respectively, on the last point of zone 1; this is due to the fact that the P_N-EW quadrature set has fewer directions clustered around the *y*-axis as compared to the P_N-T_N and level-symmetric quadrature sets.

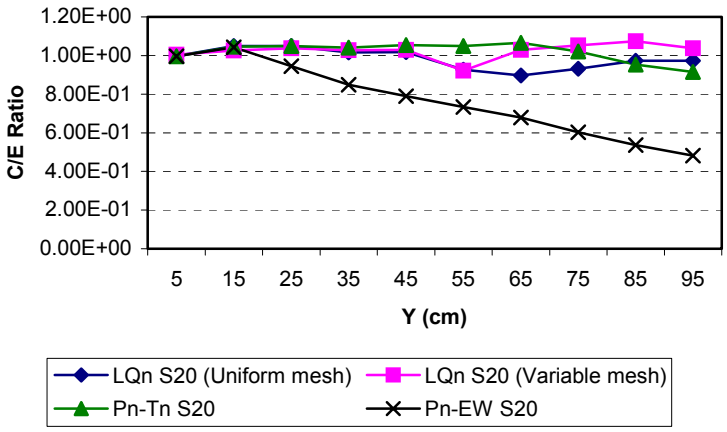

Figure 3-7. Comparison of S_{20} quadrature sets in zone 1 at x=5.0 cm and z=5.0 cm.

Figure 3-8 compares the scalar flux obtained in zone 2 of the benchmark problem. While using a uniform spatial mesh, the P_N-T_N quadrature set yields slightly better results compared to level-symmetric.

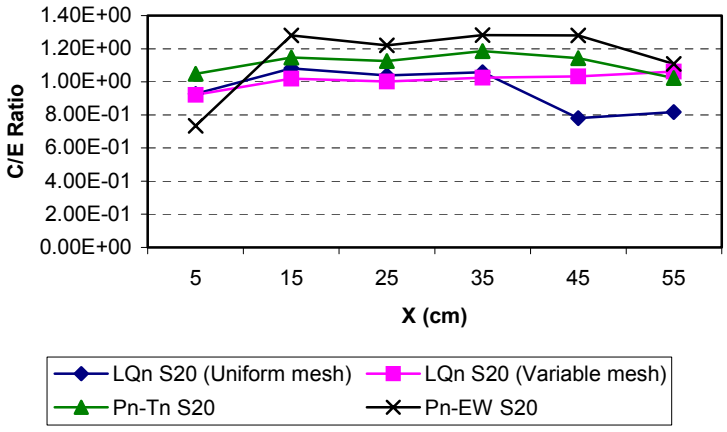

Figure 3-8. Comparison of S_{20} quadrature sets in zone 2 at y=55.0 cm and z=5.0 cm.

In zone 2, the maximum relative error obtained with P_N-T_N is ~18.6%, while for level-symmetric it is ~21.9% using the uniform mesh distribution, and ~6% using variable meshing. However, an error of ~28.2% is observed for the P_N-EW quadrature set.

Figures 3-9 and 3-10, show the solutions obtained with the P_N-EW quadrature set for different S_N orders compared to level-symmetric S_{20}, in zone 1 and 2 respectively.

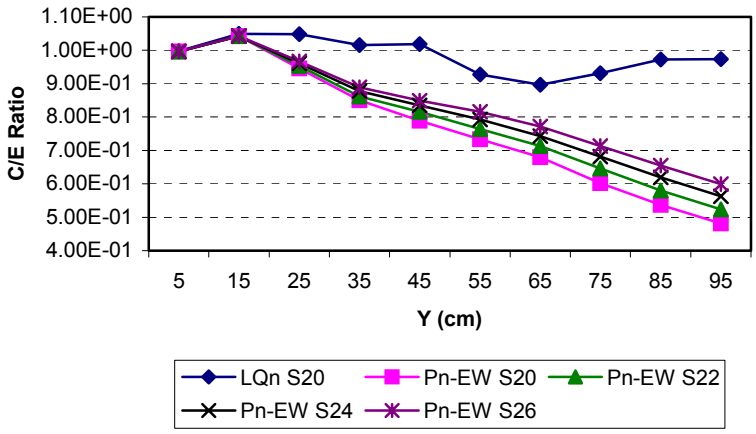

Figure 3-9. Comparison of P_N-EW quadrature sets for different S_N orders in zone 1 at x=5.0 cm and z=5.0 cm.

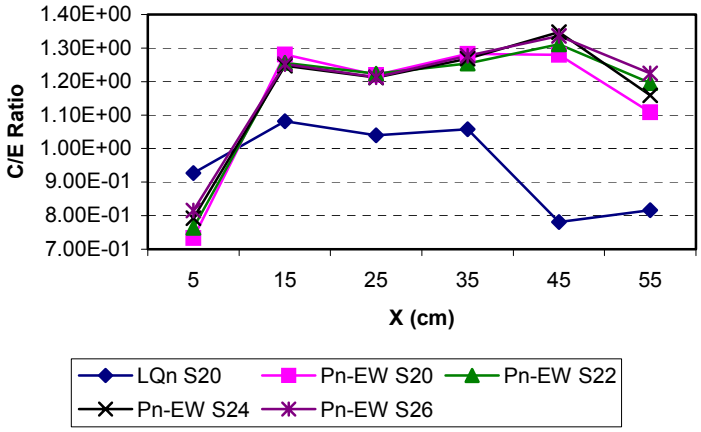

Figure 3-10. Comparison of P_N-EW quadrature sets for different S_N orders in zone 2 at y=55.0 cm and z=5.0 cm.

In zone 1 (Figure 3-9), the P_N-EW is not as accurate as level-symmetric, because fewer directions are clustered near the y-axis; however in zone 1, the solution improves somewhat by increasing the S_N order. In zone 2 (Figure 3-10) the P_N-EW set yields inaccurate results, with a maximum relative error of ~36% for the S_{20} case.

Figure 3-11 compares the ratios of different computed solutions to the exact solution; the computed solutions wered obtained with the P_N-T_N quadrature set for orders S_{20}, S_{22}, S_{24}, S_{26} and with the S_{20} level-symmetric quadrature set. It appears that the increase in the quadrature order does not have a noticeable effect in improving the

accuracy. However, this behavior can be attributed to the fact we have retained the same spatial mesh discretization.

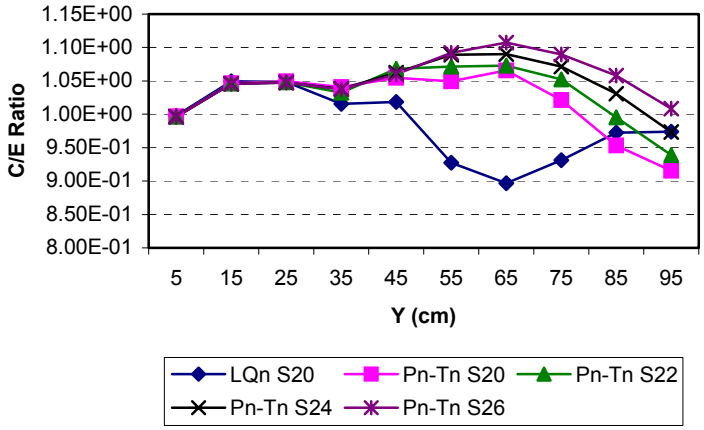

Figure 3-11. Comparison of P_N-T_N quadrature sets for different S_N orders in zone 1 at y=5.0 cm and z=5.0 cm.

In zone 2 (Figure 3-12), the solution obtained with an S_{22} P_N-T_N quadrature set is more accurate than what obtained with level-symmetric. The S_{22} P_N-T_N yields a maximum relative error of ~9% compared to ~22% from level-symmetric. Again, in zone 2, the accuracy somewhat decreases as the S_N order increases, because the spatial mesh is not consistently refined.

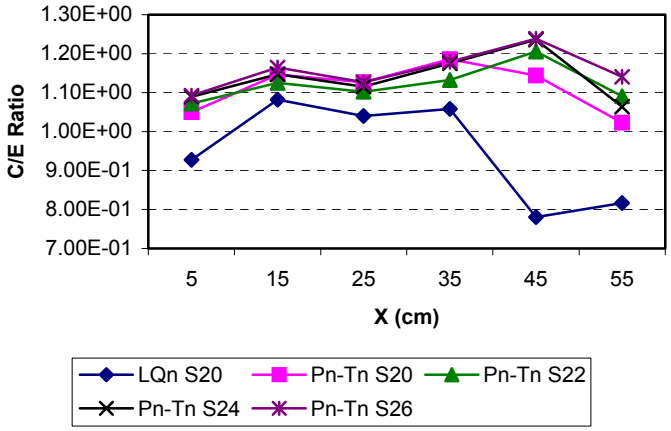

Figure 3-12. Comparison of P_N-T_N quadrature sets for different S_N orders in zone 2 at y=55.0 cm and z=5.0 cm.

3.5.2 CT-Scan Device for Medical/Industrial Imaging Applications

The model of a CT-Scan device used for medical/industrial applications is used in this section to verify the accuracy and performance of the RAR technique. A CT-Scan device utilizes a collimated x-ray source (fan-beam) to scan an object or a patient. The main components of a CT-Scan device are an x-ray source mounted on a rotating gantry and an array of sensors. The patient is positioned on a sliding bed that is moved inside the CT-Scan. The mesh distribution for this model, is shown in Figure 3-13.

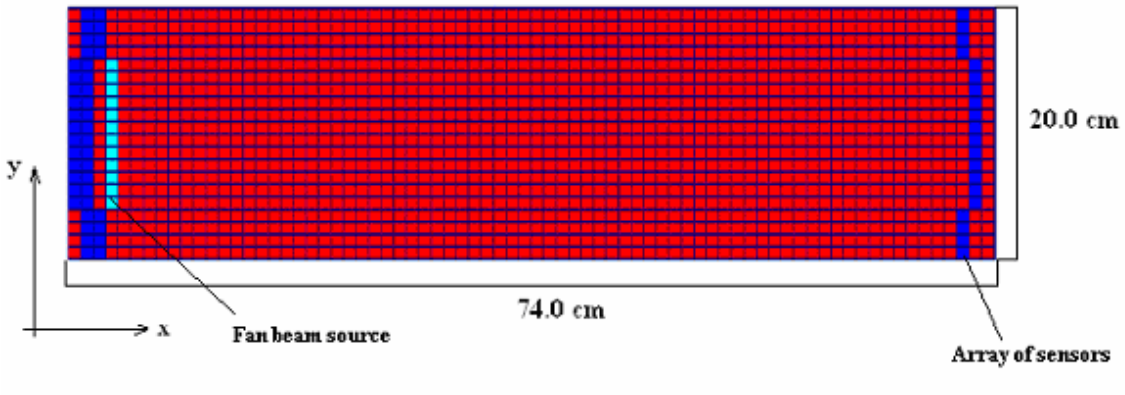

Figure 3-13. Cross-sectional view of the CT-Scan model on the *x*-*y* plane.

Figure 3-13 shows the simplified PENTRAN model which represents the x-ray directional source ("fan" beam), a large region of air and an array of sensors. The size of this model is 74 cm along the *y*-axis and 20 cm along the *x*-axis. The array of detectors is located at 72 cm from the source along the *x*-axis.

The materials are described using one-group cross-sections from the 20-group gamma of the BUGLE-96 cross-sections library. The group corresponds to an x-ray source emitting photons in an energy range of 100 KeV to 200 KeV. The cross-sections were prepared using a P_3 expansion for the scattering kernel.

Because of the presence of large void regions and a directional source, the solution of the transport equation is significantly affected by the ray-effects.[3] One remedy is to use

high order quadrature sets with biasing, such as RAR. We compared the solutions obtained with an S_{50} P_N-T_N quadrature set. The RAR technique has been applied to an S_{30} P_N-T_N quadrature set; the biased region on the positive octant extends from z=0.0 cm to z=0.3 cm and the azimuthal angle extends from 0.0 to 5.0 degrees. In the biased region an S_{10} P_N-T_N quadrature set is used. The P_N-T_N quadrature set biased with RAR resulted in 142 directions per octant. The unbiased S_{50} P_N-T_N quadrature set yielded 325 directions per octant. The S_{20} level-symmetric quadrature set yielded 55 directions per octant.

Figures 3-14, 3-15, and 3-16 show the flux distributions in the x-y plane, obtained with the level-symmetric S_{20}, S_{50} P_N-T_N and S_{30} P_N-T_N with RAR, respectively.

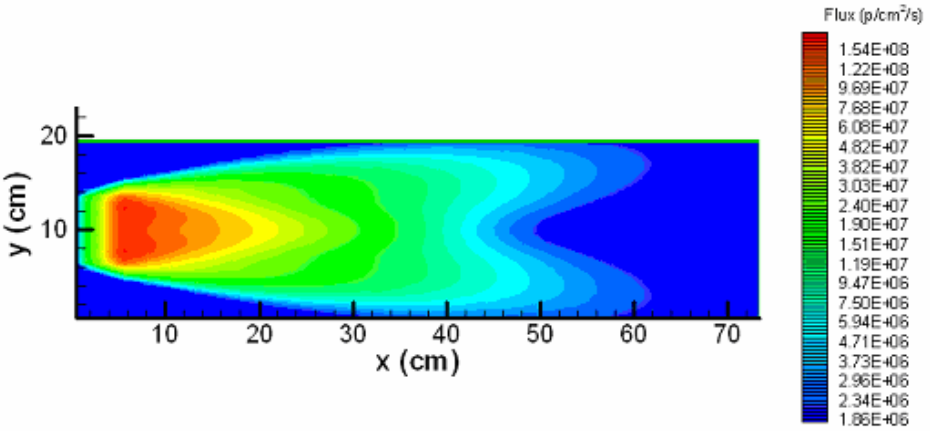

Figure 3-14. Scalar flux distribution on the x-y plane obtained with an S_{20} level-symmetric quadrature set.

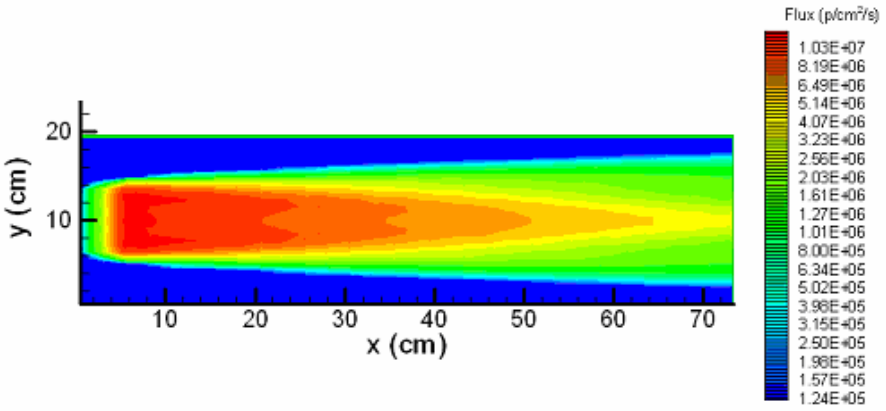

Figure 3-15. Scalar flux distribution on the x-y plane obtained with an S_{50} P_N-T_N quadrature set.

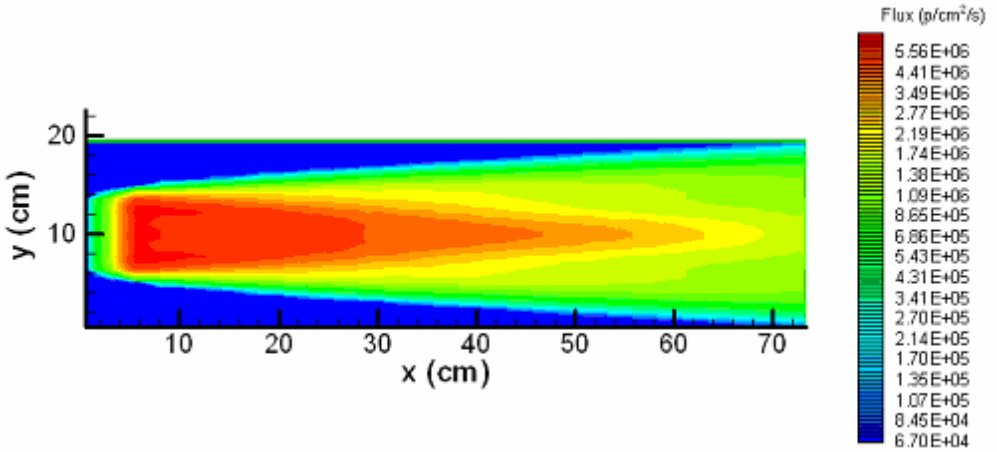

Figure 3-16. Scalar flux distribution on the *x*-*y* plane obtained with an S_{30} P_N-T_N quadrature set biased with RAR.

The above results indicate that the level-symmetric quadrature set exhibits significant ray-effects, while S_{50} P_N-T_N and S_{30} P_N-T_N with RAR quadrature sets, yield similar solutions without any ray-effects. The main advantage of using a biased quadrature set is the significant reduction in computational cost and memory requirement. Table 3-3 compares the CPU time and memory requirements for the three calculations presented above.

Table 3-3. CPU time and total number of directions required for the CT-Scan simulation.

Quadrature Set	Directions	CPU Time(sec)	Memory ratio[a]	Time ratio[a]
S_{50} P_N-T_N	2600	166.4	1.0	1.0
S_{30} P_N-T_N RAR (S_{10})	1136	79.4	0.51	0.47
S_{20} LQ_N	440	33.3	0.2	0.2

[a] memory and time ratio are referred to the S_{50} P_N-T_N quadrature set.

The RAR technique lessens the ray effect in the flux distribution and greatly reduces the computational time by more than a factor of 2 compared to S_{50}.

The new quadrature sets biased with the OS rather than the RAR technique have also been examined based on the CT-Scan model.[9] Figure 3-17 compares the results of PENTRAN with a reference Monte Carlo solution. For all cases, the first direction of the

lowest level in quadrature set is split in 9 or 25 directions; for example, P_N-T_N 22-2-55 corresponds to P_N-T_N S_{22} with direction 55 split in 9 directions. All the quadrature sets biased with the OS technique yield accurate results within the statistical uncertainty of the Monte Carlo predictions.

Due to the significant ray-effects, the level-symmetric S_{20} quadrature set without ordinate splitting yields poor accuracy. Note that, even by using high order quadrature sets, such as P_N-T_N S_{28} (840 directions), the solution at detector position is under predicted by ~21%.

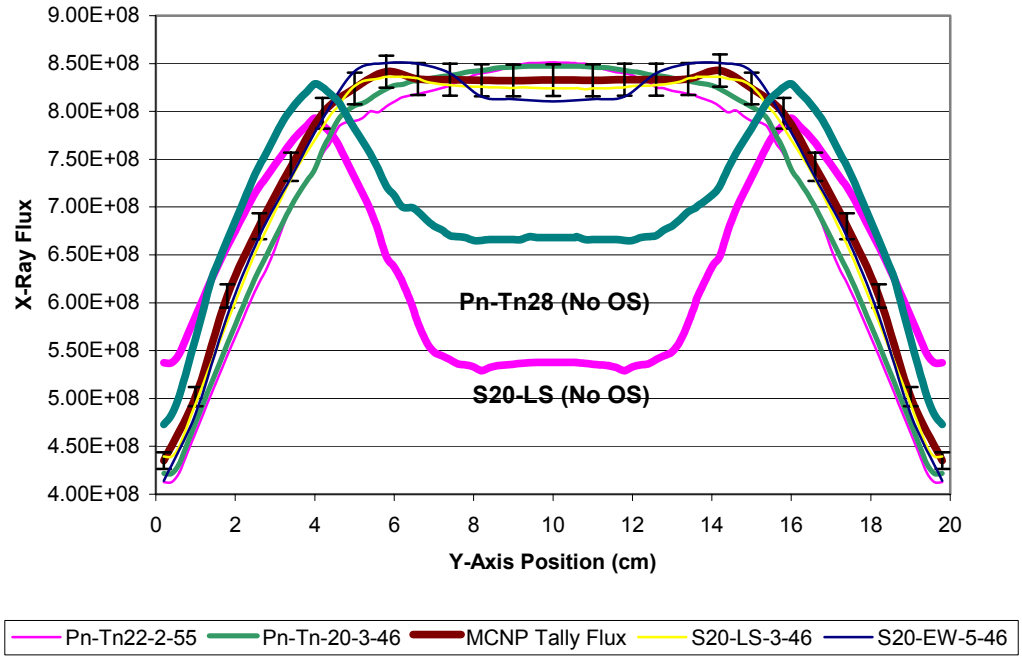

Figure 3.17. Comparison of the scalar flux at detector position (x=72.0 cm).

CHAPTER 4
DERIVATION OF THE EVEN-PARITY SIMPLIFIED S_N EQUATIONS

This chapter presents the initial derivation of the Simplified Spherical Harmonics (SP_N) equations starting from the P_N equations in 1-D geometry, and it discusses the issues related to the coupling of the SP_N moments on the vacuum boundary conditions. Because of this peculiarity, the implementation of the general SP_N equations into a computer code proved to be cumbersome. However, I will present the initial derivation of the SP_3 equations, successively implemented into a new computer code named $PENSP_3$ (Parallel Environment Neutral-particles SP_3).

To overcome the difficulties related to the coupling of the SP_N moments in the vacuum boundary conditions, I adopted a different formulation based on the Even-Parity Simplified S_N ($EP\text{-}SS_N$) equations. These equations are derived starting from the 1-D S_N equations, and using the same assumptions made for the derivation of the SP_N equations; however, the main advantage of this formulation is the natural decoupling of the even-parity angular fluxes for the vacuum boundary conditions.

Therefore, a Fourier analysis of the $EP\text{-}SS_N$ equations will follow, along with the derivation of a new formulation to accelerate the convergence of the source iteration method applied to the $EP\text{-}SS_N$ equations. This chapter is concluded with the derivation of the 3-D P_1 equations. I will compare the P_1 equations with the SP_1 equations, and I will describe the assumptions made in the derivation of the SP_1 equations and the relation with the spherical harmonics P_1 formulation.

4.1 Derivation of the Simplified Spherical Harmonics (SP_N) Equations

The SP_N equations were initially proposed by Gelbard[18] in the early 1960s. However, they did not receive much attention due to the weak theoretical support. Recently, the theoretical foundations of the SP_N equations have been significantly strengthened using a variational analysis approach.[21-22]

I derive the multigroup SP_N equations starting from the 1-D multigroup P_N equations and by applying the procedure originally outlined by Gelbard. The multigroup 1-D P_N equations are given by

$$\frac{n+1}{2n+1}\frac{\partial \phi_{n+1,g}(x)}{\partial x}+\frac{n}{2n+1}\frac{\partial \phi_{n-1,g}(x)}{\partial x}+\left[\sigma_{t,g}(x)-\sigma_{sn,g\to g}(x)\right]\phi_{n,g}(x)=q_{n,g}(x), \qquad (4.1)$$

$$\text{for } n=0, N \text{ and } g=1, G,$$

where

$$q_{n,g}(x)=\sum_{\substack{g'=1\\ g'\neq g}}^{G}\sum_{l=0}^{L}\frac{2l+1}{4\pi}P_l(\mu)\sigma_{sl,g'\to g}(x)\phi_{l,g'}(x)+q_{f,g}(x)+S_{n,g}^{ext}(x). \qquad (4.2)$$

In Eqs. 4.1 and 4.2, I defined the following quantities:

$\sigma_{t,g}(x)$, total macroscopic cross-section in group *g*.
$\sigma_{sn,g\to g}(x)$, Legendre moment of the in-group macroscopic scattering cross-section of order *n*.
$\sigma_{sl,g'\to g}(x)$, Legendre moment of the group transfer macroscopic scattering cross-section of order *l*.
$\phi_{n,g}(x)$, Legendre moment of the angular flux of order *n*.
$q_{f,g}(x)$, fission source.
$S_{n,g}^{ext}(x)$, Legendre moment of the inhomogeneous source of order *n*.
G, total number of energy groups and
L, order of the Legendre expansion of the macroscopic scattering cross-section ($L<N$).

In Eq. 4.1, the angular flux is expanded in terms of Legendre polynomials as

$$\psi_g(x,\mu) = \sum_{l=0}^{L} \frac{2l+1}{4\pi} P_l(\mu)\phi_{l,g}(x)\,. \tag{4.3}$$

In Eq. 4.1, the term $\frac{d\phi_{N+1,g}}{dx}$ is defined to be identically zero when $n=N$. This assumption closes the P_N equations, yielding $N+1$ equations with $N+1$ unknowns. The procedure prescribed by Gelbard to obtain the 3-D SP_N equations from the 1-D P_N, consists of the following steps:

4. Replace the partial derivative operator in Eq. 4.1 for *even n* with the *divergence operator* $(\vec{\nabla}\cdot)$.
5. Replace the partial derivative operator in Eq. 4.1 for *odd n* with the *gradient operator* $(\vec{\nabla})$.

By applying this procedure to Eq. 4.1, it reduces to

$$\frac{n+1}{2n+1}\vec{\nabla}\cdot\vec{\phi}_{n+1,g}(\vec{r}) + \frac{n}{2n+1}\vec{\nabla}\cdot\vec{\phi}_{n-1,g}(\vec{r}) + \left[\sigma_{t,g}(\vec{r}) - \sigma_{sn,g\to g}(\vec{r})\right]\phi_{n,g}(\vec{r}) = q_{n,g}(\vec{r})\,, \tag{4.4a}$$

for $n=0,2,\ldots,N-1$, $g=1,G$, and $\vec{r}\in V$,

$$\frac{n+1}{2n+1}\vec{\nabla}\phi_{n+1,g}(\vec{r}) + \frac{n}{2n+1}\vec{\nabla}\phi_{n-1,g}(\vec{r}) + \left[\sigma_{t,g}(\vec{r}) - \sigma_{sn,g\to g}(\vec{r})\right]\vec{\phi}_{n,g}(\vec{r}) = \vec{q}_{n,g}(\vec{r})\,, \tag{4.4b}$$

for $n=1,3,\ldots,N$, $g=1,G$, and $\vec{r}\in V$.

The SP_N equations can be reformulated in terms of a second-order elliptic operator, if one solves for the odd-parity moments using Eqs. 4.4b , i.e.,

$$\vec{\phi}_{n,g}(\vec{r}) = -\frac{1}{\sigma_{an,g}}\frac{n+1}{2n+1}\vec{\nabla}\phi_{n+1,g}(\vec{r}) - \frac{1}{\sigma_{an,g}}\frac{n}{2n+1}\vec{\nabla}\phi_{n-1,g}(\vec{r}) + \frac{\vec{q}_{n,g}(\vec{r})}{\sigma_{an,g}}, \tag{4.5}$$

where

$$\sigma_{an,g} = \sigma_{t,g}(\vec{r}) - \sigma_{sn,g\to g}(\vec{r})\,. \tag{4.6}$$

and then substitute Eqs. 4.5 into Eqs. 4.4a to obtain

$$
\begin{aligned}
&-\frac{n+1}{2n+1}\frac{n+2}{2n+3}\vec{\nabla}\cdot\frac{1}{\sigma_{an+1,g}}\vec{\nabla}\phi_{n+2,g}-\frac{n+1}{2n+1}\frac{n+1}{2n+3}\vec{\nabla}\cdot\frac{1}{\sigma_{an+1,g}}\vec{\nabla}\phi_{n,g}\\
&-\frac{n}{2n+1}\frac{n}{2n-1}\vec{\nabla}\cdot\frac{1}{\sigma_{an-1,g}}\vec{\nabla}\phi_{n,g}-\frac{n}{2n+1}\frac{n-1}{2n-1}\vec{\nabla}\cdot\frac{1}{\sigma_{an-1,g}}\vec{\nabla}\phi_{n-2,g}\\
&+\sigma_{an-1,g}\phi_{n,g}=q_{n,g}-\frac{n+1}{2n+1}\vec{\nabla}\cdot\frac{\vec{q}_{n+1,g}}{\sigma_{an+1,g}}-\frac{n}{2n+1}\vec{\nabla}\cdot\frac{\vec{q}_{n-1,g}}{\sigma_{an-1,g}},
\end{aligned}
\tag{4.7}
$$

for n=0,2,...,N-1 and g=1, G.

Note that for simplicity, the spatial dependency of ϕ has been eliminated. The SP_N equations yield a system of $(N+1)/2$ coupled partial differential equations that can be solved using standard iterative methods, such as Gauss-Seidel or Krylov subspace methods. The main disadvantage of this formulation is that it yields Marshak-like vacuum boundary conditions, coupled through the moments. Because of this issue the implementation of this formulation into a computer code becomes cumbersome. However, to study the effectiveness of the SP_N method, I developed a 3-D parallel SP_3 code,[25] $PENSP_3$ (Parallel Environment Neutral-particles SP_3). The $PENSP_3$ code is based on Eqs. 4.8a and 4.8b, which are derived assuming isotropic scattering and isotropic inhomogeneous source.

$$
\begin{aligned}
&-\vec{\nabla}\cdot D_{1,g}(\vec{r})\vec{\nabla}F(\vec{r})+\sigma_{t,g}(\vec{r})F(\vec{r})=2\sigma_{t,g}(\vec{r})\phi_{2,g}(\vec{r})+\sigma_{s0,g\to g}(\vec{r})\phi_{0,g}(\vec{r})\\
&+\sum_{\substack{g'=1\\ g'\neq g}}^{G}\sigma_{s0,g'\to g}(\vec{r})\phi_{0,g'}(\vec{r})+\frac{1}{k}\chi_g\sum_{g'=1}^{G}\nu\sigma_{f,g'}(\vec{r})\phi_{0,g'}(\vec{r})+S^{ext}_{0,g}(\vec{r}),
\end{aligned}
\tag{4.8a}
$$

$$
\begin{aligned}
&-\vec{\nabla}D_{2,g}(\vec{r})\vec{\nabla}\phi_{2,g}(\vec{r})+\sigma_{t,g}(\vec{r})\phi_{2,g}(\vec{r})=\\
&\frac{2}{5}\left(\sigma_{a0,g}(\vec{r})\phi_{0,g}(\vec{r})-\sum_{\substack{g'=1\\ g'\neq g}}^{G}\sigma_{s0,g'\to g}(\vec{r})\phi_{0,g'}(\vec{r})-\frac{1}{k}\chi_g\sum_{g'=1}^{G}\nu\sigma_{f,g'}(\vec{r})\phi_{0,g'}(\vec{r})-S^{ext}_{0,g}(\vec{r})\right),
\end{aligned}
\tag{4.8b}
$$

for g=1, G,

where

$$F_g(\vec{r}) = 2\phi_{2,g}(\vec{r}) + \phi_{0,g}(\vec{r}),\ D_{1,g}(\vec{r}) = \frac{1}{3\sigma_{t,g}(\vec{r})}, \text{ and } D_{2,g}(\vec{r}) = \frac{9}{35\sigma_{t,g}(\vec{r})}. \tag{4.9}$$

The SP_3 Marshak-like vacuum boundary conditions for Eqs. 4.8a and 4.8b are given by

$$\frac{1}{2}F_g(\vec{r}) + D_{1,g}(\vec{n}\cdot\vec{\nabla})F_g(\vec{r}) = \frac{3}{8}\phi_{2,g}(\vec{r}) + \int_0^{2\pi}\int_{-1}^{0} 2|\mu|\psi^b(\vec{r},\mu,\varphi)d\mu d\varphi\,, \tag{4.10a}$$

$$\frac{21}{40}\phi_{2,g}(\vec{r}) + D_{2,g}(\vec{n}\cdot\vec{\nabla})\phi_{2,g}(\vec{r}) = \frac{3}{40}F_g(\vec{r}) + \frac{3}{5}\int_0^{2\pi}\int_{-1}^{0} 2P_3(|\mu|)\psi^b(\vec{r},\mu,\varphi)d\mu d\varphi\,. \tag{4.10b}$$

Eqs. 4.10a and 4.10b are termed Marshak-like boundary conditions, because in 1-D geometry they reduce to standard Marshak boundary conditions. Implementation of these formulations into a computer code is difficult because of the coupling of the SP_N moments.

The reflective boundary condition is represented by setting the odd-moments equal to zero on the boundary, i.e.,

$$\hat{n}\cdot\vec{\phi}_{n,g}(\vec{r}_b) = 0\,, \text{ for } n \text{ odd}, \tag{4.11}$$

where $\vec{r}_b \in \partial V$ and $\hat{n}$ is the normal to the surface considered.

4.2 Derivation of the Even-Parity Simplified S_N ($EP\text{-}SS_N$) Equations

I have derived the Even-Parity Simplified S_N ($EP\text{-}SS_N$) equations starting from the 1-D S_N equations given by

$$\mu_m \frac{\partial\psi(x,\mu_m)}{\partial x} + \sigma_t(x)\psi(x,\mu_m) = Q(x,\mu_m)\,, \text{ for } m=1,\ N, \tag{4.12}$$

where

$$Q(x,\mu_m) = \sum_{l=0}^{L}(2l+1)P_l(\mu_m)\sigma_{sl}(x)\phi_l(x) + \sum_{l=0}^{L}(2l+1)P_l(\mu_m)S_l(x) + q_f(x)\,, \tag{4.13}$$

and

$$q_f(x) = \frac{1}{k}\nu\sigma_f(x)\phi_0(x)\,;\ \phi_l(x) = \frac{1}{2}\sum_{m=1}^{N} w_m P_l(\mu_m)\psi(x,\mu_m). \tag{4.14}$$

A Gauss-Legendre symmetric quadrature set (P_N) is considered, where $\mu_m \in (-1,1)$, $\sum_{m=1}^{M} w_m = 2.0$, and *M=N(N+2)*. In Eq. 4.13, *L* is the order of the Legendre expansion for both the macroscopic scattering cross-section and the inhomogeneous source *(L<N)*. Therefore, the even- and odd-parity angular fluxes are defined by

$$\psi_m^E = \frac{1}{2}\left[\psi(x,\mu_m) + \psi(x,-\mu_m)\right] \text{ (even)}, \tag{4.15a}$$

$$\psi_m^O = \frac{1}{2}\left[\psi(x,\mu_m) - \psi(x,-\mu_m)\right] \text{ (odd)}. \tag{4.15b}$$

To reformulate the 1-D S_N equations in terms of the even- and odd-parity angular fluxes, I rewrite Eq. 4.12 for $-\mu_m$ as

$$-\mu_m \frac{\partial\psi(x,-\mu_m)}{\partial x} + \sigma_t(x)\psi(x,-\mu_m) = Q(x,-\mu_m), \text{ for } m=1, N. \tag{4.16}$$

Then, I add Eqs. 4.12 and Eq. 4.16 to obtain

$$\mu_m \frac{\partial\left[\psi(x,\mu_m) - \psi(x,-\mu_m)\right]}{\partial x} + \sigma_t(x)\left[\psi(x,\mu_m) + \psi(x,-\mu_m)\right] = Q(x,\mu_m) + Q(x,-\mu_m), \tag{4.17}$$

and use the definitions of even- and odd-parity angular fluxes (given by Eqs. 4.15a and 4.15b), to obtain

$$\begin{aligned} 2\mu_m \frac{\partial}{\partial x}\psi_m^O(x) + 2\sigma_t(x)\psi_m^E(x) = &\sum_{l=0}^{L}(2l+1)\left[P_l(\mu_m) + P_l(-\mu_m)\right]\sigma_{sl}(x)\phi_l(x) + \\ &\sum_{l=0}^{L}(2l+1)\left[P_l(\mu_m) + P_l(-\mu_m)\right]S_l(x) + 2q_f(x). \end{aligned} \tag{4.18}$$

Consider the following identities for the Legendre polynomials.[3]

$$P_l(-\mu) = P_l(\mu), \text{ for } l \textit{ even}, \tag{4.19a}$$

$$P_l(-\mu) = -P_l(\mu)\text{, for } l \textit{ odd}. \tag{4.19b}$$

Eq. 4.18 can be rewritten as

$$\mu_m \frac{\partial}{\partial x}\psi_m^O(x) + \sigma_t(x)\psi_m^E(x) = \sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(x)\phi_l(x) + S_l(x)\right] + q_f(x)\,. \tag{4.20}$$

Similarly, by subtracting Eq. 4.16 from Eq. 4.12, I obtain

$$\mu_m \frac{\partial\left[\psi(x,\mu_m)+\psi(x,-\mu_m)\right]}{\partial x} + \sigma_t(x)\left[\psi(x,\mu_m)-\psi(x,-\mu_m)\right] = Q(x,\mu_m) - Q(x,-\mu_m)\,, \tag{4.21}$$

and by using the definitions of even- and odd-parity angular fluxes, I obtain

$$\begin{aligned} 2\mu_m \frac{\partial}{\partial x}\psi_m^E(x) + 2\sigma_t(x)\psi_m^O(x) = &\sum_{l=0}^{L}(2l+1)\left[P_l(\mu_m) - P_l(-\mu_m)\right]\sigma_{sl}(x)\phi_l(x) + \\ &\sum_{l=0}^{L}(2l+1)\left[P_l(\mu_m) - P_l(-\mu_m)\right]S_l(x). \end{aligned} \tag{4.22}$$

Following the use of the Legendre polynomial identities (Eqs. 4.19a and 4.19b), Eq. 4.22 reduces to

$$\mu_m \frac{\partial}{\partial x}\psi_m^E(x) + \sigma_t(x)\psi_m^O(x) = \sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(x)\phi_l(x) + S_l(x)\right]. \tag{4.23}$$

Now, the odd-parity angular fluxes are then obtained from Eq. 4.23 as

$$\psi_m^O(x) = -\frac{\mu_m}{\sigma_t(x)}\frac{\partial}{\partial x}\psi_m^E(x) + \frac{1}{\sigma_t(x)}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(x)\phi_l(x) + S_l(x)\right]. \tag{4.24}$$

Then, using Eq. 4.24 in Eq. 4.20, I obtain

$$\begin{aligned} -\frac{\partial}{\partial x}\frac{\mu_m^2}{\sigma_t(x)}\frac{\partial}{\partial x}\psi_m^E(x) + \sigma_t(x)\psi_m^E(x) = &\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(x)\phi_l(x) + S_l(x)\right] \\ &-\frac{\partial}{\partial x}\frac{\mu_m}{\sigma_t(x)}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(x)\phi_l(x) + S_l(x)\right] + q_f(x). \end{aligned} \tag{4.25}$$

Finally, the EP-SS_N equations in 3-D Cartesian geometry with anisotropic scattering kernel and anisotropic inhomogeneous source of arbitrary order L, are obtained by applying the procedure outlined by Gelbard (i.e., substitution of first order partial differential operators with the gradient operator) to Eq. 4.25.

$$-\vec{\nabla}\cdot\left(\frac{\mu_m^2}{\sigma_t(\vec{r})}\vec{\nabla}\psi_m^E(\vec{r})\right)+\sigma_t(\vec{r})\psi_m^E(\vec{r})=\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(\vec{r})\phi_l(\vec{r})+S_l(\vec{r})\right]$$
$$-\vec{\nabla}\cdot\left(\frac{\mu_m}{\sigma_t(\vec{r})}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(\vec{r})\vec{\phi}_l(\vec{r})+\vec{S}_l(\vec{r})\right]\right)+q_f(\vec{r}), \tag{4.26}$$

for m=1, $N/2$,

where

$$\vec{\psi}_m^O(\vec{r})=-\frac{\mu_m}{\sigma_t(\vec{r})}\vec{\nabla}\psi_m^E(\vec{r})+\frac{1}{\sigma_t(\vec{r})}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl}(\vec{r})\vec{\phi}_l(\vec{r})+\vec{S}_l(\vec{r})\right]. \tag{4.27}$$

Due to the symmetry of the Gauss-Legendre quadrature set, the EP-SS_N equations only need to be solved on half of the angular domain, e.g. $\mu\in(0,1)$. The moments of the even- and odd-parity angular fluxes are evaluated by

$$\phi_l(\vec{r})=\sum_{m=1}^{N/2}w_m P_l(\mu_m)\psi_m^E(\vec{r}),\ \text{for } l \textit{ even}, \tag{4.28a}$$

and

$$\vec{\phi}_l(\vec{r})=\sum_{m=1}^{N/2}w_m P_l(\mu_m)\vec{\psi}_m^O(\vec{r}),\ \text{for } l \textit{ odd}. \tag{4.28b}$$

The multigroup form of Eqs. 4.26 and 4.27 with anisotropic scattering and source are written as

$$-\vec{\nabla}\cdot\left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})}\vec{\nabla}\psi_{m,g}^{E}(\vec{r})\right)+\sigma_{t,g}(\vec{r})\psi_{m,g}^{E}(\vec{r})=\sum_{g'=1}^{G}\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g}(\vec{r})\phi_{l,g'}(\vec{r})$$

$$-\vec{\nabla}\cdot\left(\frac{\mu_m}{\sigma_{t,g}(\vec{r})}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g}(\vec{r})\vec{\phi}_{l,g'}(\vec{r})\right)+\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)S_{l,g}(\vec{r})$$

$$-\vec{\nabla}\cdot\left(\frac{\mu_m}{\sigma_{t,g}(\vec{r})}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\vec{S}_{l,g'}(\vec{r})\right)+q_{f,g}(\vec{r})$$

(4.29)

and

$$\vec{\psi}_{m,g}^{O}(\vec{r})=-\frac{\mu_m}{\sigma_{t,g}(\vec{r})}\vec{\nabla}\psi_{m,g}^{E}(\vec{r})+\frac{1}{\sigma_{t,g}(\vec{r})}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g}(\vec{r})\vec{\phi}_{l,g'}(\vec{r})+\vec{S}_{l,g'}(\vec{r})\right]$$

(4.30)

for m=1, *N/2* and g=1, *G*.

4.2.1 Boundary Conditions for the EP-SS$_N$ Equations

The boundary conditions for the EP-SS$_N$ equations are based on the assumption that the angular flux on the boundary surface is azimuthally symmetric about the surface normal vector. The EP-SS$_N$ boundary conditions follow directly from the 1-D even-parity S$_N$ boundary conditions. Hence, by considering the positive half of the angular domain for the EP-SS$_N$ equations, the 1-D source boundary condition at the right boundary face is given by

$$\psi^{E}(x_s,\mu_m)-\psi^{O}(x_s,\mu_m)=\psi(x_s,-\mu_m)\,, \tag{4.31}$$

and the corresponding condition in 3-D is given by

$$\psi^{E}(\vec{r}_s,\mu_m)-\vec{n}\cdot\vec{\psi}^{O}(\vec{r}_s,\mu_m)=\psi(\vec{r}_s,-\mu_m)\,. \tag{4.32}$$

Note that in 3-D geometry, the EP-SS_N formulation requires the incoming boundary flux to be azimuthally symmetric about the surface normal vector. The 3-D albedo boundary condition is given by

$$\vec{n}\cdot\vec{\psi}^{O}(\vec{r}_s,\mu_m)=\frac{1-\alpha}{1+\alpha}\psi^{E}(\vec{r}_s,\mu_m)\,. \tag{4.33}$$

Note that in Eq. 4.33, the vacuum boundary condition is obtained by setting $\alpha=0$ while the reflective boundary condition is obtained by setting $\alpha=1$.

Note that the main advantage of the EP-SS_N formulation compared to SP_N is the decoupling of the even-parity angular fluxes for the vacuum boundary conditions.

4.2.2 Fourier Analysis of the EP-SS_N Equations

The EP-SS_N equations are solved iteratively using the source iteration method. This method is based on performing iterative cycles on the scattering source; moreover, the method has a clear physical interpretation that allows one to predict classes of problems where it should yield fast convergence. The source iteration method for the EP-SS_N equations is defined by

$$H_{L,m,g}(\vec{r})\psi_{m,g}^{E,l+1}(\vec{r})=q_{s,m,g\to g}^{l}+q'\text{, for } m\text{=1, } N/2 \text{ and } g\text{=1, } G, \tag{4.34}$$

where, $H_{L,m,g}$ is the EP-SS_N leakage plus collision operator, $q_{s,m,g\to g}^{l}$ is the in-group scattering source, and q' is a fixed source term that includes scattering transfers from energy groups other than *g*, the external source and fission sources. The iterative method begins by assuming a flux guess; then, Eq. 4.34 is solved for $\psi_{m,g}^{E,l+1}$ and the in-group scattering source is updated. This process continues until a certain convergence criterion is satisfied.

The convergence rate of any iterative method is characterized by the *spectral radius*. For the source iteration method, in an infinite homogeneous medium, it is well known that the spectral radius is equal to the *scattering ratio (c)*, given by

$$c_g = \frac{\sigma_{s,g\to g}}{\sigma_{t,g}}. \tag{4.35}$$

The scattering ratio is bounded between 0 and 1; hence, a *c*-ratio approaching one means that the problem will converge slowly, while, oppositely, a *c*-ratio close to zero, indicates a fast converging problem.

Fourier analysis is the tool of choice to analyze the convergence behavior of iterative methods. For simplicity, I will consider the 1-D EP-SS_N equations with isotropic scattering and source, given by

$$-\frac{\partial}{\partial x}\frac{\mu^2}{\sigma_t(x)}\frac{\partial}{\partial x}\psi^E_{l+1/2}(x,\mu)+\sigma_t(x)\psi^E_{l+1/2}(x,\mu)=\sigma_{s0}(x)\phi_{0,l}(x)+S_{ext}(x), \tag{4.36}$$

which following division by $\sigma_t(x)$, reduces to

$$-\frac{\partial}{\partial x}\frac{\mu^2}{\sigma_t^2(x)}\frac{\partial}{\partial x}\psi^E_{l+1/2}(x,\mu)+\psi^E_{l+1/2}(x,\mu)=c\phi_{0,l}(x)+\frac{S_{ext}(x)}{\sigma_t(x)}, \tag{4.37}$$

where *c* is the scattering ratio defined by Eq. 4.35.

The EP-SS_N equations can be rewritten in terms of the error between two consecutive iterations as

$$-\frac{\partial}{\partial x}\frac{\mu^2}{\sigma_t^2(x)}\frac{\partial}{\partial x}\varepsilon^E_{l+1/2}(x,\mu)+\varepsilon^E_{l+1/2}(x,\mu)=c\delta_{0,l}(x), \tag{4.38}$$

where

$$\varepsilon^E_{l+1/2}(x,\mu)=\psi^E_{l+1/2}(x,\mu)-\psi^E_{l-1/2}(x,\mu), \tag{4.39}$$

and

$$\delta_{0,l}(x) = \phi_{0,l}(x) - \phi_{0,l-1}(x)\,. \tag{4.40}$$

The error terms are then expanded in terms of the Fourier modes, considering an infinite homogeneous medium, the Fourier ansatz is defined as follows

$$\varepsilon^{E}_{l+1/2}(x,\mu) = f(\mu)\exp(i\lambda x)\,,\ \text{and}\ \delta_{0,l}(x) = \exp(i\lambda x)\,, \tag{4.41}$$

where

$$i = \sqrt{-1}\ \text{ and }\ \lambda \in (-\infty, \infty).$$

By substituting the above relations into Eq. 4.38, I obtain the EP-SS_N equations mapped onto the frequency domain, resulting in function $f(\mu)$ given by

$$f(\mu) = \frac{c}{1+\left(\dfrac{\lambda}{\sigma_t}\mu\right)^2}\,. \tag{4.42}$$

Therefore, the spectrum of eigenvalues is obtained by observing that the error in the scalar flux at iteration *l+1* can be written as

$$\delta_{0,l+1}(x) = \int_0^1 d\mu\,\varepsilon^{E}_{l+1/2}(x,\mu) = \exp(i\lambda x)\int_0^1 d\mu f(\mu) = \exp(i\lambda x)\int_0^1 d\mu \frac{c}{1+\left(\dfrac{\lambda}{\sigma_t}\mu\right)^2}\,. \tag{4.43}$$

By performing the integration over the angular variable in Eq. 4.43, the spectrum of eigenvalues is found to be equal to

$$\omega(\lambda) = c\frac{\arctan\left(\dfrac{\lambda}{\sigma_t}\right)}{\dfrac{\lambda}{\sigma_t}}\,. \tag{4.44}$$

The result obtained in Eq. 4.44 is similar to what is obtained for the S_N equations. The spectral radius is found to be equal to $\rho = \max[\omega(\lambda)] = c$. However, the convergence behavior of the EP-SS_N equations is also affected by the value of the total scattering

cross-section. For optically thin media, where the total cross-section assumes small values, Eq. 4.44 suggests that the convergence should be very fast, and in the limit as $\sigma_t \to 0$, the spectral radius will tend to zero.

4.2.3 A New Formulation of the EP-SS$_N$ Equations for Improving the Convergence Rate of the Source Iteration Method

As discussed in the previous paragraph, the performance of the source iteration method applied to the EP-SS$_N$ equations is similar to the S$_N$ equations. However, I have derived a new formulation of the EP- SS$_N$ equations which reduces the spectral radius for the source iteration method. Appendix B addresses the performance of the new formulation for a criticality eigenvalue benchmark problem; note that the new formulation is a key aspect for the successful implementation of an acceleration method for the S$_N$ equations. The main idea behind the new formulation is to remove the in-group component of the scattering kernel for each direction. In order to reformulate the EP- SS$_N$ equations, we note that the even-moments in the in-group portion of the scattering kernel can be expanded as follows

$$\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g\to g}(\vec{r})\phi_{l,g}(\vec{r}) = \sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g\to g}(\vec{r})\sum_{m=1}^{N/2}w_m P_l(\mu_m)\psi^E_{m,g}(\vec{r})\,. \tag{4.45}$$

The term $\psi^E_{m,g}(\vec{r})$ is consistently removed from the in-group portion of the scattering kernel and from the collision term on the left-hand side of the EP- SS$_N$ equations.

$$-\vec{\nabla}\cdot\left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})}\vec{\nabla}\psi_{m,g}^{E}(\vec{r})\right)+\sigma_{m,g}^{R}(\vec{r})\psi_{m,g}^{E}(\vec{r})=\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g\to g}(\vec{r})\sum_{\substack{m'=1\\ m'\neq m}}^{N/2}w_{m'}P_l(\mu_{m'})\psi_{m',g}^{E}(\vec{r})$$

$$+\sum_{\substack{g'=1\\ g'\neq g}}^{G}\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g}(\vec{r})\phi_{l,g'}(\vec{r})-\vec{\nabla}\cdot\left(\frac{\mu_m}{\sigma_{t,g}(\vec{r})}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g}(\vec{r})\vec{\phi}_{l,g'}(\vec{r})\right)$$

$$+\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)S_{l,g}(\vec{r})-\vec{\nabla}\cdot\left(\frac{\mu_m}{\sigma_{t,g}(\vec{r})}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\vec{S}_{l,g'}(\vec{r})\right)+q_{f,g}(\vec{r})$$

$$\text{for } m=1,\ N/2 \text{ and } g=1,\ G. \tag{4.46}$$

Note that in Eq. 4.46, the total cross-section is replaced, in the collision term, with a direction-dependent removal cross-section as follows

$$\sigma_{m,g}^{R}(\vec{r})=\sigma_{t,g}(\vec{r})-\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)^2\,w_m\sigma_{sl,g\to g}(\vec{r})\,. \tag{4.47}$$

In this new formulation, the main idea is to remove a "degree of freedom" from the iteration process in order to reduce the iterations on the component $\psi_{m,g}^{E}(\vec{r})$. This modification leads to a drastic reduction of the spectral radius.

4.3 Comparison of the P_1 Spherical Harmonics and SP_1 Equations

In order to understand the assumptions on which the SP_N and the EP-SS_N equations are based, it is useful to examine the 3-D P_1 spherical harmonics equations.[6] The expansion in spherical harmonics of the angular flux can be written as follows

$$\psi\left(\vec{r},\hat{\Omega}\right)=\sum_{l=0}^{N}\sum_{m=0}^{l}(2l+1)P_l^m(\cos\vartheta)\left[\psi_{lm}(\vec{r})\cos(m\varphi)+\gamma_{lm}(\vec{r})\sin(m\varphi)\right], \tag{4.48}$$

where

$$0<\vartheta<\pi \ \text{, and}\ 0<\varphi<2\pi\,.$$

In the following discussion, for simplicity I will assume a P_1 expansion of the angular flux in spherical harmonics, given by

$$\psi\left(\vec{r},\hat{\Omega}\right)= P_0^0(\cos\vartheta)\psi_{00}(\vec{r})+3P_1^0(\cos\vartheta)\psi_{10}(\vec{r})+3P_1^1(\cos\vartheta)\left[\psi_{11}(\vec{r})\cos\varphi+\gamma_{11}(\vec{r})\sin\varphi\right].$$

(4.49)

By substituting the definitions of the Associated Legendre polynomials[6] in Eq. 4.49, I obtain the P_1 expansion for the angular flux:

$$\psi\left(\vec{r},\hat{\Omega}\right)= \psi_{00}(\vec{r})+3\mu\psi_{10}(\vec{r})-3\sin\vartheta\left[\psi_{11}(\vec{r})\cos\varphi+\gamma_{11}(\vec{r})\sin\varphi\right], \tag{4.50}$$

where

$$\mu=\cos\vartheta .$$

The derivation of the SP_N equations outlined by Gelbard, assumes implicitly that the angular flux be azimuthally independent, and hence symmetric with respect to the azimuthal variable. By introducing this assumption on the P_1 expansion of the angular flux in Eq. 4.50, I obtain

$$\tilde{\psi}\left(\vec{r},\mu\right)=\int_0^{2\pi}\psi\left(\vec{r},\hat{\Omega}\right)d\varphi=\psi_{00}(\vec{r})+3\mu\psi_{10}(\vec{r})-3\sin\vartheta\int_0^{2\pi}\left[\psi_{11}(\vec{r})\cos\varphi+\gamma_{11}(\vec{r})\sin\varphi\right]d\varphi .$$

(4.51)

Therefore, by performing the integration on Eq. 4.51, I obtain

$$\tilde{\psi}\left(\vec{r},\mu\right)=\psi_{00}(\vec{r})+3\mu\psi_{10}(\vec{r}) . \tag{4.52}$$

It is evident that the angular flux obtained in Eq. 4.52 is equivalent to the SP_1 angular flux where, ψ_{00} is the scalar flux and ψ_{10} is the total current.

The general formulation of the multigroup P_N equations[6], with anisotropic scattering and source, is obtained by substituting Eq. 4.48 into the linear Boltzmann equation and deriving a set of coupled partial differential equations for the moments $\psi_{lm}^{g}(\vec{r})$ and $\gamma_{lm}^{g}(\vec{r})$.

$$2(l+m+1)\frac{\partial \psi^g_{l+1m}}{\partial z}+2(l-m)\frac{\partial \psi^g_{l-1m}}{\partial z}+\frac{\partial \psi^g_{l-1m-1}}{\partial x}-\frac{\partial \gamma^g_{l-1m-1}}{\partial y}-\frac{\partial \psi^g_{l+1m-1}}{\partial x}+\frac{\partial \gamma^g_{l+1m-1}}{\partial y}$$
$$+(l+m+2)(l+m+1)\left(\frac{\partial \psi^g_{l+1m+1}}{\partial x}+\frac{\partial \gamma^g_{l+1m+1}}{\partial y}\right)-(l-m-1)(l-m)\left(\frac{\partial \psi^g_{l-1m+1}}{\partial x}+\frac{\partial \gamma^g_{l-1m+1}}{\partial y}\right)$$
$$+2(2l+1)\sigma_{l,g}\psi^g_{lm}=2S_{lm,g},$$

(4.53a)

$$2(l+m+1)\frac{\partial \gamma^g_{l+1m}}{\partial z}+2(l-m)\frac{\partial \gamma^g_{l-1m}}{\partial z}+\frac{\partial \psi^g_{l-1m-1}}{\partial y}+\frac{\partial \gamma^g_{l-1m-1}}{\partial x}-\frac{\partial \psi^g_{l+1m-1}}{\partial y}-\frac{\partial \gamma^g_{l+1m-1}}{\partial x}$$
$$+(l+m+2)(l+m+1)\left(-\frac{\partial \psi^g_{l+1m+1}}{\partial y}+\frac{\partial \gamma^g_{l+1m+1}}{\partial x}\right)-(l-m-1)(l-m)\left(-\frac{\partial \psi^g_{l-1m+1}}{\partial y}+\frac{\partial \gamma^g_{l-1m+1}}{\partial x}\right)$$
$$+2(2l+1)\sigma_{l,g}\gamma^g_{lm}=2S'_{lm,g},$$

(4.53b)

for g=1, G,

where

$$\sigma_{l,g}=\sigma_{t,g}-\sigma_{sl,g\to g}.$$

Therefore, the P_1 equations are obtained by evaluating Eqs. 4.53a and 4.53b for l=0, 1 and m=0, 1, as follows

(l=0, m=0)

$$2\frac{\partial \psi^g_{10}}{\partial z}+2\left(\frac{\partial \psi^g_{11}}{\partial x}+\frac{\partial \gamma^g_{11}}{\partial y}\right)+2\sigma_{0,g}\psi^g_{00}=2S_{00,g},\tag{4.54a}$$

$$2\frac{\partial \gamma^g_{10}}{\partial z}+2\left(-\frac{\partial \psi^g_{11}}{\partial y}+\frac{\partial \gamma^g_{11}}{\partial x}\right)+2\sigma_{0,g}\gamma^g_{00}=2S'_{00,g}.\tag{4.54b}$$

(l=1, m=0)

$$4\frac{\partial \psi^g_{20}}{\partial z}+2\frac{\partial \psi^g_{00}}{\partial z}+6\left(\frac{\partial \psi^g_{21}}{\partial x}+\frac{\partial \gamma^g_{21}}{\partial y}\right)+6\sigma_{1,g}\psi^g_{10}=2S_{10,g},\tag{4.54c}$$

$$4\frac{\partial \gamma_{20}^{g}}{\partial z}+2\frac{\partial \gamma_{00}^{g}}{\partial z}+6\left(-\frac{\partial \psi_{21}^{g}}{\partial y}+\frac{\partial \gamma_{21}^{g}}{\partial x}\right)+6\sigma_{1,g}\gamma_{10}^{g}=2S'_{10,g}\,. \tag{4.54d}$$

(l=1, *m*=1*)*

$$6\frac{\partial \psi_{21}^{g}}{\partial z}+2\left(\frac{\partial \psi_{00}^{g}}{\partial x}-\frac{\partial \gamma_{00}^{g}}{\partial y}\right)-\frac{\partial \psi_{20}^{g}}{\partial x}+\frac{\partial \gamma_{20}^{g}}{\partial y}+12\left(\frac{\partial \psi_{22}^{g}}{\partial x}+\frac{\partial \gamma_{22}^{g}}{\partial y}\right)+6\sigma_{1,g}\psi_{11}^{g}=2S_{11,g}\,, \tag{4.54e}$$

$$6\frac{\partial \gamma_{21}^{g}}{\partial z}+2\left(\frac{\partial \psi_{00}^{g}}{\partial y}+\frac{\partial \gamma_{00}^{g}}{\partial x}\right)-\frac{\partial \psi_{20}^{g}}{\partial y}-\frac{\partial \gamma_{20}^{g}}{\partial x}+12\left(-\frac{\partial \psi_{22}^{g}}{\partial y}+\frac{\partial \gamma_{22}^{g}}{\partial x}\right)+6\sigma_{1,g}\gamma_{11}^{g}=2S'_{11,g}\,. \tag{4.54f}$$

The terms with *l*>1 and *m*>1 are dropped from Eqs. 4.54c through f, yielding the following relationships

$$\psi_{10}^{g}=-\frac{1}{3\sigma_{1,g}}\frac{\partial \psi_{00}^{g}}{\partial z}+\frac{S_{10,g}}{3\sigma_{1,g}}, \tag{4.55a}$$

$$\gamma_{10}^{g}=-\frac{1}{3\sigma_{1,g}}\frac{\partial \gamma_{00}^{g}}{\partial z}+\frac{S'_{10,g}}{3\sigma_{1,g}}, \tag{4.55b}$$

$$\psi_{11}^{g}=-\frac{1}{3\sigma_{1,g}}\left(\frac{\partial \psi_{00}^{g}}{\partial x}-\frac{\partial \gamma_{00}^{g}}{\partial y}\right)+\frac{S_{11,g}}{3\sigma_{1,g}}, \tag{4.55c}$$

$$\gamma_{11}^{g}=-\frac{1}{3\sigma_{1,g}}\left(\frac{\partial \psi_{00}^{g}}{\partial y}+\frac{\partial \gamma_{00}^{g}}{\partial x}\right)+\frac{S'_{11,g}}{3\sigma_{1,g}}\,. \tag{4.55d}$$

Then, by substituting Eqs. 4.55a, c and d in Eq. 4.54a, I obtain

$$-\vec{\nabla}\cdot\frac{1}{3\sigma_{1,g}}\vec{\nabla}\psi_{00}^{g}+\frac{\partial}{\partial x}\frac{1}{3\sigma_{1,g}}\frac{\partial \gamma_{00}^{g}}{\partial y}-\frac{\partial}{\partial y}\frac{1}{3\sigma_{1,g}}\frac{\partial \gamma_{00}^{g}}{\partial x}+\sigma_{0,g}\psi_{00}^{g}=S_{00,g}+\widetilde{S}_{1,g}\,, \tag{4.56}$$

where

$$\widetilde{S}_{1,g}=-\frac{\partial}{\partial z}\left(\frac{S_{10,g}}{3\sigma_{1,g}}\right)-\frac{\partial}{\partial x}\left(\frac{S_{11,g}}{3\sigma_{1,g}}\right)-\frac{\partial}{\partial y}\left(\frac{S'_{11,g}}{3\sigma_{1,g}}\right). \tag{4.56a}$$

Analogously, by using Eqs. 4.55b, c and d in Eq. 4.54b, I obtain

$$-\vec{\nabla}\cdot\frac{1}{3\sigma_{1,g}}\vec{\nabla}\gamma_{00}^{g}-\frac{\partial}{\partial x}\frac{1}{3\sigma_{1,g}}\frac{\partial\psi_{00}^{g}}{\partial y}+\frac{\partial}{\partial y}\frac{1}{3\sigma_{1,g}}\frac{\partial\psi_{00}^{g}}{\partial x}+\sigma_{0,g}\gamma_{00}^{g}=S'_{00,g}+\widetilde{S}'_{1,g}, \tag{4.57}$$

where

$$\widetilde{S}'_{1,g}=-\frac{\partial}{\partial z}\left(\frac{S'_{10,g}}{3\sigma_{1,g}}\right)-\frac{\partial}{\partial x}\left(\frac{S'_{11,g}}{3\sigma_{1,g}}\right)+\frac{\partial}{\partial y}\left(\frac{S_{11,g}}{3\sigma_{1,g}}\right). \tag{4.57a}$$

Eqs. 4.56 and 4.57 constitute a coupled system of partial differential equations for ψ_{00}^{g} and γ_{00}^{g}, which must be solved iteratively. Recall that the assumption made in the SP_N methodology is that the angular flux is azimuthally symmetric; therefore, to obtain the SP_1 equations (Eq. 4.58 or 4.59), terms such as γ_{00}^{g} are dropped from Eqs. 4.56 and 4.57, as follows

$$-\vec{\nabla}\cdot\frac{1}{3\sigma_{1,g}}\vec{\nabla}\psi_{00}^{g}+\sigma_{0,g}\psi_{00}^{g}=S_{00,g}-\vec{\nabla}\cdot\left(\frac{\vec{S}_{1,g}}{3\sigma_{1,g}}\right), \tag{4.58}$$

or

$$-\vec{\nabla}\cdot\frac{1}{3\sigma_{1,g}}\vec{\nabla}\phi_{0}^{g}+\sigma_{0,g}\phi_{0}^{g}=S_{0,g}-\vec{\nabla}\cdot\left(\frac{\vec{S}_{1,g}}{3\sigma_{1,g}}\right). \tag{4.59}$$

Here, I can also conclude that in the case of a homogeneous medium, with isotropic scattering, the P_1 and the SP_1 equations yield the same solution, because the azimuthal dependency on the angular flux is removed. Note that this result can also be generalized to the SP_N equations.

CHAPTER 5
NUMERICAL METHODS FOR SOLVING THE EP-SS_N EQUATIONS

This chapter addresses the numerical techniques utilized to solve the EP-SS_N equations; I will describe the discretization of the EP-SS_N equations in a 3-D Cartesian geometry using the finite-volume method, along with the matrix operator formulation utilized and the boundary conditions. I will also introduce the Compressed Diagonal Storage (CDS) method, which is fundamental for reducing the memory requirements and the computational complexity of the iterative solvers. Further, a new coarse mesh based projection algorithm for elliptic-type partial differential equations will be presented.

Finally, I will describe a class of iterative solvers based on the Krylov subspace methods, such as the Conjugate Gradient (CG) and the Bi-Conjugate Gradient methods (Bi-CG). The CG and Bi-CG methods have been implemented to solve the linear systems of equations arising from the finite-volume discretization of the EP-SS_N equations. Furthermore, the issue of preconditioning of the CG methodology will be discussed.

5.1 Discretization of the EP-SS_N Equations Using the Finite-Volume Method

The EP-SS_N equations derived in Chapter 4 are discretized using the finite-volume approach. For this purpose, I consider a general volume V in a 3-D Cartesian geometry. The volume V is then partitioned into non-overlapping sub-domains V_j, called *coarse meshes*. Note that, the coarse mesh sub-domains are generally defined along the boundaries of material regions. As I will discuss in Chapter 7, the main purpose of this approach is to partition the problem for parallel processing.

The discretization of the spatial domain is completed by defining a fine-mesh grid onto each coarse mesh. I have derived a formulation of the discretized EP-SS_N equations which allows for variable fine mesh density on different regions of the problem; this approach is very effective to generate an effective mesh distribution, because it allows a finer refinement only in those regions where higher accuracy is needed.

The finite-volume discretization of the multigroup EP-SS_N equations (Eqs. 4.29) is obtained by performing a triple integration on a finite volume, $dr \equiv dxdydz$, as follows

$$\iiint_v \left[-\vec{\nabla} \cdot \frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \vec{\nabla} \cdot + \sigma_{t,g}(\vec{r}) \right] \psi^E_{m,g}(\vec{r}) dr = \iiint_v \left[Q_{s,g,m}(\vec{r}) + Q_{ext,g,m}(\vec{r}) + Q_{f,g}(\vec{r}) \right] dr \,, \quad (5.1)$$

where

$$\begin{aligned} Q_{s,g,m}(\vec{r}) = & \sum_{g'=1}^{G} \sum_{\substack{l=0,2.. \\ even}}^{L-1} (2l+1) P_l(\mu_m) \sigma_{sl,g' \to g}(\vec{r}) \phi_{l,g'}(\vec{r}) \\ & - \vec{\nabla} \cdot \frac{\mu_m}{\sigma_{t,g}(\vec{r})} \sum_{g'=1}^{G} \sum_{\substack{l=1,3.. \\ odd}}^{L} (2l+1) P_l(\mu_m) \sigma_{sl,g' \to g}(\vec{r}) \vec{\phi}_{l,g'}(\vec{r}), \end{aligned} \quad (5.1a)$$

$$Q_{ext,g,m}(\vec{r}) = \sum_{\substack{l=0,2.. \\ even}}^{L-1} (2l+1) P_l(\mu_m) S_{l,g}(\vec{r}) - \vec{\nabla} \cdot \frac{\mu_m}{\sigma_{t,g}(\vec{r})} \sum_{\substack{l=1,3.. \\ odd}}^{L} (2l+1) P_l(\mu_m) \vec{S}_{l,g'}(\vec{r}) \,, \quad (5.1b)$$

and

$$Q_{f,g}(\vec{r}) = \frac{1}{k} \nu \sigma_{f,g}(\vec{r}) \phi_0(\vec{r}) \,. \quad (5.1c)$$

For this derivation, I consider a central finite-difference scheme for generic mesh element with coordinates x_i, y_j and z_k; an example of a fine mesh element and its neighbor points is shown in Figure 5.1.

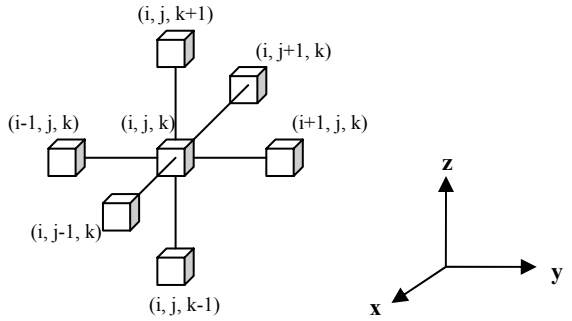

Figure 5.1. Fine mesh representation on a 3-D Cartesian grid.

The generic fine mesh element is defined by the discretization step sizes, Δx_c, Δy_c, and Δz_c, along the *x*-, *y*- and *z*-axis, respectively. Note that the discretization steps are constant within each coarse mesh; hence, a non-uniform mesh distribution is not allowed. The discretization steps are defined as follows

$$\Delta x_c = \frac{L_x^c}{N_x^c},\ \Delta y_c = \frac{L_y^c}{N_y^c},\ \Delta z_c = \frac{L_z^c}{N_z^c}, \text{ and } \Delta v_c = \Delta x_c \Delta y_c \Delta z_c, \qquad (5.2)$$

$$\text{for } c=1,\ N_{cm}$$

where, N_{cm} is the total number of coarse meshes; L_x^c, L_y^c, and L_z^c are the dimensions of the coarse mesh (c), along the *x*-, *y*- and *z*-axis, respectively; and N_x^c, N_y^c, and N_z^c refer to the number of fine meshes along the *x*-, *y*- and *z*-axis, respectively. Note that, Eq. 5.1 is numerically integrated on a generic finite volume Δv_c.

I will first consider the integration of the elliptic or leakage operator (first term in Eq. 5.1) as follows

$$
\begin{aligned}
&-\int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy \int_{z_{k-1/2}}^{z_{k+1/2}} dz \left(\vec{\nabla} \cdot \frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \vec{\nabla} \psi_{m,g}^E(\vec{r}) \right) = \\
&-\Delta y_c \Delta z_c \left[\left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial x} \psi_{m,g}^E(\vec{r}) \right)_{x_{i+1/2}} - \left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial x} \psi_{m,g}^E(\vec{r}) \right)_{x_{i-1/2}} \right] \\
&-\Delta x_c \Delta z_c \left[\left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial y} \psi_{m,g}^E(\vec{r}) \right)_{y_{j+1/2}} - \left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial y} \psi_{m,g}^E(\vec{r}) \right)_{y_{j-1/2}} \right] \\
&-\Delta x_c \Delta y_c \left[\left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial z} \psi_{m,g}^E(\vec{r}) \right)_{z_{k+1/2}} - \left(\frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \frac{\partial}{\partial z} \psi_{m,g}^E(\vec{r}) \right)_{z_{k-1/2}} \right].
\end{aligned}
\tag{5.3}
$$

For simplicity, I will derive the discretized operator along the *x*-axis; the treatment is analogous along the *y*- and *z*-axis. Figure 5.2 represents the view of a fine mesh and its neighbor points along the *x*-axis.

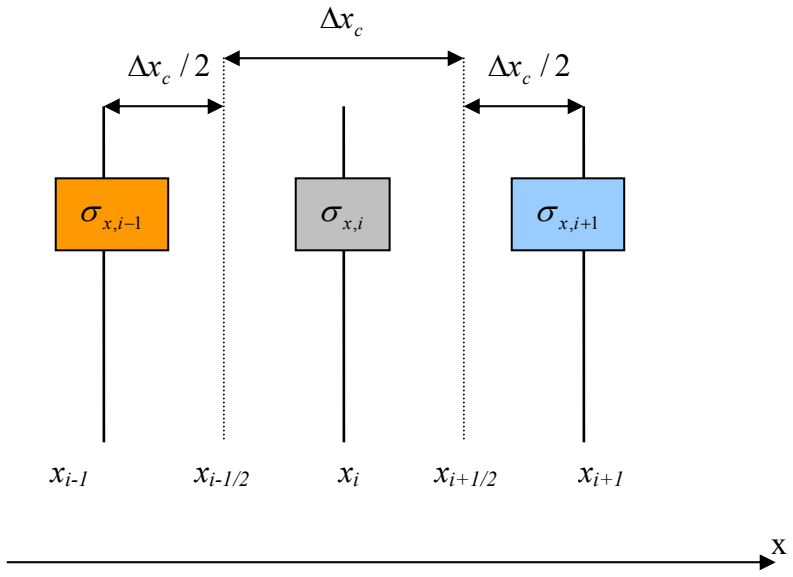

Figure 5.2. View of a fine mesh along the *x*-axis.

In Figure 5.2, σ_x represents a generic macroscopic cross-section (e.g., total, fission, etc.) which is constant within the fine mesh. In Eq. 5.3, I evaluate the right-side and left-side partial derivatives along the *x*-axis at $x_{i+1/2}$.

$$f_{m,g}^{E(-)}(x_{i+1/2},y_j,z_k)=\frac{\mu_m^2}{\sigma_{t,g}(x_i,y_j,z_k)}\frac{\psi_{m,g}^{E(-)}(x_{i+1/2},y_j,z_k)-\psi_{m,g}^{E}(x_i,y_j,z_k)}{\Delta x_c/2}. \tag{5.4}$$

$$f_{m,g}^{E(+)}(x_{i+1/2},y_j,z_k)=\frac{\mu_m^2}{\sigma_{t,g}(x_{i+1},y_j,z_k)}\frac{\psi_{m,g}^{E}(x_{i+1},y_j,z_k)-\psi_{m,g}^{E(+)}(x_{i+1/2},y_j,z_k)}{\Delta x_c/2}. \tag{5.5}$$

In order for the elliptic operator to be defined, the function $\psi_{m,g}^{E}(x,y,z)$ must be continuous along with its first derivative $f_{m,g}^{E}(x,y,z)$ and second derivative, which translates into the fact that the even-parity angular flux belongs to a C^2 functional space, or $\psi_{m,g}^{E}(x,y,z)\in C^2$. Therefore, the following relationships hold true

$$\psi_{m,g}^{E}(x_{i+1/2},y_j,z_k)\equiv\psi_{m,g}^{E(-)}(x_{i+1/2},y_j,z_k)=\psi_{m,g}^{E(+)}(x_{i+1/2},y_j,z_k), \tag{5.6}$$

and

$$f_{m,g}^{E}(x_{i+1/2},y_j,z_k)\equiv f_{m,g}^{E(-)}(x_{i+1/2},y_j,z_k)=f_{m,g}^{E(+)}(x_{i+1/2},y_j,z_k). \tag{5.7}$$

Therefore, I eliminate the value of $\psi_{m,g}^{E}(x_{i+1/2},y,z)$ in Eqs. 5.4, obtaining the second order, central-finite differencing formula for the even-parity angular flux:

$$\psi_{i+1/2,i,j,m,g}^{E}=\frac{d_{i+1,j,k,m,g}^{x}\psi_{i+1,j,k,m,g}^{E}+d_{i,j,k,m,g}^{x}\psi_{i,j,k,m,g}^{E}}{d_{i+1,j,k,m,g}+d_{i,j,k,m,g}}, \tag{5.8}$$

and the even-parity current density

$$f_{i+1/2,j,k,m,g}^{E}=\frac{2d_{i,j,k,m,g}^{x}d_{i+1,j,k,m,g}^{x}}{d_{i,j,k,m,g}^{x}+d_{i+1,j,k,m,g}^{x}}\left(\psi_{i+1,j,k,m,g}^{E}-\psi_{i,j,k,m,g}^{E}\right). \tag{5.9}$$

In Eqs. 5.8 and 5.9, I have defined the pseudo-diffusion coefficients along the *x*-axis, as

$$d_{i,j,k,m,g}^{x}=\frac{\mu_m^2}{\sigma_{t,i,j,k,g}\Delta x_c},\ d_{i+1,j,k,m,g}^{x}=\frac{\mu_m^2}{\sigma_{t,i+1,j,k,g}\Delta x_c},\ d_{i-1,j,k,m,g}^{x}=\frac{\mu_m^2}{\sigma_{t,i-1,j,k,g}\Delta x_c}. \tag{5.10}$$

Analogously, the expression for $f^E(x_{i-1/2}, y_j, z_k)$ is obtained as follows

$$f^E_{i-1/2,j,k,m,g} = \frac{2d^x_{i,j,k,m,g} d^x_{i-1,j,k,m,g}}{d^x_{i,j,k,m,g} + d^x_{i-1,j,k,m,g}} \left(\psi^E_{i,j,k,m,g} - \psi^E_{i-1,j,k,m,g} \right). \tag{5.11}$$

The partial derivatives along the *y*- and *z*-axis are discretized in a similar fashion, yielding the finite-volume discretized elliptic operator given by

$$\begin{aligned} &-\int_{x_{i-1/2}}^{x_{i+1/2}} dx \int_{y_{j-1/2}}^{y_{j+1/2}} dy \int_{z_{k-1/2}}^{z_{k+1/2}} dz \left(\vec{\nabla} \cdot \frac{\mu_m^2}{\sigma_{t,g}(\vec{r})} \vec{\nabla} \psi^E_{m,g}(\vec{r}) \right) = \\ &-\Delta y_c \Delta z_c \left[\alpha_{i,i+1,m,g} \left(\psi^E_{i+1,j,k,m,g} - \psi^E_{i,j,k,m,g} \right) - \alpha_{i,i-1,m,g} \left(\psi^E_{i,j,k,m,g} - \psi^E_{i-1,j,k,m,g} \right) \right] \\ &-\Delta x_c \Delta z_c \left[\beta_{j,j+1,m,g} \left(\psi^E_{i,j+1,k,m,g} - \psi^E_{i,j,k,m,g} \right) - \beta_{j,j-1,m,g} \left(\psi^E_{i,j,k,m,g} - \psi^E_{i,j-1,k,m,g} \right) \right] \\ &-\Delta x_c \Delta y_c \left[\gamma_{k,k+1,m,g} \left(\psi^E_{i,j,k+1,m,g} - \psi^E_{i,j,k,m,g} \right) - \gamma_{k,k-1,m,g} \left(\psi^E_{i,j,k,m,g} - \psi^E_{i,j,k-1,m,g} \right) \right], \end{aligned} \tag{5.12}$$

where

$$\alpha_{i,i+1,m,g} = \frac{2d^x_{i,j,k,m,g} d^x_{i+1,j,k,m,g}}{d^x_{i,j,k,m,g} + d^x_{i+1,j,k,m,g}}, \alpha_{i,i-1,m,g} = \frac{2d^x_{i,j,k,m,g} d^x_{i-1,j,k,m,g}}{d^x_{i,j,k,m,g} + d^x_{i-1,j,k,m,g}}, \tag{5.13a}$$

$$\beta_{j,j+1,m,g} = \frac{2d^y_{i,j,k,m,g} d^y_{i,j+1,k,m,g}}{d^y_{i,j,k,m,g} + d^y_{i,j+1,k,m,g}}, \beta_{j,j-1,m,g} = \frac{2d^y_{i,j,k,m,g} d^y_{i,j-1,k,m,g}}{d^y_{i,j,k,m,g} + d^y_{i,j-1,k,m,g}}, \tag{5.13b}$$

$$\gamma_{k,k+1,m,g} = \frac{2d^z_{i,j,k,m,g} d^z_{i,j,k+1,m,g}}{d^z_{i,j,k,m,g} + d^z_{i,j,k+1,m,g}}, \gamma_{k,k-1,m,g} = \frac{2d^z_{i,j,k,m,g} d^z_{i,j,k-1,m,g}}{d^z_{i,j,k,m,g} + d^z_{i,j,k-1,m,g}}, \tag{5.13c}$$

and

$$d^y_{i,j,k,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j,k,g} \Delta y_c}, d^y_{i,j+1,k,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j+1,k,g} \Delta y_c}, \; d^y_{i,j-1,k,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j-1,k,g} \Delta y_c}, \tag{5.14}$$

$$d^z_{i,j,k,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j,k,g} \Delta z_c}, d^z_{i,j,k+1,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j,k+1,g} \Delta z_c}, \; d^z_{i,j,k-1,m,g} = \frac{\mu_m^2}{\sigma_{t,i,j,k-1,g} \Delta z_c}. \tag{5.15}$$

Finally, by integrating the remaining terms of the EP-SS_N equations, I obtain the complete multigroup EP-SS_N formulation with anisotropic scattering and source as follows

$$
\begin{aligned}
&-\Delta y_c \Delta z_c \left[\alpha_{i,i+1,m,g}\left(\psi^E_{i+1,j,k,m,g}-\psi^E_{i,j,k,m,g}\right)-\alpha_{i,i-1,m,g}\left(\psi^E_{i,j,k,m,g}-\psi^E_{i-1,j,k,m,g}\right)\right]\\
&-\Delta x_c \Delta z_c \left[\beta_{j,j+1,m,g}\left(\psi^E_{i,j+1,k,m,g}-\psi^E_{i,j,k,m,g}\right)-\beta_{j,j-1,m,g}\left(\psi^E_{i,j,k,m,g}-\psi^E_{i,j-1,k,m,g}\right)\right]\\
&-\Delta x_c \Delta y_c \left[\gamma_{k,k+1,m,g}\left(\psi^E_{i,j,k+1,m,g}-\psi^E_{i,j,k,m,g}\right)-\gamma_{k,k-1,m,g}\left(\psi^E_{i,j,k,m,g}-\psi^E_{i,j,k-1,m,g}\right)\right]+\\
&+\sigma_{t,g,i,j,k}\psi^E_{i,j,k,m,g}\Delta v_c=\sum_{g'=1}^{G}\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\phi_{l,g',i,j,k}\Delta v_c-\\
&-\Delta y_c\Delta z_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(\phi_{l,g',i+1/2,j,k}-\phi_{l,g',i-1/2,j,k}\right)-\\
&-\Delta x_c\Delta z_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(\phi_{l,g',i,j+1/2,k}-\phi_{l,g',i,j-1/2,k}\right)-\\
&-\Delta x_c\Delta y_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(\phi_{l,g',i,j,k+1/2}-\phi_{l,g',i,j,k-1/2}\right)+\\
&+\sum_{\substack{l=0,2..\\ even}}^{L-1}(2l+1)P_l(\mu_m)S_{l,g,i,j,k}\Delta v_c+\\
&-\Delta y_c\Delta z_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(S_{l,g',i+1/2,j,k}-S_{l,g',i-1/2,j,k}\right)-\\
&-\Delta x_c\Delta z_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(S_{l,g',i,j+1/2,k}-S_{l,g',i,j-1/2,k}\right)-\\
&-\Delta x_c\Delta y_c\frac{\mu_m}{\sigma_{t,g,i,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\sigma_{sl,g'\to g,i,j,k}\left(S_{l,g',i,j,k+1/2}-S_{l,g',i,j,k-1/2}\right)+\\
&Q_{f,g,i,j,k}\Delta v_c,
\end{aligned}
\qquad (5.16)
$$

for $c=1, N_{cm}$, $m=1, N/2$, $L=0, N-1$, $g=1, G$.

The EP-SS_N equations discretized with the finite-volume method can be expressed in a matrix operator form characterized by a 7-diagonal banded structure.

$$A_{c,m,g}=\begin{bmatrix} D_{m,g} & U^{x}_{m,g} & & U^{y}_{m,g} & & U^{z}_{m,g} & \\ L^{x}_{m,g} & D_{m,g} & U^{x}_{m,g} & & U^{y}_{m,g} & & U^{z}_{m,g} \\ & L^{x}_{m,g} & D_{m,g} & U^{x}_{m,g} & & U^{y}_{m,g} & \\ L^{y}_{m,g} & & L^{x}_{m,g} & D_{m,g} & U^{x}_{m,g} & & U^{y}_{m,g} \\ & L^{y}_{m,g} & & L^{x}_{m,g} & D_{m,g} & U^{x}_{m,g} & \\ L^{z}_{m,g} & & L^{y}_{m,g} & & L^{x}_{m,g} & D_{m,g} & U^{x}_{m,g} \\ & L^{z}_{m,g} & & L^{y}_{m,g} & & L^{x}_{m,g} & D_{m,g} \end{bmatrix},$$

for c=1, N_{cm}; m=1, N/2; g=1, G,

where

$$D^{x}_{m,g}=\Delta y_c\Delta z_c\left(\alpha_{i,i+1,m,g}+\alpha_{i,i-1,m,g}\right)+\Delta x_c\Delta z_c\left(\beta_{j,j+1,m,g}+\beta_{j,j-1,m,g}\right)+\Delta x_c\Delta y_c\left(\gamma_{k,k+1,m,g}+\gamma_{k,k-1,m,g}\right),$$

$$U^{x}_{m,g}=-\Delta y_c\Delta z_c\alpha_{i,i+1,m,g},\ L^{x}_{m,g}=-\Delta y_c\Delta z_c\alpha_{i,i-1,m,g},$$

$$U^{y}_{m,g}=-\Delta x_c\Delta z_c\beta_{j,j+1,m,g},\ L^{y}_{m,g}=-\Delta x_c\Delta z_c\beta_{j,j-1,m,g},$$

$$U^{z}_{m,g}=-\Delta x_c\Delta y_c\gamma_{k,k+1,m,g},\ L^{z}_{m,g}=-\Delta x_c\Delta y_c\gamma_{k,k-1,m,g},$$

for i=2, $N^{c}_{x}-1$, j=2, $N^{c}_{y}-1$, k=2, $N^{c}_{z}-1$, c=1, N_{cm}, m=1, N/2, g=1, G.

5.2 Numerical Treatment of the Boundary Conditions

The boundary conditions for the EP-SS_N equations are discretized as well using the finite-volume method. In general, the BCs can be prescribed at back ($-x_b$), front ($+x_b$), left ($-y_b$), right ($+y_b$), bottom ($-z_b$), and top ($+z_b$). The reflective boundary conditions are simply derived as follows:

$-x_b$) $-\psi^{O}_{m,g,1/2,j,k}=0$, $+x_b$) $\psi^{O}_{m,g,N_x+1/2,j,k}=0$, (5.17a)

$-y_b$) $-\psi^{O}_{m,g,i,1/2,k}=0$, $+y_b$) $\psi^{O}_{m,g,i,N_y+1/2,k}=0$, (5.17b)

$-z_b$) $-\psi^{O}_{m,g,i,j,1/2}=0$, $+z_b$) $\psi^{O}_{m,g,i,j,N_z+1/2}=0$. (5.17c)

The vacuum boundary conditions are obtained from Eq. 4.32, by setting $\alpha = 0$. Hence, the vacuum boundary conditions along the *x*-, *y*- and *z*-axis are given below:

Front side vacuum boundary condition x = $+x_b$

$$\psi^{O}_{N_x+1/2,j,k,m,g} = \frac{a_{N_x,m,g}}{1+a_{N_x,m,g}}\psi^{E}_{N_x,j,k,m,g} - \\ -\frac{a_{N_x,m,g}}{1+a_{N_x,m,g}}\frac{1}{\sigma_{t,g,N_x,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,N_x,j,k}\phi_{l,g',N_x+1/2,j,k} + S_{l,g',N_x+1/2,j,k}\right] \tag{5.18a}$$

Back side vacuum boundary condition x = $-x_b$

$$\psi^{O}_{1/2,j,k,m,g} = -\frac{a_{1,m,g}}{1+a_{1,m,g}}\psi^{E}_{1,j,k,m,g} + \\ +\frac{a_{1,m,g}}{1+a_{1,m,g}}\frac{1}{\sigma_{t,g,1,j,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,1,j,k}\phi_{l,g',3/2,j,k} + S_{l,g',3/2,j,k}\right] \tag{5.18b}$$

Right side vacuum boundary condition y = $+y_b$

$$\psi^{O}_{i,N_y+1/2,k,m,g} = \frac{b_{N_y,m,g}}{1+b_{N_y,m,g}}\psi^{E}_{i,N_y,k,m,g} - \\ -\frac{b_{N_y,m,g}}{1+b_{N_y,m,g}}\frac{1}{\sigma_{t,g,i,N_y,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,i,N_y,k}\phi_{l,g',i,N_y+1/2,k} + S_{l,g',i,N_y+1/2,k}\right] \tag{5.19a}$$

Left side vacuum boundary condition y = $-y_b$

$$\psi^{O}_{i,1/2,k,m,g} = -\frac{b_{1,m,g}}{1+b_{1,m,g}}\psi^{E}_{i,1,k,m,g} + \\ +\frac{b_{1,m,g}}{1+b_{1,m,g}}\frac{1}{\sigma_{t,g,i,1,k}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,i,1,k}\phi_{l,g',i,3/2,k} + S_{l,g',i,3/2,k}\right] \tag{5.19b}$$

Bottom side vacuum boundary condition $z = -z_b$

$$\psi^{O}_{i,j,N_z+1/2,m,g} = \frac{c_{N_z,m,g}}{1+c_{N_z,m,g}}\psi^{E}_{i,j,N_z,m,g} - \\ -\frac{c_{N_z,m,g}}{1+c_{N_z,m,g}}\frac{1}{\sigma_{t,g,i,j,N_z}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,i,j,N_z}\phi_{l,g',i,j,N_z+1/2} + S_{l,g',i,j,N_z+1/2}\right] \tag{5.20a}$$

Top side vacuum boundary condition $z = +z_b$

$$\psi^{O}_{i,j,1/2,m,g} = -\frac{c_{1,m,g}}{1+c_{1,m,g}}\psi^{E}_{i,j,1,m,g} + \\ +\frac{c_{1,m,g}}{1+c_{1,m,g}}\frac{1}{\sigma_{t,g,i,j,1}}\sum_{g'=1}^{G}\sum_{\substack{l=1,3..\\ odd}}^{L}(2l+1)P_l(\mu_m)\left[\sigma_{sl,g'\to g,i,j,1}\phi_{l,g',i,j,3/2} + S_{l,g',i,j,3/2}\right] \tag{5.20b}$$

where

$$a_{1,m,g} = \frac{2\mu_m}{\Delta x_c \sigma_{t,g,1,j,k}},\ a_{N_x,m,g} = \frac{2\mu_m}{\Delta x_c \sigma_{t,g,N_x,j,k}},$$

$$b_{1,m,g} = \frac{2\mu_m}{\Delta y_c \sigma_{t,g,i,1,k}},\ b_{N_y,m,g} = \frac{2\mu_m}{\Delta y_c \sigma_{t,g,i,N_y,k}},$$

$$c_{1,m,g} = \frac{2\mu_m}{\Delta z_c \sigma_{t,g,i,j,1}},\ c_{N_z,m,g} = \frac{2\mu_m}{\Delta z_c \sigma_{t,g,i,j,N_z}},$$

and

$$N_x = \sum_{c=1}^{N_{cm}} N_x^c\ ,\ N_y = \sum_{c=1}^{N_{cm}} N_y^c\ ,\ N_z = \sum_{c=1}^{N_{cm}} N_z^c\ .$$

5.3 The Compressed Diagonal Storage Method

Due to the sparse structure of the matrices involved, I have adopted the Compressed Diagonal Storage (CDS) method in order to efficiently store the matrix operators. The CDS method stores only the non-zero elements of the coefficient matrix and it uses an auxiliary vector to identify the column position of each element. Due to the

banded structure of the coefficients matrix, a mapping algorithm is easily defined for a generic square matrix as follows:

$$\begin{array}{lcl} a(i,j) \in A_{i,j} & & \widetilde{a}(i,d) \in \widetilde{A}_{i,d} \\ i = 1, I & \Rightarrow & i = 1, I \quad , \; jcol(i,d) . \\ j = 1, J & & d = -3,3 \end{array} \tag{5.21}$$

The algorithm defined in Eq. 5.21, maps the full structure of the matrix A into a compressed diagonal structure, where for each element on row *i*, there is an associated diagonal index ranging from -3 to 3, with index 0 being the main diagonal, and an auxiliary vector *jcol*, which stores the column position of each element. If we consider a 360x360 full matrix in single precision, with a total of 129600 elements, the memory required for allocating the matrix is roughly 2.1 MB. However, if the CDS method is used, the total number of non-zero elements to be stored is only 2520, for a total memory requirement of 42 KB, which is a reduction of a factor of 50 compared to the full matrix storage. Moreover, since the CDS method stores only non-zero elements, I have also obtained a reduction in the number of operations involved in the matrix-vector multiplication algorithms.

5.4 Coarse Mesh Interface Projection Algorithm

The partitioning of the spatial domain into non-overlapping coarse meshes leads to a situation in which the EP-SS_N equations have to be discretized independently for each coarse mesh. Therefore, each coarse mesh is considered as an independent transport problem; however, to obtain the solution on the whole domain, an interface projection algorithm has to be used in conjunction with an iterative method. The matrix operators have to be modified on the interfaces in order to couple the equations on each coarse

mesh. For explanatory purposes, consider Figure 5.3, which shows the interface region between two coarse meshes.

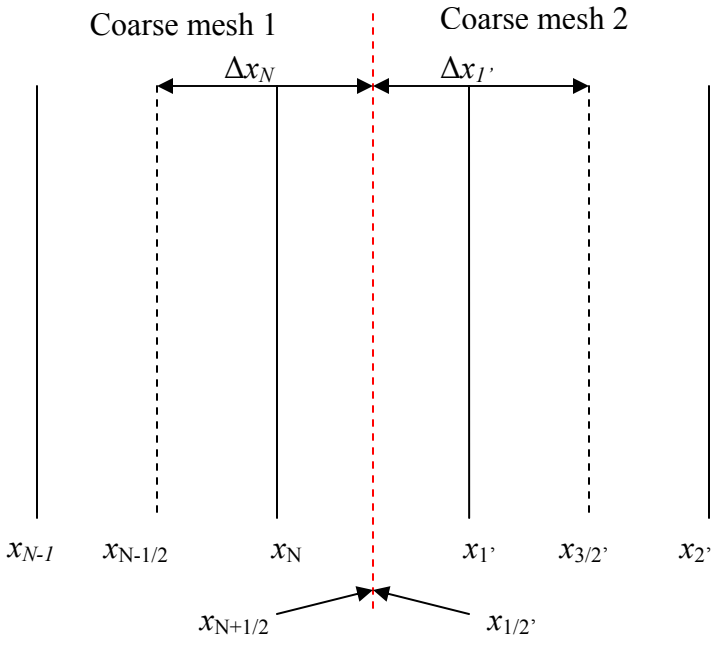

Figure 5.3. Representation of a coarse mesh interface

The coordinates $x_{N+1/2}$ and $x_{1/2'}$ represent the interface on coarse mesh 1 and 2, respectively. As shown in Figure 5.3, the discretization of the elliptic operator for coarse mesh 1, using the central finite difference method, would require the values of the even-parity angular flux at points x_{N-1}, x_N, and $x_{1'}$. Similarly, in coarse mesh 2, the discretization would involve the value of the EP angular flux at points x_N, $x_{1'}$, and $x_{2'}$. However, the point $x_{1'}$ is located on coarse mesh 2 and point x_N is located on coarse mesh 1; hence this term does not appear explicitly in the matrix operator for both coarse meshes.

In order to couple the equations on the interface, I have reformulated the discretized equations by bringing the unknown points on the right side of the equations. The numerical discretization of the EP-SS_N equations in coarse mesh 1 would yield

$$-\left[\alpha_{N,1',m,g}\left(\tilde{\psi}^{E}_{1',m,g}-\psi^{E}_{N,m,g}\right)-\alpha_{N,N-1,m,g}\left(\psi^{E}_{N,m,g}-\psi^{E}_{N-1,m,g}\right)\right]+$$
$$+\sigma_{t,g,N}\psi^{E}_{N,m,g}\Delta x_N=\Delta x_N Q_{s,g,m,N}+\Delta x_N Q_{ext,g,m,N}+\Delta x_N Q_{f,g,N},$$

(5.22)

where

$$\alpha_{N,1',m,g}=\frac{2d^{x}_{N,m,g}d^{x}_{1',m,g}}{d^{x}_{N,m,g}+d^{x}_{1',m,g}},\ \text{and}\ \alpha_{N,N-1,m,g}=\frac{2d^{x}_{N,m,g}d^{x}_{N-1,m,g}}{d^{x}_{N,m,g}+d^{x}_{N-1,m,g}}. \tag{5.23}$$

The coefficient $d^{x}_{1',m,g}$ depends on the material properties and fine mesh discretization of coarse mesh 2, and it is calculated a priori; however, in Eq. 5.22, the term $\tilde{\psi}^{E}_{1',m,g}$ is unknown, and hence has to be evaluated iteratively by placing it in the source term, as shown in Eq. 5.24.

$$\alpha_{N,1',m,g}\psi^{E}_{N,m,g}+\alpha_{N,N-1,m,g}\psi^{E}_{N,m,g}-\alpha_{N,N-1,m,g}\psi^{E}_{N-1,m,g}+\sigma_{t,g,N}\psi^{E}_{N,m,g}\Delta x_N=$$
$$\Delta x_N Q_{s,g,m,N}+\Delta x_N Q_{ext,g,m,N}+\Delta x_N Q_{f,g,N}+\alpha_{N,1',m,g}\tilde{\psi}^{E}_{1',m,g}.$$

(5.24)

A similar equation can be formulated for coarse mesh 2, as follows

$$-\left[\alpha_{1',2',m,g}\left(\psi^{E}_{2',m,g}-\psi^{E}_{1',m,g}\right)-\alpha_{1',N,m,g}\left(\psi^{E}_{1',m,g}-\tilde{\psi}^{E}_{N,m,g}\right)\right]$$
$$+\sigma_{t,g,1'}\psi^{E}_{1',m,g}\Delta x_{1'}=\Delta x_{1'}Q_{s,g,m,1'}+\Delta x_{1'}Q_{ext,g,m,1'}+\Delta x_{1'}Q_{f,g,1'}, \tag{5.25}$$

or

$$-\alpha_{1',2',m,g}\psi^{E}_{2',m,g}+\alpha_{1',2',m,g}\psi^{E}_{1',m,g}+\alpha_{1',N,m,g}\psi^{E}_{1',m,g}+\sigma_{t,g,1'}\psi^{E}_{1',m,g}\Delta x_{1'}=$$
$$\Delta x_{1'}Q_{s,g,m,1'}+\Delta x_{1'}Q_{ext,g,m,1'}+\Delta x_{1'}Q_{f,g,1'}+\alpha_{1',N,m,g}\tilde{\psi}^{E}_{N,m,g}, \tag{5.26}$$

where

$$\alpha_{1',2',m,g}=\frac{2d^{x}_{1',m,g}d^{x}_{2',m,g}}{d^{x}_{1',m,g}+d^{x}_{2',m,g}},\ \text{and}\ \alpha_{1',N,m,g}=\frac{2d^{x}_{1',m,g}d^{x}_{N,m,g}}{d^{x}_{1',m,g}+d^{x}_{N,m,g}}. \tag{5.27}$$

Therefore, Eq. 5.24 and 5.26 are coupled through the value of the EP angular fluxes $\widetilde{\psi}^{E}_{1',m,g}$ and $\widetilde{\psi}^{E}_{N,m,g}$. The EP-$SS_N$ equations are solved iteratively starting in coarse mesh 1, and assuming an initial guess for $\widetilde{\psi}^{E}_{1',m,g}$. Once the calculation is completed the value of $\widetilde{\psi}^{E}_{N,m,g}$ in Eq. 5.26, is set equal to $\psi^{E}_{N,m,g}$. Hence, once the calculation is completed on coarse mesh 2, the value obtained for $\psi^{E}_{1',m,g}$ is used in Eq. 5.24, to update the value of $\widetilde{\psi}^{E}_{1',m,g}$; this procedure continues until a convergence criterion is satisfied.

In a 3-D Cartesian geometry the coupling on the coarse mesh interfaces is achieved exactly as described above; however, in this case the coarse meshes can be discretized with different fine mesh grid densities. The variable grid density requires a projection algorithm in order to map the EP angular fluxes and the pseudo-diffusion coefficients among different grids. As stated earlier in this chapter, the variable density grid approach is very effective to refine only those regions of the model where a higher accuracy is needed; note that the main constraint on the fine mesh grid is the mesh size being smaller than the mean free path for that particular material region. The main philosophy behind the projection algorithm is derived from the multigrid method, where a prolongation/injection operator is used to map a vector onto grids with different discretizations.

Figure 5-4 shows the application of the projection algorithm along the *y*-axis on the interface between two coarse meshes.

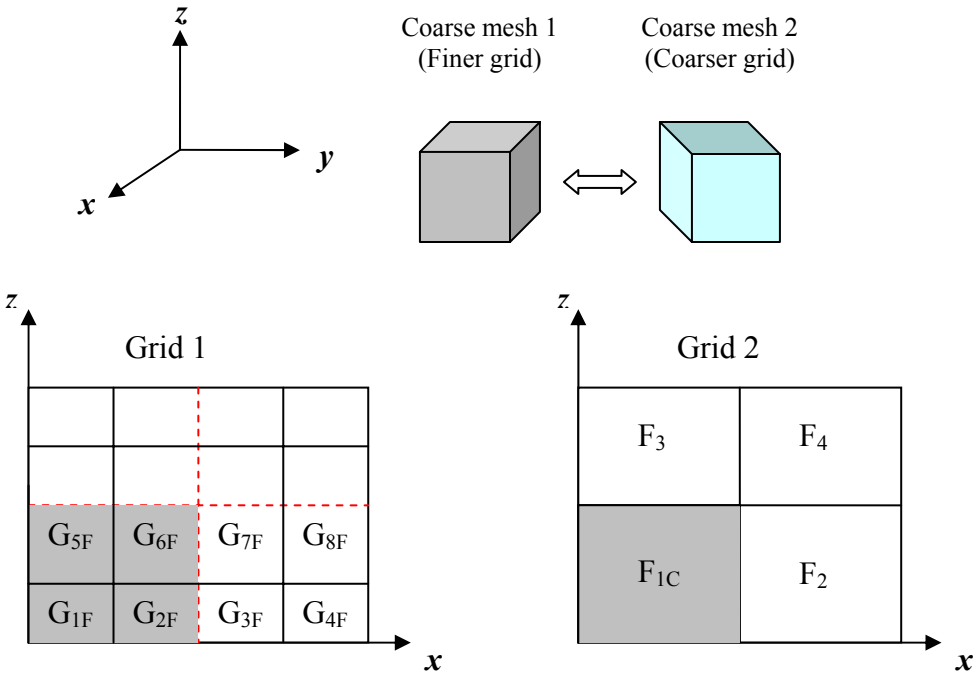

Figure 5.4. Representation of the interface projection algorithm between two coarse meshes.

For simplicity, I will consider the projection of a vector between two coarse meshes, along the y-axis, as shown in Figure 5.4. The fine-to-coarse projection of a vector is obtained by collapsing the values as follows

$$F_{1C} = \sum_{i=1}^{4} w_{iF} G_{iF}, \tag{5.28}$$

where

$$w_{iF} = \frac{A_{iF}}{A_C}, \text{ for } i=1, 4 \tag{5.29}$$

In Eq. 5.29, A_{iF} and A_C, are the areas associated with the fine-mesh and coarse-mesh grid, respectively. Conversely, the coarse-to-fine projection is obtained as follows

$$G_{1F} = w_{1F} F_{1C}, \tag{5.30a}$$

$$G_{2F} = w_{2F} F_{1C}, \tag{5.30b}$$

$$G_{3F} = w_{3F} F_{1C}, \tag{5.30c}$$

$$G_{4F} = w_{4F} F_{1C}. \tag{5.30d}$$

In general, the fine-to-coarse mesh projection is obtained with the following formulation

$$F_{iC} = \sum_{j=1}^{N_F} w_{jF} G_{jF}, \tag{5.31}$$

where

$$w_{jF} = \frac{A_{jF}}{A_{iC}}. \tag{5.32}$$

The weights in Eq. 5.32 are the ratios of the areas of the fine meshes intercepted by the coarse meshes on which the values are being mapped. Similarly, the coarse-to-fine mesh projection algorithm is defined as follows

$$G_{iF} = \sum_{j=1}^{N_C} w_{jC} F_{jC}, \tag{5.33}$$

where

$$w_{jC} = \frac{A_{iF}}{A_{jC}}. \tag{5.34}$$

By using the above formulations, the even-parity angular fluxes and the pseudo-diffusion coefficients are projected among coarse meshes with different grid densities. Note that the projected pseudo-diffusion coefficients need to be calculated only one time at the beginning of the calculation, while, the projections for the EP angular fluxes have to be updated at every iteration.

5.5 Krylov Subspace Iterative Solvers

Due to the size and sparse structure of the matrix operators obtained from the discretization of the EP-SS_N equations, direct solution methods such as LU

decomposition and Gaussian elimination do not perform effectively both in terms of computation time and memory requirements. In contrast, the Krylov subspace iterative methods, such as Conjugate Gradient (CG), are specifically designed to efficiently solve large linear systems of equations characterized by sparse matrix operators.

Note that in many engineering applications, the matrix operators resulting from a finite-difference discretization is usually positive-definite and diagonally dominant. These conditions are fundamental in ensuring the existence of a unique solution. A matrix is positive-definite if it satisfies the following condition

$$\vec{x}^T A\vec{x} > 0\,, \text{ for every vector } \vec{x} \neq 0\,. \tag{5.35}$$

Moreover, a matrix is defined to be diagonally dominant if the following condition holds true.

$$\left|a_{ii}\right| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} \left|a_{ij}\right|, \text{ for } i=1, n. \tag{5.36}$$

The CG algorithm is based on the fact that the solution of the linear system $A\vec{x} = \vec{b}$ is equivalent to finding the minimum of a quadratic form given by

$$f(\vec{x}) = \frac{1}{2}\vec{x}^T A\vec{x} - b^T\vec{x} + c\,. \tag{5.37}$$

The minimum of the quadratic form of Eq. 5.37 is evaluated by calculating its gradient as follows

$$f'(\vec{x}) = \begin{bmatrix} \dfrac{\partial}{\partial x_1} f(\vec{x}) \\ \vdots \\ \dfrac{\partial}{\partial x_n} f(\vec{x}) \end{bmatrix}. \tag{5.38}$$

The gradient of a function is a vector field, and for a given point *x*, points in the direction of the greatest increase of $f(\vec{x})$. Because the matrix *A* is positive-definite, the surface defined by the function $f(\vec{x})$ presents a paraboloid shape, which ensures the existence of a global minimum. Moreover, the diagonal dominance of the matrix *A* ensures the existence of a unique solution. By applying Eqs. 5.37 and Eq. 5.38, we derive the formulation for the gradient of the function $f(\vec{x})$, given by

$$f'(\vec{x}) = \frac{1}{2} A^T \vec{x} + \frac{1}{2} A\vec{x} - b \,. \tag{5.39}$$

If the matrix *A* is symmetric, Eq. 5.39 reduces to

$$f'(\vec{x}) = A\vec{x} - b \,. \tag{5.40}$$

Therefore, by setting $f'(\vec{x})$ in Eq. 5.40 equal to zero, we find the initial problem that we wish to solve.

5.5.1 The Conjugate Gradient (CG) Method

The CG method is based on finding the minimum of the function $f(\vec{x})$ using a line search method. The calculation begins by guessing a first set of search directions $\vec{d}_0$ using the residual as follows:

$$\vec{d}_0 = \vec{r}_0 = \vec{b} - A\vec{x}_0 \,. \tag{5.41}$$

The multiplier α for the search directions is calculated as follows

$$\alpha_i = \frac{\vec{r}_i^{\,T} \vec{r}_i}{\vec{d}_i^{\,T} A \vec{d}_i} \,, \tag{5.42}$$

where *i* is the iteration index.

The multiplier α is chosen such that the function $f(\vec{x})$ is minimized along the search direction. Therefore, the solution and the residuals are updated using Eqs. 5.43 and 5.44.

$$\vec{x}_{i+1} = \vec{x}_i + \alpha_i \vec{d}_i , \tag{5.43}$$

$$\vec{r}_{i+1} = \vec{r}_i - \alpha_i A\vec{d}_i . \tag{5.44}$$

The Gram-Schmidt orthogonalization method is used to update the search directions by requiring the residuals to be orthogonal at two consecutive iterations. The orthogonalization method consists of calculating the search directions

$$\vec{d}_{i+1} = \vec{r}_{i+1} + \beta_{i+1}\vec{d}_i , \tag{5.45}$$

where the coefficients β are given by

$$\beta_{i+1} = \frac{\vec{r}_{i+1}^{\,T}\vec{r}_{i+1}}{\vec{r}_i^{\,T}\vec{r}_i} . \tag{5.46}$$

Note that Eq. 5.44 indicates that the new residuals are a linear combination of the residual at the previous iteration and $A\vec{d}_i$. It follows that the new search directions are produced by a successive application of the matrix operator A on the directions at a previous iteration $\vec{d}_i$. The successive application of the matrix operator A on the search directions $\vec{d}_i$ generates a vector space called Krylov subspace, represented by

$$\mathrm{K}_i = span\left\{\vec{d}_0, A\vec{d}_0, A^2\vec{d}_0, ..., A^{i-1}\vec{d}_0\right\}. \tag{5.47}$$

This iterative procedure is terminated when the residuals satisfy the following convergence criterion

$$MAX\left(\left|r_{i+1}\right|\right) \le \varepsilon , \tag{5.48}$$

where ε is the value of the tolerance, which is usually set to 1.0e-6.

5.5.2 The Bi-Conjugate Gradient Method

The Bi-Conjugate Gradient (Bi-CG) has been developed for solving non-symmetric linear systems. The update relations for the residuals are similar to the CG method;

however, they involve the transpose of the matrix operator. Hence, the residuals and the search directions are updated with the following equations:

$$r_i = r_{i-1} - \alpha_i A p_i , \tag{5.49a}$$

$$\tilde{r}_i = \tilde{r}_{i-1} - \alpha_i A^T \tilde{p}_i , \tag{5.49b}$$

$$p_i = r_{i-1} + \beta_{i-1} p_{i-1} , \tag{5.49c}$$

$$\tilde{p}_i = \tilde{r}_{i-1} + \beta_{i-1} \tilde{p}_{i-1} , \tag{5.49d}$$

where

$$\alpha_i = \frac{\tilde{r}_{i-1}^T r_{i-1}}{\tilde{p}_i^T A p_i} , \text{ and } \beta_i = \frac{\tilde{r}_i^T r_i}{\tilde{r}_{i-1}^T r_{i-1}} . \tag{5.50}$$

5.5.3 Preconditioners for Krylov Subspace Methods

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. The main philosophy of preconditioning is based on the attempt to transform the linear system into one that preserve the solution, but that has more favorable spectral properties. The spectral radius in norm L_2 for a symmetric matrix A is defined by

$$\rho(A) = \|A\|_2 . \tag{5.51}$$

The spectral radius so defined, gives an indication of the convergence behavior of the iterative method used. In the case of preconditioning, if a matrix M approximates the coefficient matrix A, the transformed system

$$M^{-1} A\vec{x} = M^{-1}\vec{b} , \tag{5.52}$$

has the same solution of the original system $A\vec{x} = \vec{b}$, but the spectral radius of its coefficient matrix $M^{-1}A$ is generally smaller than the original system. Various preconditioning techniques include the Jacobi or diagonal scaling, the Incomplete

Cholesky, and the multigrid. The Jacobi preconditioner is the most straightforward preconditioner and it is based on using the main diagonal of the matrix *A*.

$$m_{i,j} = \begin{cases} a_{i,i} \text{ if i = j} \\ 0 \quad \text{otherwise.} \end{cases} \tag{5.53}$$

This method is the least demanding in terms of memory requirements and computation time; however, the method also presents limited performance characteristics.

I have developed an Incomplete Cholesky preconditioner for the Conjugate Gradient (ICCG) method. This method is well suited for symmetric definite matrices and it is based on decomposing the matrix *A* using the Cholesky factorization method.[24] Since the matrix is symmetric, only the lower triangular part *L* is computed, thereby saving half of the operation required for a classic LU decomposition. The preconditioning matrix can be written as follows

$$M = LL^T . \tag{5.54}$$

The elements of the matrix *L*, decomposed with Incomplete Cholesky algorithm are given by

$l_{11} = a_{11}^{1/2}$
For $i = 2$ to n
  For $j = 1$ to $i - 1$
    If $a_{ij} = 0$ then $l_{ij} = 0$ else
      $l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}\right) / l_{jj}$
  $l_{ij} = \left(a_{ii} - \sum_{k=1}^{j-1} l_{ik}^2\right)^{1/2}$

CHAPTER 6
DEVELOPMENT AND BENCHMARKING OF THE PENSSN CODE

In this chapter, I will present the development of the new PENSSn code, and then I will test its numerics and the accuracy. In particular, I will address the performance of the Krylov subspace methods, including the CG and Bi-CG iterative solvers, along with the Incomplete Cholesky preconditioner for the CG method. The accuracy of the EP-SS_N method will be tested for the following parameters

- Scattering ratio;
- Spatial truncation error;
- Low density materials;
- Material discontinuities;
- Anisotropic scattering.

In addition, I will analyze the method based on two 3-D criticality benchmark problems proposed by Takeda and Ikeda.[43] The first problem involves the simulation of the Kyoto University Critical Assembly (KUCA) reactor. This problem is characterized by significant transport effects due to the presence of a control rod and a void-like region. The second problem involves the simulation of a small Fast Breeder Reactor (FBR) with a control rod half-inserted into the core. The solutions obtained for these two benchmarks will be compared with the Monte Carlo and S_N methods.

Finally, I will present the results obtained for the OECD/NEA[1] MOX 2-D Fuel Assembly Benchmark problem.[44]

[1] OECD/NEA - Organisation for Economic Co-operation and Development/Nuclear Energy Agency

6.1 Development of the PENSSn (Parallel Environment Neutral-particle Simplified Sn) Code

I have developed a new 3-D radiation transport code, PENSSn, based on the EP-SS_N formulation. The code development began in 2001 utilizing the Simplified P_3 formulation, that led to the development of the $PENSP_3$ (Parallel Environment Neutral-particle SP_3) code.[25] However, the extension of the SP_3 algorithm to an arbitrary order (N) proved to be impractical. Hence, I redirected the work by deriving a 3-D EP-SS_N formulation. PENSSn consists of ~10,000 lines of code entirely written in ANSI/FORTRAN-90, using the Message Passing Interface (MPI) libraries for parallelization.[27]

PENSSn is a standalone code which solves the multigroup EP-SS_N equations of arbitrary order with arbitrary anisotropic scattering expansion. To improve the convergence rate of the Source Iteration method, a modified formulation of the EP-SS_N equations (see Section 4.2.3) has been integrated into PENSSn. Currently both fixed source and criticality eigenvalue calculations can be performed with up- and down-scattering processes.

The discretized EP-SS_N equations are solved using the Krylov subspace methods described in Chapter 5, i.e. CG and Bi-CG. However, in the parallel version of PENSSn, only the Bi-CG algorithm is implemented due to its superior parallel performance and numerical robustness as compared to CG.

Angular, spatial and hybrid (spatial/angular) domain decomposition algorithms have been developed to achieve full-memory partitioning and multi-tasking. The code is capable of parallel I/O in order to deal efficiently with large data structures. A complete description of the domain decomposition algorithms is given in Chapter 7.

PENSSn produces balance tables and a complete description of the model solved, along with performance and timing data. The code is completed by a parallel data processor, PDATA, which collects the output files produced by different processors and generates a single file for each energy group for plotting or further analysis.

Currently, the geometry and material distribution are prepared for PENSSn using the PENMSH[45] tool in the PENTRAN Code System. PENSSn requires only one additional input file which is defined as *problem_name.psn*. The PENSSn input file is shown in Figure 6-1.

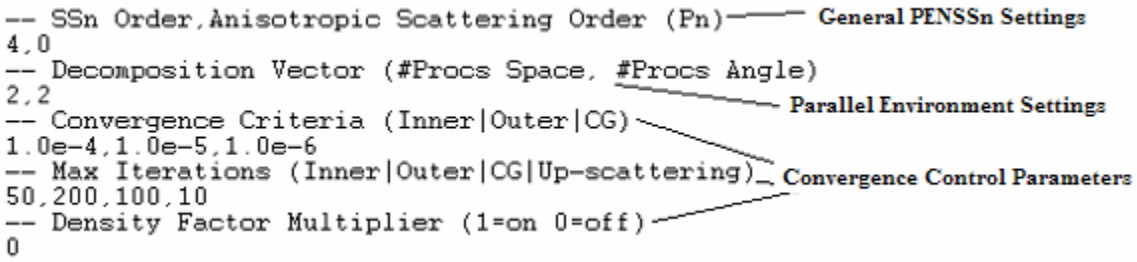

Figure 6-1. Description of PENSSn input file.

As shown above, the input file provides three groups of information:

- General PENSSn settings;
- Parallel Environment settings;
- Convergence control parameters.

The General PENSSn group is used to input the SS_N and P_N order for the calculation. Note that the SS_N order is an even number and also it holds the condition $SS_N > P_N$.

The Parallel Environment group is used to specify the decomposition vector for the parallel environment. Note that the number of coarse meshes has to be divisible by the number of processors specified for the spatial domain, and also the number of directions has to be divisible by the number of processors specified for the angular domain.

The Convergence control parameters group is used to specify the inner, outer, and Krylov subspace (CG) tolerances. Also the maximum number of inner, outer, up-scattering and Krylov iterations can be specified.

PENSSn can be run in parallel or serial mode; note that for the serial mode version, both CG and Bi-CG algorithms are available. A flow-chart for the PENSSn code is shown in Figure 6-2.

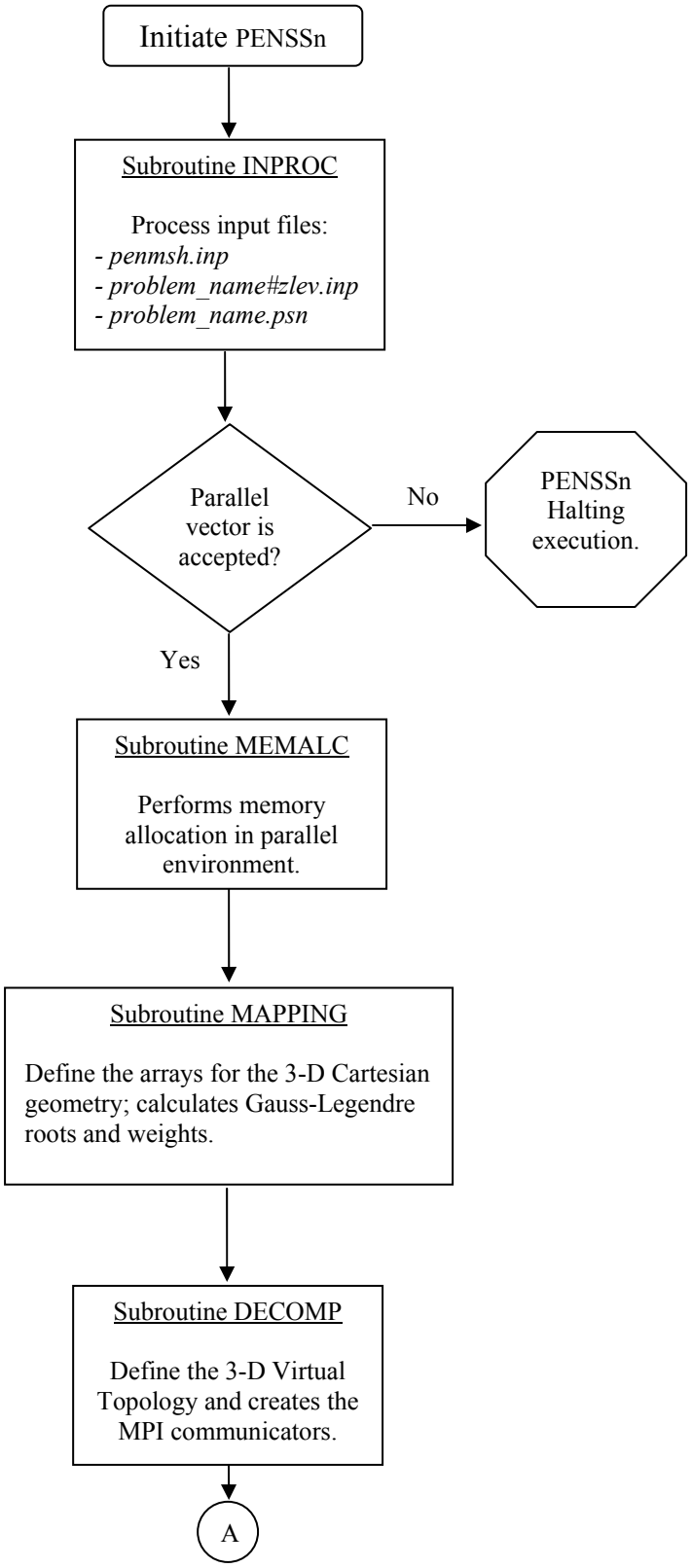

Figure 6-2. Flow-chart of the PENSSn code.

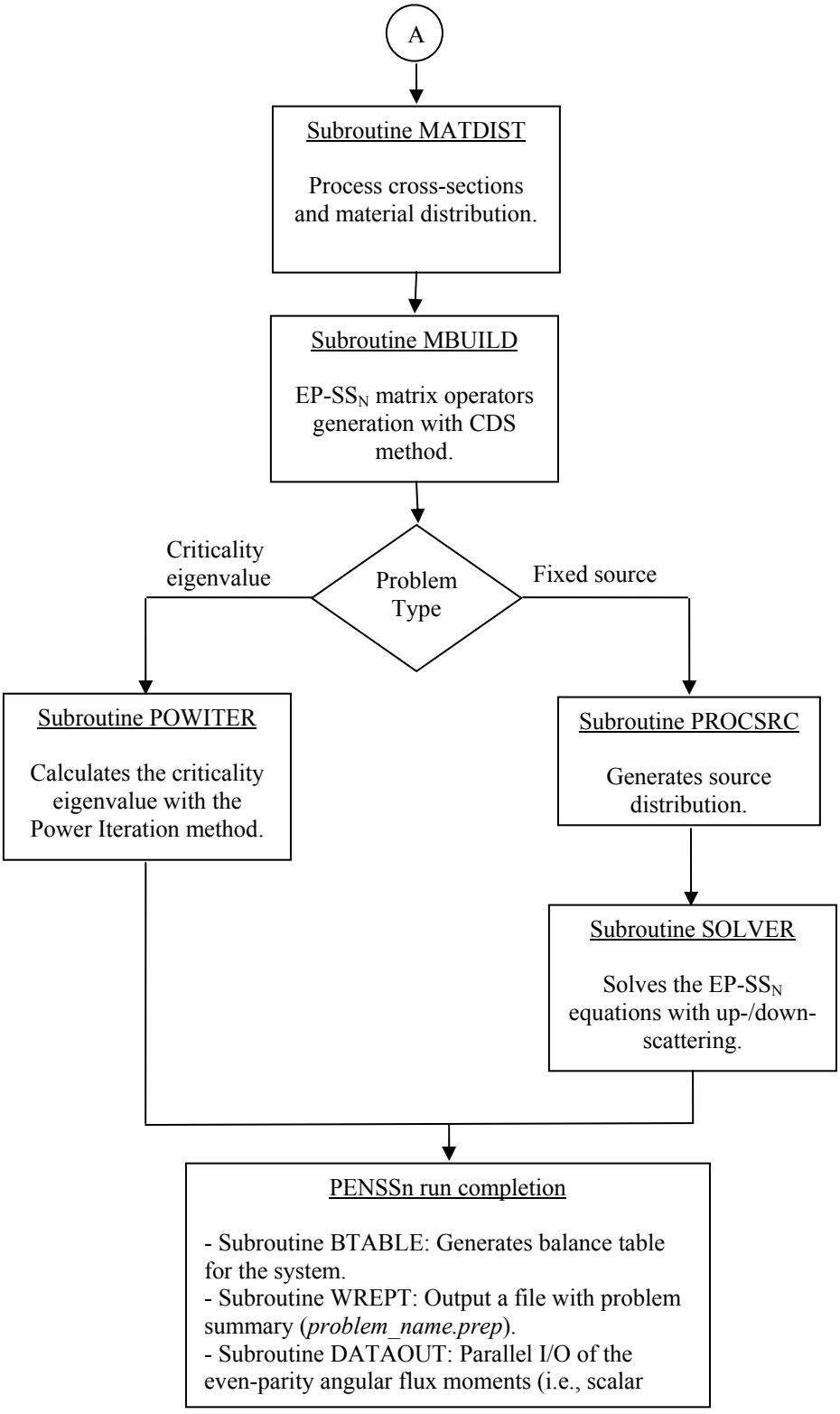

Figure 6-2. Flow-chart of the PENSSn code (Continued).

6.2 Numerical Analysis of Krylov Subspace Methods

In this section, I will present a detailed analysis for the CG and Bi-CG algorithms as applied to problems with different numerical properties. In particular, I will analyze the convergence performance of these algorithms in the following cases:

- Coarse mesh partitioning of the model;
- Boundary conditions;
- Material heterogeneities;
- Higher order EP-SS_N methods.

6.2.1 Coarse Mesh Partitioning of the Model

In this section, I will study the performance of the iterative solvers when the model is partitioned into coarse meshes. The first test problem consists of a simple symmetric 3-D problem shown in Figure 6-3. The problem size is 10.0x10.0x10.0 cm; a uniform distributed source is located within a cube of side 5.0 cm. Vacuum boundary conditions are prescribed for this model on every surface. The model is characterized by one homogeneous material with one-group cross-sections; the total cross-section is equal to 1.0, and the scattering ratio is equal to 0.9.

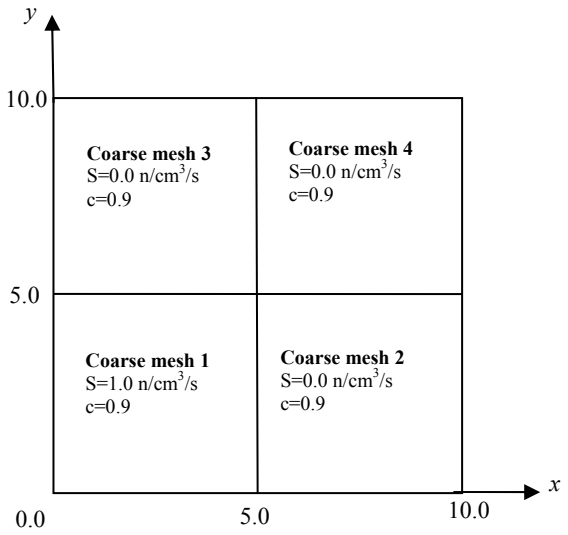

Figure 6-3. Configuration of the 3-D test problem.

The system is discretized with a 1.0 cm uniform mesh along the *x*-, *y*- and *z*-axes. The EP-SS_2 equation is solved using the CG and Bi-CG algorithms; the convergence criteria for the source iteration and the Krylov methods are 1.0e-5 and 1.0e-6, respectively. The formulation used for the convergence criterion in the source iteration is given in Eqs. 6.1.

Source iteration method convergence criterion

$$\left\| \frac{\psi_{m,g}^{i}(\vec{r}) - \psi_{m,g}^{i-1}(\vec{r})}{\psi_{m,g}^{i-1}(\vec{r})} \right\|_{\infty} < 1.0e^{-5}. \tag{6.1}$$

Table 6-1 compares the number of iterations for the Krylov solvers, CG and Bi-CG in two cases. In the first case, the model is partitioned into coarse meshes (Partitioned model); in the second case, the model is considered as a whole and no coarse meshes are specified (Non-partitioned).

Table 6-1. Comparison of number of iterations required to converge for the CG and Bi-CG algorithms.

	Partitioned model		Non-partitioned model	
Method	Krylov iterations	Inner iterations	Krylov iterations	Inner iterations
Bi-CG	995	58	165	57
CG	1620	58	270	57

An increase of a factor of 6 is observed in the Krylov iterations by partitioning the model into coarse meshes. The coarse mesh partitioned model requires a larger number of iterations to converge, because the values of the angular fluxes on the interfaces of the coarse meshes are calculated iteratively. Notice that this effect is purely numerical and only related to the Krylov solvers. In fact, I did not observe any significant change in the number of inner iterations, which is exclusively related to the scattering ratio and hence to the physics of the problem.

I calculated the spectral condition number with an L_2 norm for the partitioned and the non-partitioned system. The spectral condition number in L_2 norm is defined by

$$k_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}, \tag{6.2}$$

where, $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the maximum and minimum eigenvalues of the matrix A.

The spectral condition number gives an indication of the convergence behavior of the iterative method. For the CG algorithm the number of iterations required to reach a relative reduction of ε (one order of magnitude) in the error is proportional to $\sqrt{k_2}$. For the non-partitioned model, I obtained $k_2 = 4.6$, while for the coarse mesh partitioned model, I obtained $k_2 = 4.0$ in each coarse mesh.

Figure 6-4 confirms the prediction based on the spectral condition number; the number of iterations required to reduce the error by one order of magnitude is approximately 2.1.

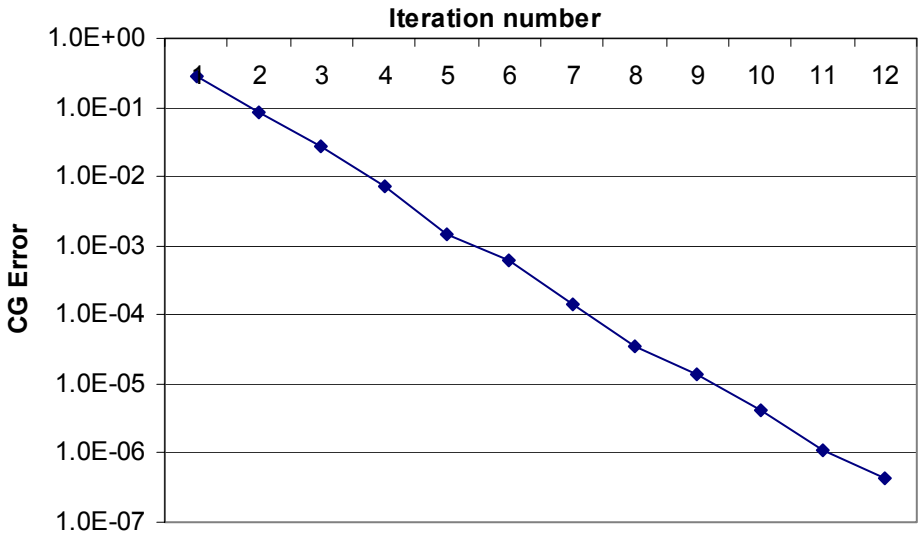

Figure 6-4. Convergence behavior of the CG algorithm for the non-partitioned model.

Based on these results, I conclude that the increase in the number of Krylov iterations observed between the partitioned and non-partitioned models is due to the presence of the coarse mesh interfaces. Moreover, these tests show the superior performance of the Bi-CG algorithm compared to CG; the Bi-CG algorithm requires only ~61% of the CG iterations for both the non-partitioned and partitioned models.

6.2.2 Boundary Conditions

The objective of the following test problem is to analyze the effect of different boundary conditions on the convergence behavior of the Krylov solvers. The 3-D test problem used in the previous section has been modified by prescribing reflective boundary conditions on the planes at x=0.0, y=0.0 and z=0.0, and vacuum boundary conditions on the planes at x=10.0 cm, y=10.0 cm and z=10.0 cm. The model is partitioned into four coarse meshes, which are discretized with a 1.0 cm uniform mesh. Table 6-2 lists the number of iterations required by the Bi-CG and CG method to achieve convergence, along with the spectral condition number (k_2) calculated for each coarse mesh.

Table 6-2. Number of Krylov iterations required to converge for the CG and Bi-CG algorithms with different boundary conditions.

Coarse mesh	k_2	Bi-CG method	CG method
1	4.5	399	649
2	4.25	361	595
3	4.25	362	594
4	4.05	338	546

As expected, the number of Krylov iterations is higher for coarse meshes with larger condition number. However, the Krylov solvers require a different number of iterations per each coarse mesh, due to the effect of boundary conditions. As will be

discussed in Chapter 7, this situation will have a slight impact on the performance of the parallel algorithm.

6.2.3 Material Heterogeneities

Realistic engineering applications are characterized by material heterogeneities; hence, it is important to determine their impact on the Krylov solvers. For this purpose, I have modified the 3-D test problem described earlier, with a heterogeneous material configuration, shown in Figure 6-5. The boundary conditions prescribed are the same as for the previous test case.

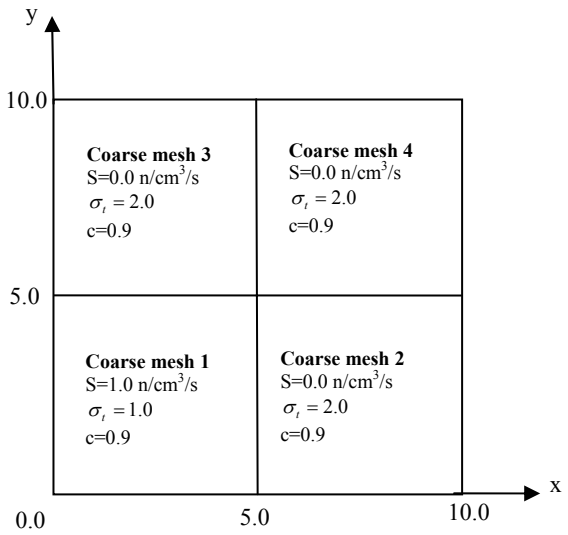

Figure 6-5. Heterogeneous configuration for the 3-D test problem.

Table 6-3 demonstrates that the spectral condition number is affected by the different material configuration in each coarse mesh, leading to a different number of iterations required by the Krylov solvers for each coarse mesh.

Table 6-3. Number of Krylov iterations required to converge for CG and Bi-CG.

Coarse mesh	k_2	Bi-CG method	CG method
1	4.5	539	875
2	1.88	302	518
3	1.88	302	518
4	1.86	282	483

Note that the spectral condition numbers for coarse meshes 2, 3, and 4 are relatively lower than for coarse mesh 1. This is due to the fact that the materials in coarse meshes 2, 3, and 4 are optically thicker than region 1. The optical thickness directly impacts the condition number. In general, optically thin regions present matrix operators with larger spectral condition numbers; conversely, optically thick regions present matrices with smaller spectral condition numbers.

6.2.4 Convergence Behavior of Higher EP-SS_N Order Methods

In this section, I will analyze the convergence behavior of the Krylov methods for high EP-SS_N order methods. The test problem considered is a cube with homogeneous material, and one group cross-sections with c=0.9. The side of the cube measures 5.0 cm and it is discretized with a 1.0 uniform mesh. Vacuum boundary conditions are prescribed on every side of the model; also a uniform distributed source is present in the model, which emits 1.0 particles/cm^3/sec. This problem has been solved with the EP-SS_8 equations; the convergence criteria prescribed for the source iteration and Krylov methods are 1.0e-5 and 1.0e-6, respectively. Table 6-4 shows the spectral condition number (k_2) as a function of direction for the EP-SS_8 equations.

For simplicity I have selected the EP-SS_8 equations; however the discussion below can be extended to any SS_N order.

Table 6-4. Number of Krylov iterations required to converge for the CG and Bi-CG algorithms for the EP-SS_8 equations.

Direction number	Direction cosine (μ)	k_2	Bi-CG method	CG method
1	0.1834346	1.34	49	85
2	0.5255324	3.35	73	114
3	0.7966665	5.62	89	144
4	0.9602898	7.11	101	161

Table 6-4 indicates that the spectral condition number increases as the direction cosine approaches 1.0; hence, the number of Krylov iterations required to achieve convergence increases as well. This behavior can be explained by considering the definition of condition number (i.e., Eq. 6.2), and by observing the distribution of the eigenvalues for the matrix operators on each direction (μ), as shown in Figure 6-6.

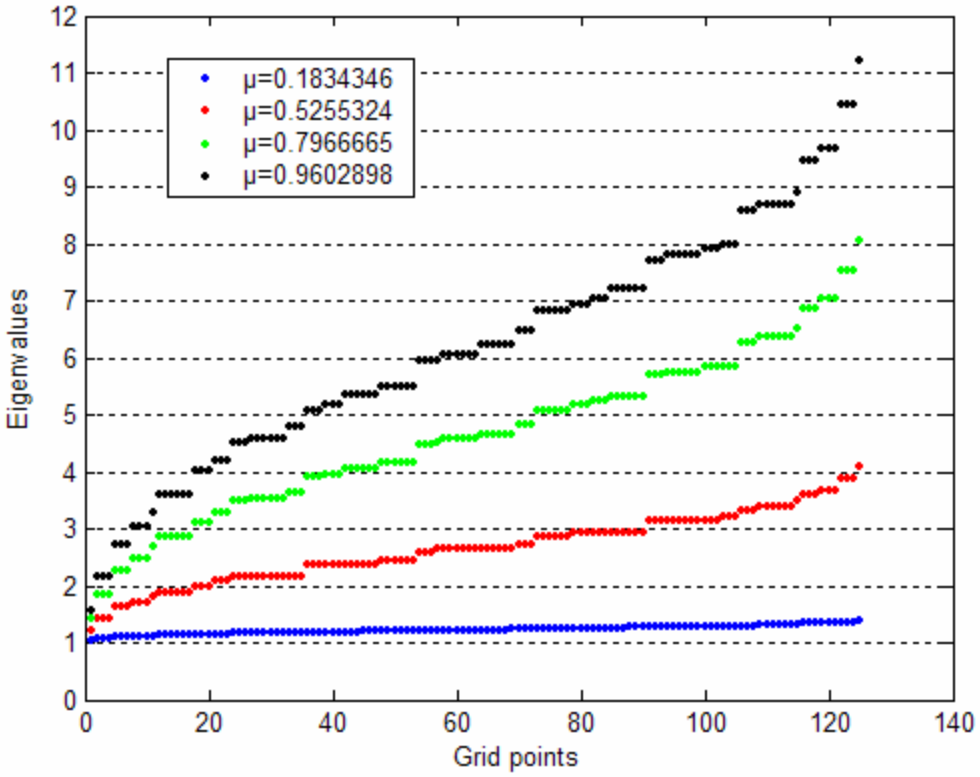

Figure 6-6. Distribution of eigenvalues for the EP-SS_8 equations.

The distribution of eigenvalues is clustered toward the value of 1.0 for smaller values of μ; however, as the direction cosine increases, the eigenvalue distributions start to drift away from 1.0. Therefore, based on the definition of condition number, those matrices with a distribution of eigenvalues clustered around 1.0 present the smallest condition number, and consequently the Krylov method requires fewer iterations to converge.

6.3 Testing the Incomplete Cholesky Conjugate Gradient (ICCG) Algorithm

This section presents the numerical testing of the Incomplete Cholesky Conjugate Gradient (ICCG) algorithm. In order to optimize the preconditioner for large sparse matrices, I have utilized the Incomplete Cholesky (IC0) no-fill factorization. With this method the Cholesky factorization is computed only for non-zero elements of the EP-SS_N matrix operators.

The test problem utilized is a 3-D cube with a homogeneous material. The side of the cube measures 5.0 cm. The boundary conditions prescribed, are reflective on the planes along x=0.0 cm, y=0.0 cm, and z=0.0 cm, and vacuum on the planes along x=5.0 cm, y=5.0 cm, and z=5.0 cm. The model is discretized with a 1.0 cm uniform mesh along the x, y, and z axis. The convergence criteria prescribed for the source iteration and ICCG methods are 1.0e-5 and 1.0e-6, respectively. Table 6-5 shows the number of iterations for the ICCG method compared to the non-preconditioned CG algorithm.

Table 6-5. Number of iterations for the ICCG and CG algorithms.

Method	ICCG	CG
EP-SS_4	37	68
EP-SS_6	65	120
EP-SS_8	91	173

For SS_N orders ranging from 4 to 8, the ICCG method yields a reduction of the number of iterations by a factor of two.

The main disadvantage of the ICCG method is the computation time and memory required to perform the Incomplete Cholesky factorization of the coefficient matrices; I have observed that for large problems, the factorization phase account for ~30% of the total computation time. In addition, the time spent for each Krylov iteration increases because the preconditioning matrix has to be solved as well. However, the ICCG method

yields an overall reduction of the computation time by a factor of ~1.5. In spite of its good performance, the ICCG method has not been implemented in the current algorithms; this decision has been dictated mainly by the large memory required by the ICCG method.

6.4 Testing the Accuracy of the EP-SS_N Method

In this section, I will test the accuracy of the EP-SS_N equations, and I will identify the limitations of the methodology with respect to scattering ratio, spatial truncation error, low density materials, material discontinuities, and anisotropic scattering order.

In conclusion, I will analyze the accuracy of the EP-SS_N using two 3-D criticality benchmark problems proposed by Takeda and Ikeda.[44]

6.4.1 Scattering Ratio

The objective of this test is to calculate the criticality eigenvalue (k_{eff}) as a function of the scattering ratio (c). For this purpose, I will consider a simple 2-D criticality eigenvalue problem with $0.0 \le x, y \le 4.0$. The boundary conditions prescribed are reflective at x=0.0, y=0.0, and vacuum at x=4.0 cm, y=4.0. The model is discretized with a 0.25 cm uniform mesh. The model configuration is shown in Figure 6-7.

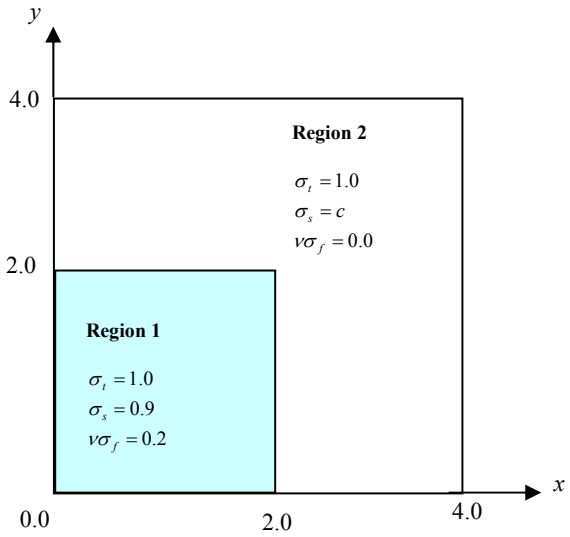

Figure 6-7. Configuration of the 2-D criticality eigenvalue problem.

For this problem, I will progressively modify the scattering ratio in region 2; however, as the scattering ratio decreases, the outer region becomes less diffusive, so it is expected that the diffusion equation will be less accurate compared to higher order SS_N methods.

Figure 6-8 shows the comparison among the criticality eigenvalues obtained with the EP-SS_N and S_{16} methods for different scattering ratios in a range of 0.6 – 0.99. The S_{16} transport solutions have been obtained with the PENTRAN and DORT codes.

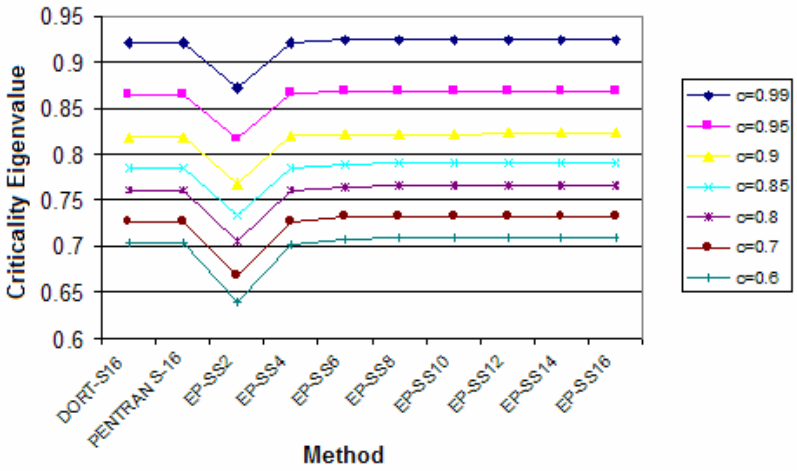

Figure 6-8. Criticality eigenvalues as a function of the scattering ratio (c) for different methods.

Based on the data presented in Figure 6-8, the EP-SS_2 method yields inaccurate results for every scattering ratio. Because of large particle leakage from the system, the EP-SS_2 method yields inaccurate results also for scattering ratio greater than 0.9, where the physics is dominated by diffusive processes. However, higher order EP-SS_N methods yield accurate results for every scattering ratio, with a maximum relative difference of 0.85% compared to the S_{16} transport calculation.

Note also that by increasing the SS_N order, the accuracy is not improved as well; this behavior is due to the fact that the EP-SS_N formulation does not yield the transport solution as the order increases.

Figure 6-9, clearly shows the increased accuracy obtained using higher order EP-SS_N methods relative to the diffusion equation.

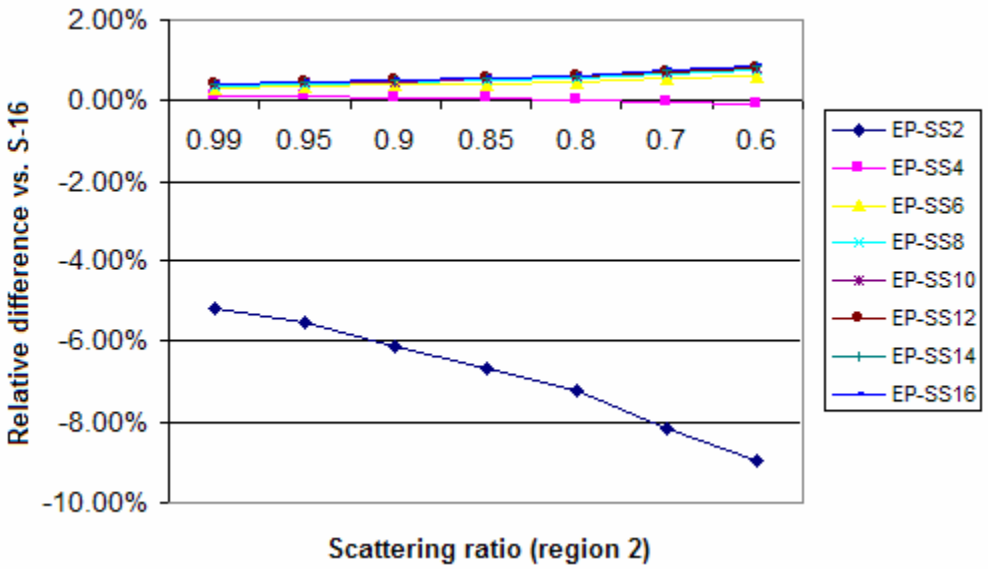

Figure 6-9. Relative difference for criticality eigenvalues obtained with different EP-SS_N methods compared to the S_{16} solution (PENTRAN code).

It is interesting to note that the highest accuracy is achieved with the EP-SS_4 method with a relative difference of 0.12% for c=0.99. As will be shown in the next section, this behavior is partly due to the spatial truncation error. As expected all the EP-

SS_N methods with N>2 yield accurate results for every scattering ratio, while the diffusion equation degrades as the scattering ratio decreases.

6.4.2 Spatial Truncation Error

In order to further test the accuracy of the EP-SS_N method, I have investigated the effect of the spatial discretization for criticality eigenvalue calculations. For this purpose I solved the problem presented in Section 6.3.1 with different fine mesh discretizations. The case considered has scattering ratio in region 2 equal to 0.6 (see Figure 6-6).

I have compared the solutions obtained with the EP-SS_N method with orders ranging from 4 to 10, with an S_{16} transport calculation obtained with the PENTRAN and DORT codes. Figure 6-10 presents the criticality eigenvalues obtained with these methods for different fine mesh discretizations.

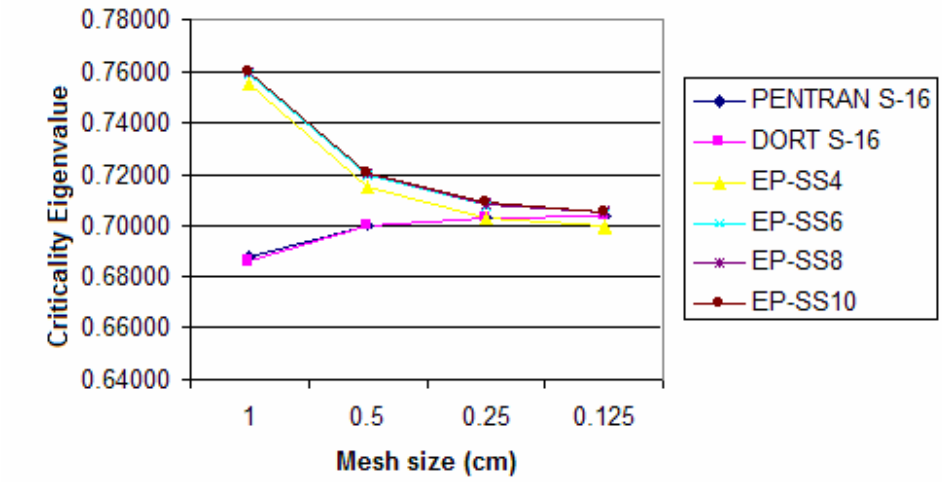

Figure 6-10. Plot of criticality eigenvalues for different mesh sizes.

As we can see, the EP-SS_N method converges to the transport solutions by increasing the spatial resolution of the problem; however, the EP-SS_4 equations yield more accurate results compared to higher order SS_N equations, down to a mesh size of 0.25 cm.

Figure 6-11 shows the relative difference of the EP-SS_N solutions compared to the S_{16} transport calculation obtained with the PENTRAN code.

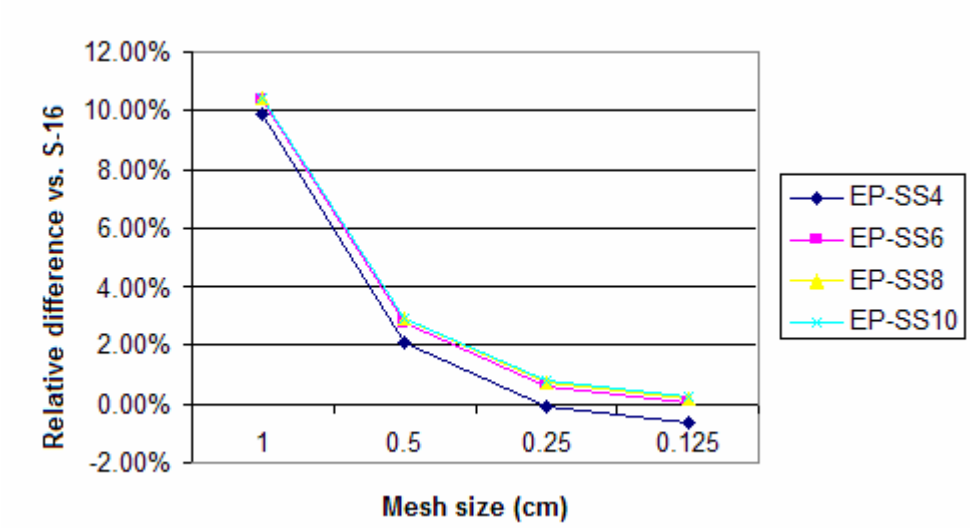

Figure 6-11. Plot of the relative difference of the EP-SS_N solutions versus transport S_{16} for different mesh sizes.

The EP-SS_4 method yields relative differences of 9.87%, 2.06% and -0.09% for mesh sizes of 1.0, 0.5 and 0.25 cm, respectively. For a mesh size of 0.125 cm, the EP-SS_6 method yields the most accurate results, with a relative difference of 0.03%. In conclusion, this test problem demonstrates that the spatial discretization has to be refined as the SS_N is increased; note that this behavior is similar to what is observed for the S_N method.

6.4.3 Low Density Materials

The problem considered in this section consists of two blocks of uranium dioxide highly enriched at 93.2 %, surrounded by air with an 80% relative humidity. Figure 6-12 shows a view on the *x-y* plane of test problem considered. The model extends along the *z*-axis from 0.0 to 12.0 cm, and vacuum boundary conditions are prescribed on all surfaces.

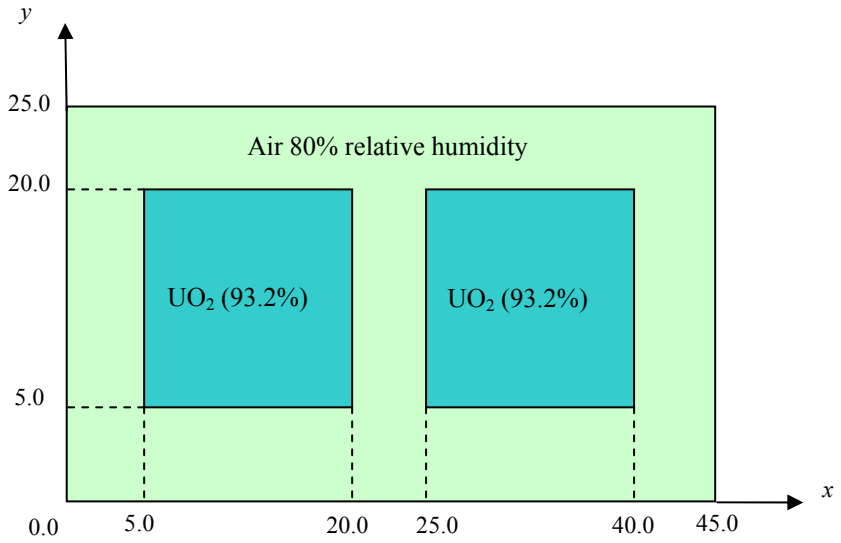

Figure 6-12. Uranium assembly test problem view on the *x*-*y* plane.

The purpose of this test problem is to assess the accuracy of the EP-SS_N methodology in the presence of low density media such as air gaps. In low density or void-like regions, the particle physics is not dominated by diffusive processes, where elliptic-type mathematical models such as the EP-SS_N equations yield accurate results. In this type of problems the particle behavior is well described by the transport equation which behaves like a hyperbolic wave equation.

The major issue affecting the accuracy of the EP-SS_N equations is the diffusion coefficient defined by

$$D_m(\vec{r}) = \frac{\mu_m^2}{\sigma_t(\vec{r})}. \tag{6.3}$$

In low density media, the value of the total scattering cross-section is usually below 1.0e-3 [1/cm]. Hence, the value of the diffusion coefficient becomes abnormally large, leading to numerical difficulties and to an underestimation of the leakage term. In order to remedy this situation, I have introduced a *density factor multiplier* (DFM) in order to scale up the cross-section only in low density regions. The density factor is applied to the

cross-sections of materials for which the total cross-sections have a value within 1.0e-3 and 1.0e-7. The value of DFM is chosen such that the scaled cross-sections have a value of ~1.0e-1. This is an ad-hoc treatment, but it has been proven effective in improving the convergence properties and accuracy of the EP-SS_N method for problems with low density regions. The two-group cross-sections and the fission spectrum are listed in Table 6-6; group 1 spans an energy range between 0.4 and 10.0 MeV, while group 2 spans a range between 0.0 and 0.4 MeV.

Table 6-6. Two groups cross-sections and fission spectrum.

Material	Group (g)	σ_a	$\nu\sigma_f$	σ_t	$\sigma_{s,g\to1}$	$\sigma_{s,g\to2}$	$\chi(g)$
UO_2	1	6.1902e-02	1.4436e-01	2.3968e-01	1.5220e-01	0.0	0.896
	2	8.6126e-02	1.7309e-01	4.2551e-01	3.3938e-01	2.5582e-02	0.104
Air	1	3.3372e-06	0.0	1.0115e-04	8.6948e-05	0.0	0.0
	2	6.1639e-06	0.0	2.8127e-04	2.7511e-04	1.0868e-05	0.0

For this problem, I calculated physical quantities of interest such as k-effective, leakage, collision and scattering term. Then, I compared these quantities with a transport solution obtained with the PENTRAN code. Table 6-7 compares the criticality eigenvalues obtained by the EP-SS_N method with different DFM values, with the PENTRAN S_6 solution.

Table 6-7. Comparison of k_{eff} obtained with the EP-SS_N method using DFM versus PENTRAN* S_6 (Note that DFM=1.0 implies no cross-sections scaling).

Method	k_{eff} (DFM=100.0)	Rel. difference vs. S_6	k_{eff} (DFM=1.0)	Rel. difference vs. S_6
EP-SS_2	0.77621	-18.47%	0.69437	-27.07%
EP-SS_4	0.90544	-4.90%	0.83097	-12.72%
EP-SS_6	0.92138	-3.23%	0.8464	-11.10%
EP-SS_8	0.92436	-2.91%	0.84894	-10.84%
EP-SS_{10}	0.92529	-2.82%	0.8496	-10.77%

*PENTRAN S_6 predicts k_{eff}= 0.95211.

Table 6-7 indicates that a density factor of 100.0, largely improves the accuracy of k_{eff}. However, if DFM is not utilized, the accuracy of the EP-SS_N method is poor for this

problem, and the convergence trend is characterized by an oscillatory behavior which leads to divergence within the maximum number of outer iterations specified (50).

Figure 6-13 shows the relative difference between the physical quantities calculated with EP-SS_N (DFM=100.0) and S_6 PENTRAN methods.

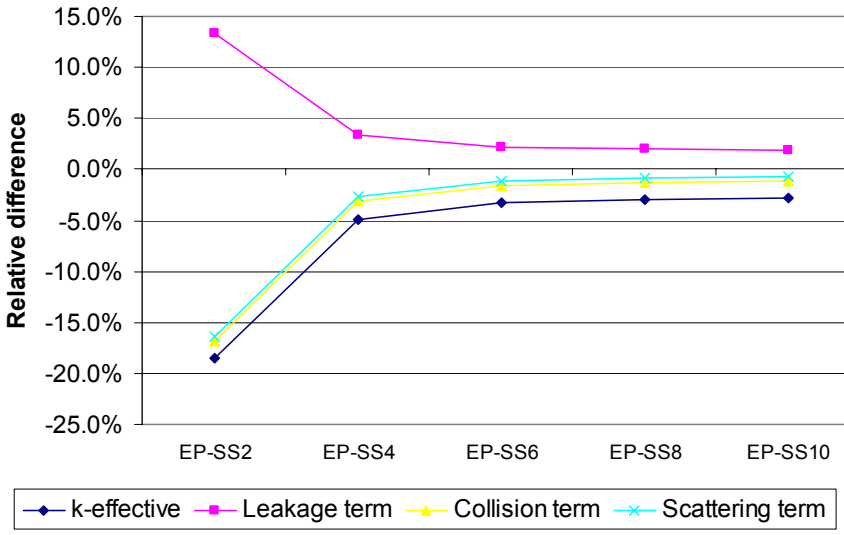

Figure 6-13. Relative difference of physical quantities of interest calculated with EP-SS_N method compared to the S_6 PENTRAN solution.

For this problem, transport effects are significant due to the large boundary leakage and highly angular behavior due to the low density medium. Hence, higher order EP-SS_N methods yield a better angular representation of the particle flux, therefore leading to more accurate results.

The EP-SS_{10} method yields a relative difference compared to the S_6 solution of -2.8%, 1.9%, -1.2% and -0.7% for k_{eff}, leakage, collision and scattering term, respectively. Note also that the density factor (DFM=100.0) improves the convergence behavior of the EP-SS_N method.

Figure 6-14 presents the criticality eigenvalue relative error as a function of the outer iteration number for PENSSn with DFM=100.0 and PENTRAN S_6.

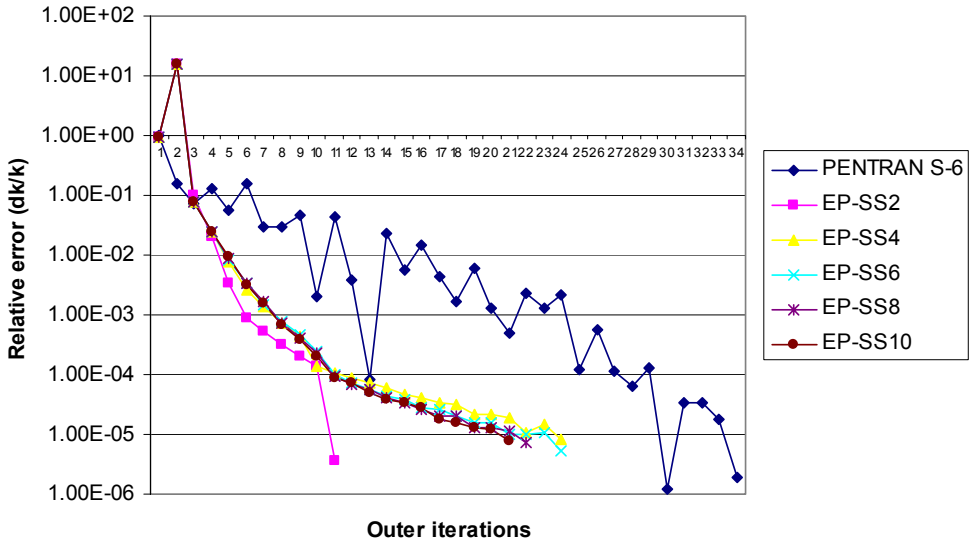

Figure 6-14. Convergence behavior of the PENSSn with DFM=100.0 and PENTRAN S_6.

The PENTRAN relative error presents an oscillatory behavior due to the Aitken's extrapolation method utilized.[39] The EP-SS_2 relative error presents a sudden drop from 1.0e-4 to 1.0e-5, probably indicating false convergence. The EP-SS_N calculations with N>2 all indicate a rather stable convergence behavior.

6.4.4 Material Discontinuities

In this section, I will analyze material discontinuities which may introduce significant angular dependencies on the particle flux at the material interface. The test problem considered is a simple 2-D model made of two heterogeneous regions, with a fixed source. The geometric and material configuration for the test problem is shown in Figure 6-15. The test problem is characterized by a steep change in the total cross-section between regions 1 and 2; also, region 2 is defined as a highly absorbent material. Because of these features the problem presents strong transport effects.

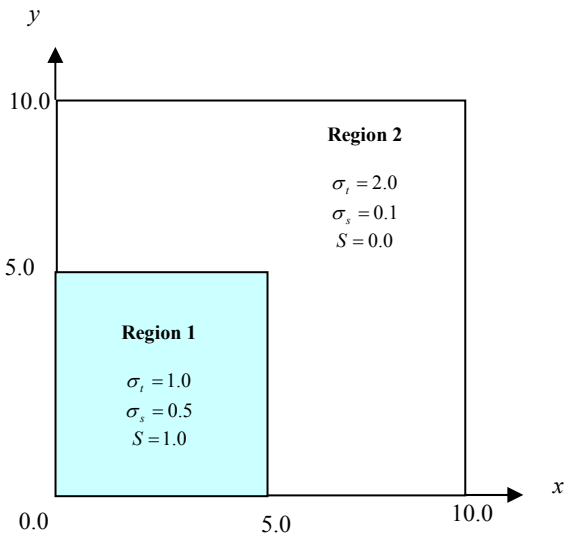

Figure 6-15. Geometric and material configuration for the 2-D test problem.

The solution for this problem is obtained with the EP-SS_2, EP-SS_4 and PENTRAN S_{16} methods. Figure 6-16 shows the flux distribution, along the *x*-axis.

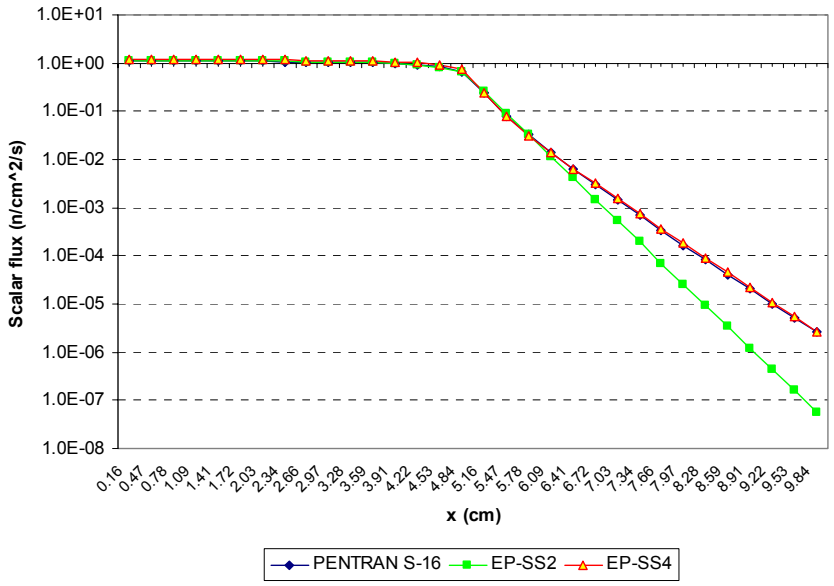

Figure 6-16. Scalar flux distribution at material interface (*y*=4.84 cm).

As indicated by Figure 6-16, the EP-SS_4 yields an accurate solution compared to S_{16}; the maximum relative difference between the two methods (15.58%) is found at *x*=4.84 cm and *y*=4.84 cm, which is the fine mesh on the corner of region 1. At this mesh

location, the transport effects due to material transition are significant, resulting in the largest difference between the EP-SS_4 and S_{16} methods. As expected, the EP-SS_2 method is accurate in region 1; however, the solution rapidly degrades as we move into region 2 where the transport effects are significant.

Figure 6-17 presents the relative difference for the EP-SS_2 and EP-SS_4 methods as compared to PENTRAN S_{16} at the material interface.

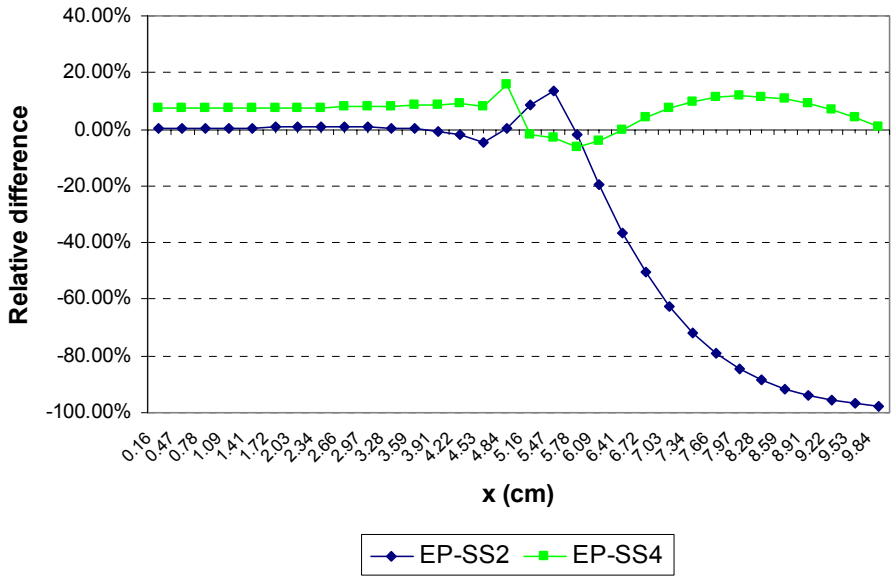

Figure 6-17. Relative difference versus S_{16} calculations at material interface (y=4.84 cm).

Figure 6-17 shows that the EP-SS_4 method exhibits a maximum relative difference of ~15.6% at the material interface. This problem clearly shows how higher order EP-SS_N methods introduce more transport physics into the solution compared to the diffusion-like equation.

The balance table (Table 6-8) demonstrates that the leakage term is the major component affecting the accuracy of the EP-SS_N method for problems with strong transport effects. The EP-SS_4 method yields a relative difference of only -1.12% for the

leakage term. Note that the collision and scattering terms, are relatively well represented by both EP-SS_2 and EP-SS_4 methods.

Table 6-8. Balance tables for the EP-SS_N and S_{16} methods and relative differences versus the S_{16} solution.

	Integral system balance			Relative difference vs. S_{16}		
Method	Leakage	Collision	Scatter	Leakage	Collision	Scatter
EP-SS_2	-1.76e-06	-4.61e+01	2.11e+01	-94.62%	-0.37%	-0.81%
EP-SS_4	-3.23e-05	-4.65e+01	2.15e+01	-1.12%	0.47%	1.03%
S_{16}	-3.27e-05	-4.62e+01	2.12e+01	-	-	-

These findings are further confirmed by observing the integral boundary leakage for different boundary surfaces. Table 6-9 clearly indicates that the predicted leakage rate is underestimated by ~98.7% using the EP-SS_2 method, while it is only underestimated by ~2% using the EP-SS_4 method.

Table 6-9. Integral boundary leakage for the EP-SS_N and S_{16} methods and relative differences versus the S_{16} solution.

	Integral boundary leakage		Relative difference vs. S_{16}	
Method	East (+x)	North(+y)	East (+x)	North(+y)
EP-SS_2	2.12e-07	2.12e-07	-98.70%	-98.70%
EP-SS_4	1.59e-05	1.59e-05	-1.98%	-1.99%
S_{16}	1.63e-05	1.63e-05	-	-

East (+x) refers to the right boundary of the system at x=10.0 cm, while North (+y) refers to the top boundary of the system at y=10.0 cm.

This is very encouraging because it indicates that the EP-SS_N methodology could be applicable for shielding problems.

6.4.5 Anisotropic Scattering

This section addresses the accuracy of the EP-SS_N equations for problems characterized by anisotropic scattering. The test problem consists of a cylinder with a 20.0 cm radius, extending axially for 30.0 cm, representing a fuel region; the cylinder is then surrounded by water extending from 20.0 cm to 30.0 cm along the x- and y-axis, and from 30.0 to 40.0 cm along the z-axis. Reflective boundary conditions are prescribed on

the planes at x=0.0 cm, y=0.0 cm and z=0.0 cm; vacuum boundary conditions are prescribed on the planes at x=30.0 cm, y=30.0 cm and z=40.0 cm. A two-group cross-section set is generated using the first two groups of the BUGLE-96 library with P_3 anisotropic scattering order. A uniform fixed source is placed in the fuel region, with an energy spectrum given in Table 6-10.

Table 6-10. Fixed source energy spectrum and energy range.

Energy group	Group upper boundary (MeV)	Group lower boundary (MeV)	Fixed source ($n/cm^3/s$)
1	17.3	14.2	4.25838e-5
2	14.2	12.2	1.84253e-4

I have compared the results obtained with the EP-SS_N method with a PENTRAN S_8 transport solution. The convergence criterion for the angular flux has been set to 1.0e-4. I calculated the relative difference between the solutions obtained with the EP-SS_N and the S_8 methods. Figures 6-18 and 6-19 show the fraction of scalar flux values within different ranges of relative difference (compared to S_8) for energy group 1 and 2 respectively.

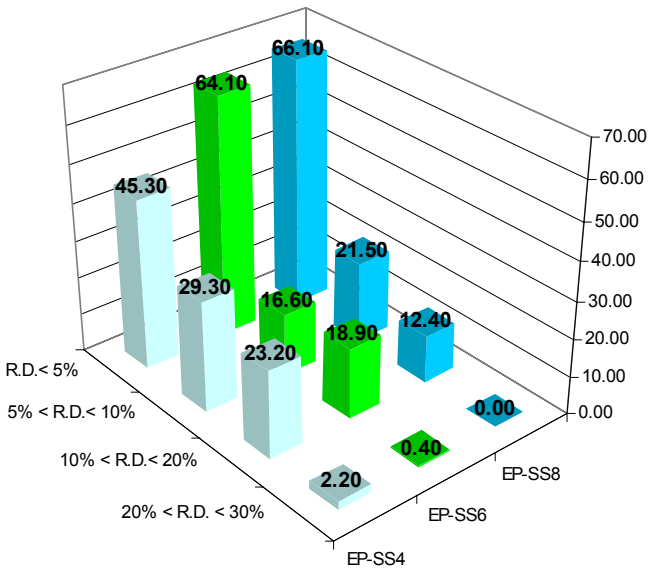

Figure 6-18. Fraction of scalar flux values within different ranges of relative difference (R.D.) in energy group 1.

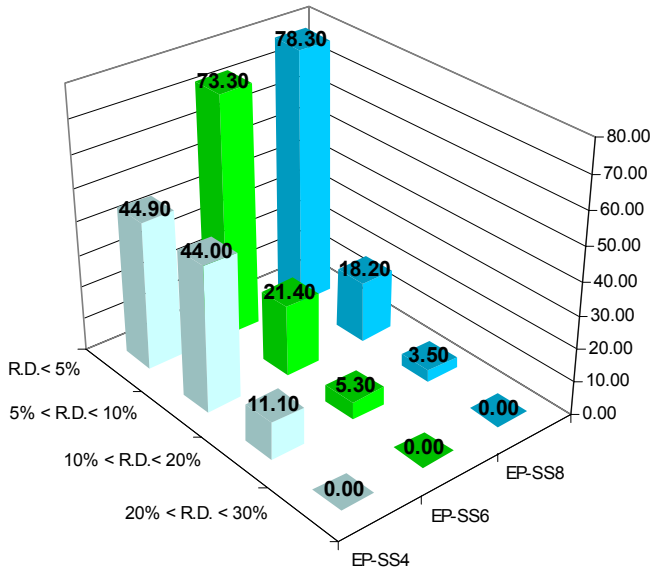

Figure 6-19. Fraction of scalar flux values within different ranges of relative difference (R.D.) in energy group 2.

Note that by increasing the SS_N order, the number of scalar flux values with relative difference less than 5% increases in both groups; this behavior demonstrates that higher order EP-SS_N methods improve the accuracy of the solution, especially for highly angular dependent problems. As expected the accuracy of the EP-SS_N method increases for lower energy groups because the probability of leakage decreases and the medium becomes optically thicker. Table 6-11 shows the maximum and minimum relative difference in the scalar flux versus the S_8 method[2], in energy groups 1 and 2.

Table 6-11. Maximum and minimum relative differences in the scalar flux versus the S_8 method for energy group 1 and 2.

	Group 1		Group 2	
Method	MAX	MIN	MAX	MIN
EP-SS_4	24.42	1.292e-03	17.86	1.379e-04
EP-SS_6	21.34	1.401e-04	15.01	6.053e-05
EP-SS_8	18.37	4.226e-04	13.74	2.508e-04

[2] The MAX and MIN relative difference compared to the S_8 method are defined as $[MAX|(\varphi_{S8}-\varphi_{EP\text{-}SSn})|/\varphi_{S8}]$ and $MIN[|(\varphi_{S8}-\varphi_{EP\text{-}SSn})|/\varphi_{S8}]$ respectively.

Note that the EP-SS_8 method significantly improves the accuracy yielding a maximum relative difference in the scalar flux of 18.37% and 13.74% in energy groups 1 and 2, respectively.

Figures 6-20 and 6-21 show the relative difference between the EP-SS_8 and S_8 flux solutions in group 1. The front view results, Figure 6-20, indicate that the largest differences occur on the external surface of the model, where vacuum boundary conditions are specified; as expected the relative difference is larger in this region due to the approximate vacuum boundary conditions derived for the EP-SS_N method. The rear view results, shown in Figure 6-21, indicate a noticeable a larger relative difference on the material interface between the fuel region and the moderator due to higher order angular dependencies.

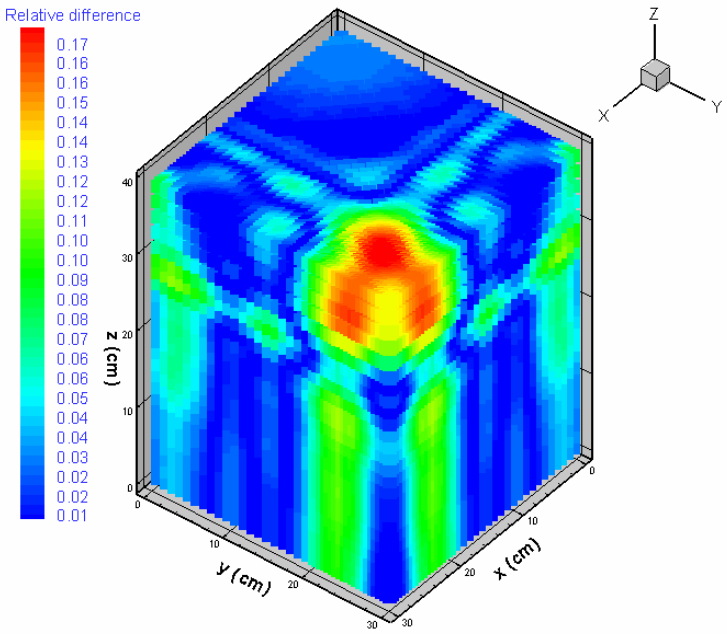

Figure 6-20. Front view of the relative difference between the scalar fluxes obtained with the EP-SS_8 and S_8 methods in energy group 1.

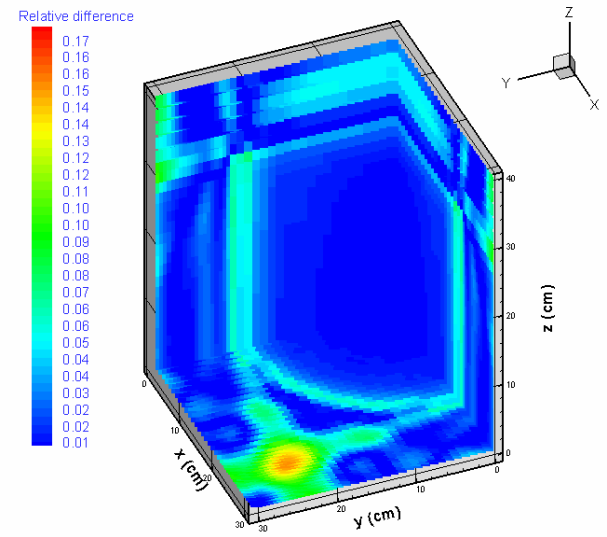

Figure 6-21. Rear view of the relative difference between the scalar fluxes obtained with the EP-SS_8 and S_8 methods in energy group 1.

6.4.6 Small Light Water Reactor (LWR) Criticality Benchmark Problem

A small LWR benchmark problem has been proposed by Takeda and Ikeda and it is one of the 3-D Neutron Transport Benchmarks by OECD/NEA.[43] The model represents the core of the Kyoto University Critical Assembly (KUCA) as shown in Figures 6-22 and 6-23.

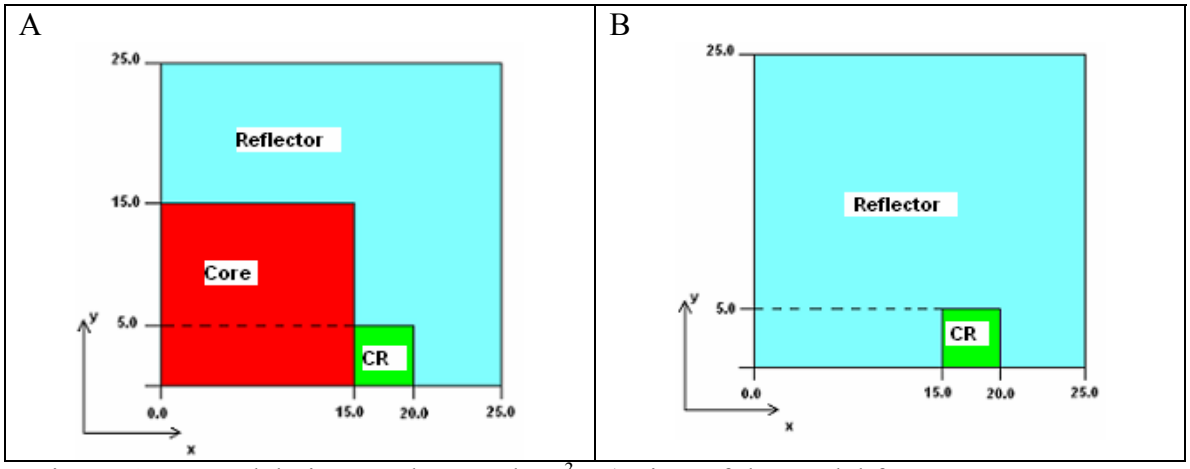

Figure 6-22. Model view on the *x*-*y* plane[3]. A) view of the model from *z*=0.0 cm to 15.0 cm, B) view of the model from *z*=15.0 cm to *z*=25.0 cm.

[3] CR is the abbreviation for Control Rod.

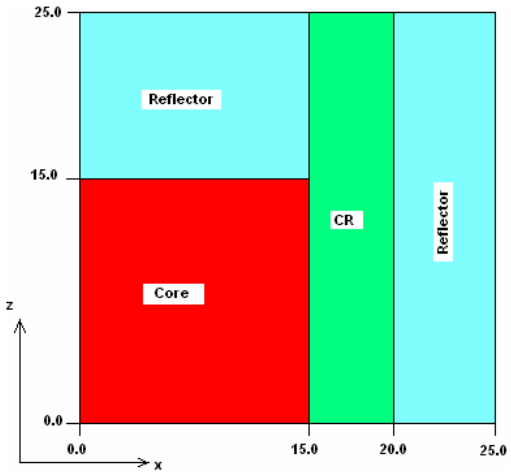

Figure 6-23. Model view on the *x*-*z* plane.

The model is discretized with a 1.0 cm uniform mesh. The core is polyethylene moderated and it consists of 93 w/o enriched U-Al alloy and natural uranium metal plates, with a moderation ratio of 1.5. The two-group cross-sections have been modified using the transport cross-section in place of the total cross-section in order to account for P_1 anisotropic scattering. The cross-sections are given in Table 6-12 and the fission spectrum along with energy range are given in Table 6-13.

Table 6-12. Two-group cross-sections for the small LWR problem.

Material	Group (g)	σ_a	$\nu\sigma_f$	σ_t
Core	1	8.52709e-03	9.09319e-03	2.23775e-01
	2	1.58196e-01	2.90183e-01	1.03864
Reflector	1	4.16392e-04	0.0	2.50367e-01
	2	2.02999e-02	0.0	1.64482
CR	1	1.74439e-02	0.0	8.52325e-02
	2	1.82224e-01	0.0	2.17460e-01
Void	1	4.65132e-05	0.0	1.28407e-02
	2	1.32890e-03	0.0	1.20676e-02

Table 6-12. Two-group cross-sections for the small LWR problem (Continued).

Material	Group (g)	$\sigma_{s,g\to1}$	$\sigma_{s,g\to2}$
Core	1	1.92423e-01	0.0
	2	8.80439e-01	2.28253e-02
Reflector	1	1.93446e-01	0.0
	2	1.62452	5.65042e-02
CR	1	6.77241e-02	0.0
	2	3.52358e-02	6.45461e-05

Table 6-12. Two-group cross-sections the small LWR problem (Continued).

Material	Group (g)	$\sigma_{s,g\to1}$	$\sigma_{s,g\to2}$
Void	1	1.27700e-02	0.0
	2	1.07387e-02	2.40997e-05

Table 6-13. Fission spectrum and energy range for the small LWR problem.

Group	Upper energy boundary (eV)	Lower energy boundary (eV)	Fission spectrum
1	10.0e7	6.8256e-01	1.0
2	6.8256e-01	1.0e-05	0.0

For this problem, two cases have been considered: in case 1, the control rod is withdrawn from the reactor and it is replaced with a void-like region; in case 2 the control rod is completely inserted into the core. This problem is particularly challenging due to the transport effects introduced by the control rod and the void-like region. For this problem, I have not modified the cross-sections with the density factor multiplier in order to show the limitations of the EP-SS_N method in dealing with this type of medium. Therefore, I have calculated the Control Rod Worth[44] (CRW) defined by

$$CRW = \left(\frac{1}{k_{eff}}\right)_{Case2} - \left(\frac{1}{k_{eff}}\right)_{Case1.} \tag{6.4}$$

The criticality eigenvalues (k_{eff}) calculated with different SS_N orders and the error relative to the Monte Carlo predictions are given in Table 6-14 for both cases.

Table 6-14. Criticality eigenvalues calculated with different EP-SS_N orders and relative error compared to Monte Carlo predictions.

	Case 1		Case 2	
Method	k_{eff}	Error (pcm)	k_{eff}	Error (pcm)
Monte Carlo	0.9790±0.0006	-	0.9624±0.0006	-
EP-SS_2	0.92325	-5598.2	0.9288	-3491.3
EP-SS_4	0.95266	-2591.0	0.95854	-401.1
EP-SS_6	0.95338	-2517.4	0.95931	-321.1
EP-SS_8	0.95341	-2514.3	0.95926	-326.3

The EP-SS_N method predicts relatively accurate values of the k_{eff} for case 2; however, the method under-predicts the criticality eigenvalue when the control rod is withdrawn in case 1. This behavior is due to the intrinsic limitations of the method in dealing with void-like regions. The CRWs estimated with different EP-SS_N methods and the Monte Carlo method, are given in Table 6-15.

Table 6-15. CRWs estimated with the EP-SS_N method.

Method	CRW
Monte Carlo	1.66e-2±0.09e-2
EP-SS_2	-6.47e-03
EP-SS_4	-6.44e-03
EP-SS_6	-6.48e-03
EP-SS_8	-6.40e-03

Based on the definition given in Eq. 6.4, a negative CRW would represent a positive insertion of reactivity by the control rod, which is clearly unphysical. Figure 6-24 shows the normalized scalar flux[4] for case 1, in energy group 1 along the x-axis at y=2.5 cm and z=7.5 cm (i.e., core mid-plane).

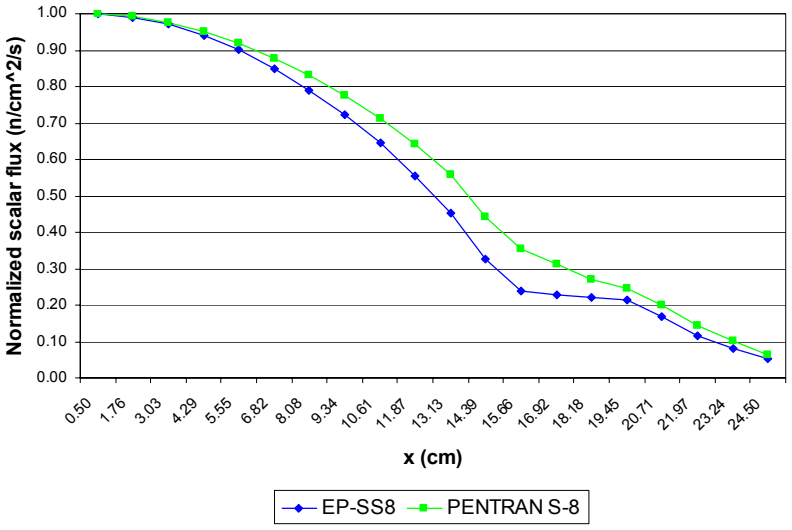

Figure 6-24. Normalized scalar flux for case 1, in group 1 along the x-axis at y=2.5 cm and z=7.5 cm.

[4] The scalar flux is normalized to the maximum value.

The EP-SS_8 method underestimates the flux distribution in the core region (0.0 to 15.0 cm) compared to the S_8 solution; this is due to an overestimation of the leakage term in the void-like region. Note that the integrated leakage term estimated by the EP-SS_8 method in the void-like region, is equal to 8.46280e-04 particles/sec while the S_8 method yields 6.61564e-05 particles/sec in the same region. Hence, due to the underestimation of the scalar flux in the core region, the criticality eigenvalues obtained in case 1 with the EP-SS_N method are also underestimated. Figures 6-25 show the EP-SS_8 predicted flux distributions for both cases in both energy groups.

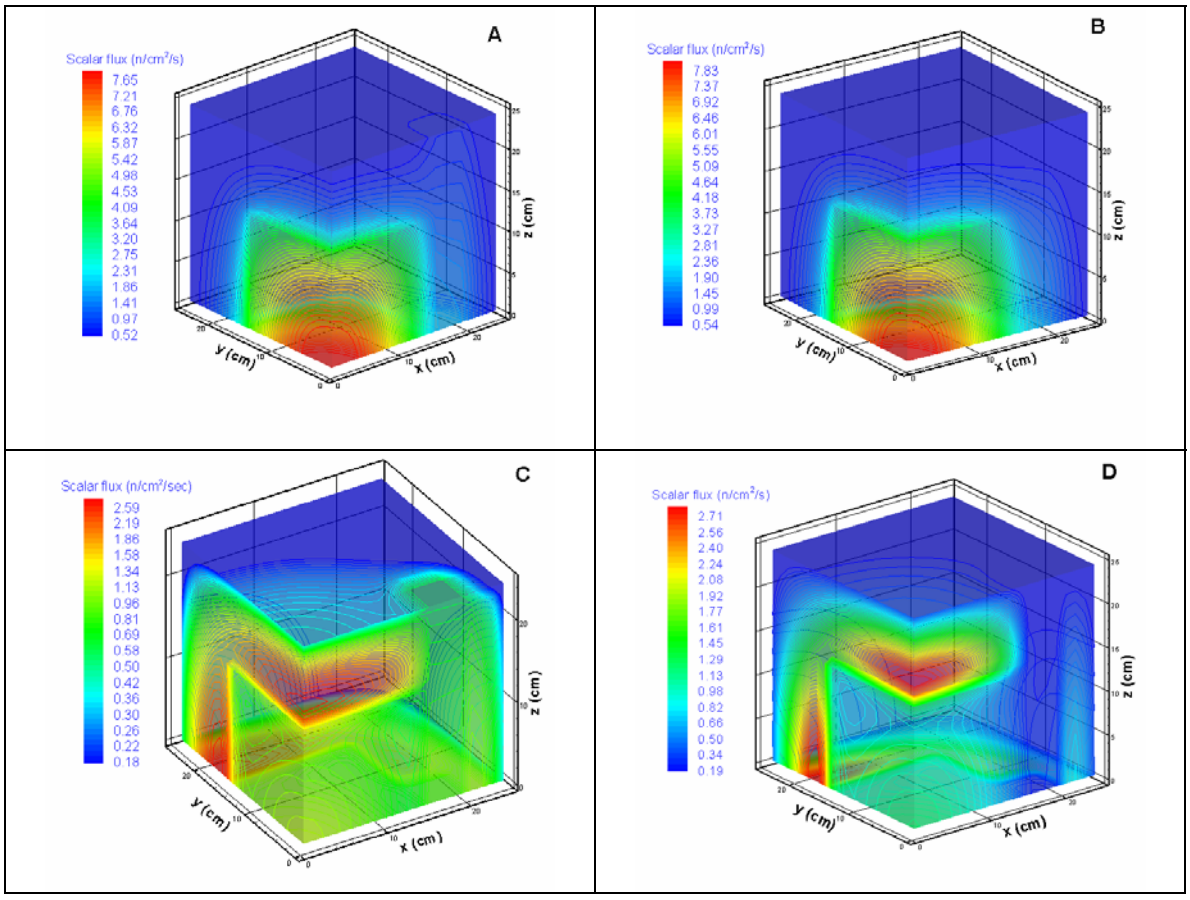

Figure 6-25. Scalar flux distributions. A) Case 1 energy group 1, B) Case 2 energy group 1, C) Case 1 energy group 2, D) Case 2 energy group 2.

Figure 6-25A and C clearly show the flat scalar flux distribution in the void-like region for both energy groups. In contrast, Figure 6-25D shows a pronounced flux depression in the control rod region in energy group 2. Note, that transport effects are significant in case 2, due to a steep flux gradient in the control rod region in the thermal range. However for this case, the EP-SS_N method (N>2) yields an accurate solution due to its higher order angular representation of the particle flux.

6.4.7 Small Fast Breeder Reactor (FBR) Criticality Benchmark Problem

The small FBR benchmark problem has also been proposed by Takeda and Ikeda and it is part of the OECD/NEA 3-D Neutron Transport Benchmarks. Views of the model on the *x*-*y* and *x*-*z* planes are shown in Figure 6-26 and 6-27 respectively.

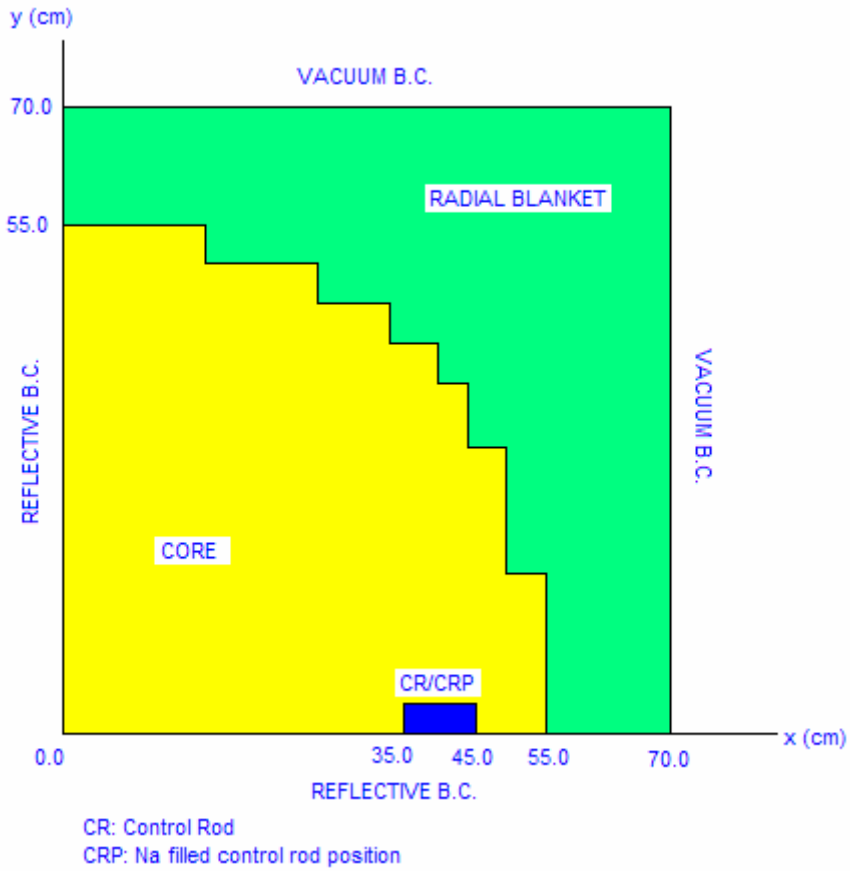

Figure 6-26. View on the *x*-*y* plane of the small FBR model.

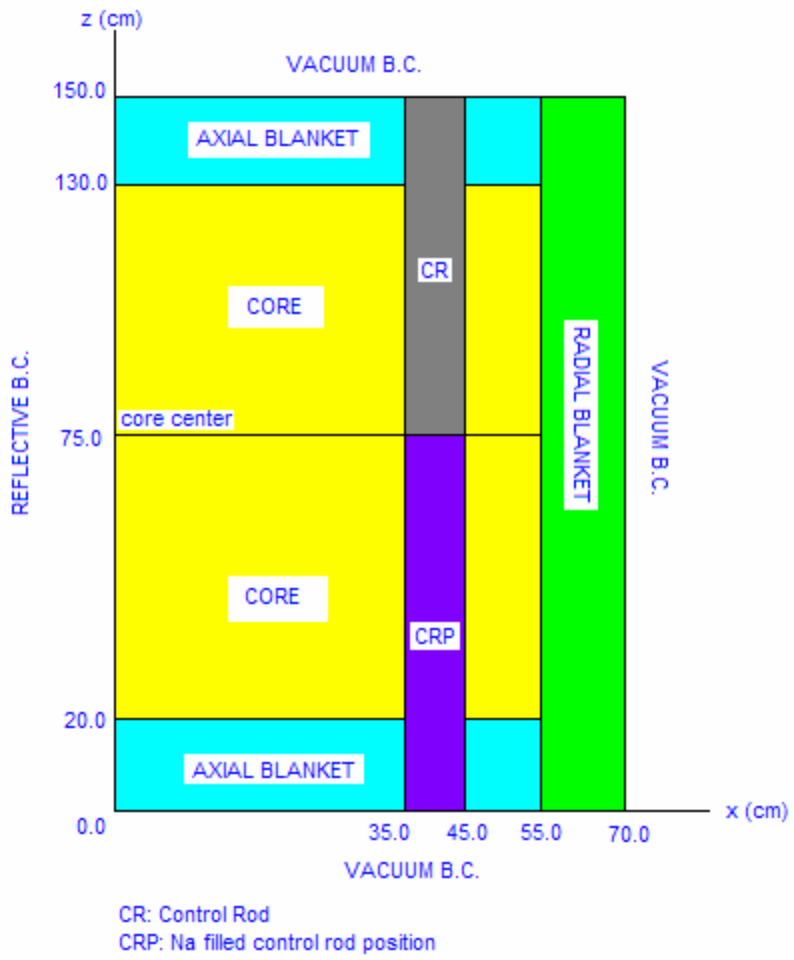

Figure 6-27. View on the *x*-*z* plane of the small FBR model.

For this benchmark two cases have been considered: in case 1 the control rod is fully withdrawn from the reactor and the channel is filled with sodium; in case 2 the control rod is half-inserted as shown in Figure 6-27. Note that the control rod introduces strong transport effects; however, no void-like regions are present for this problem, hence the EP-SS_N method is expected to yield relatively accurate solutions. The model is discretized with a 5 cm uniform mesh, which is the reference mesh size used in the benchmark; also, four-group cross-sections are used in these calculations.

The criticality eigenvalues (k_{eff}) obtained for this problem are given in Table 6-16, along with the relative error compared to the Monte Carlo reference solutions.

Table 6-16. Criticality eigenvalues for the small FBR model.

Case 1 - Control rod withdrawn			Case 2 - Control rod half-inserted		
Method	k_{eff}	Δk_{eff}[5] (pcm)	Method	k_{eff}	Δk_{eff}[d] (pcm)
Monte Carlo	0.9732±0.0002	-	Monte Carlo	0.9594±0.0002	-
EP-SS_2	0.96888	-443.90	EP-SS_2	0.95467	-493.02
EP-SS_4	0.97388	69.87	EP-SS_4	0.96017	80.26
EP-SS_6	0.97396	78.09	EP-SS_6	0.96024	87.55
EP-SS_8	0.97394	76.04	EP-SS_8	0.96026	89.64

Higher order EP-SS_N methods yield relatively accurate results compared to Monte Carlo; however, note that k_{eff} relative differences are higher for case 2. This can be attributed to the strong transport effects introduced by the control rod. The CRWs (Eq. 6.4) obtained with the EP-SS_N and Monte Carlo methods are given in Table 6-17.

Table 6-17. CRWs estimated with the EP-SS_N and Monte Carlo methods.

Method	CRW
Monte Carlo	1.47e-02
EP-SS_2	1.54e-02
EP-SS_4	1.47e-02
EP-SS_6	1.47e-02
EP-SS_8	1.46e-02

Except for the SS_2 order, all the other SS_N orders yield a very accurate CRW compared to the Monte Carlo solution. Figures 6-28 and 6-29 show the 3-D scalar flux distribution obtained with the EP-SS_8 method for both cases in energy groups 1 and 4, respectively. In Figure 6-28A and 6-29A the effect of the sodium channel is visible, especially in group 4, where neutron moderation occurs. In Figures 6-28B and 6-29B, the flux distortion due to the half-inserted control rod is clearly noticeable.

[5]The relative difference with Monte Carlo is calculated in pcm as Δk_{eff}=1.0e5*[k_{eff}(EP-SS_N)-k_{eff}(MC)]/k_{eff}(MC).

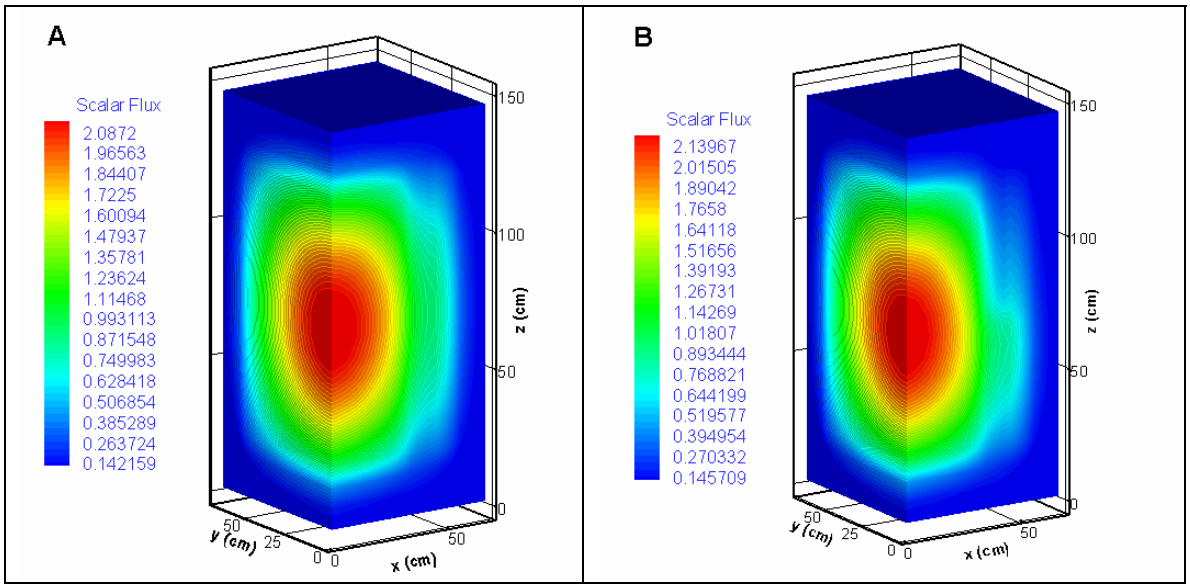

Figure 6-28. Scalar flux distribution in energy group 1: A) Case 1; B) Case 2.

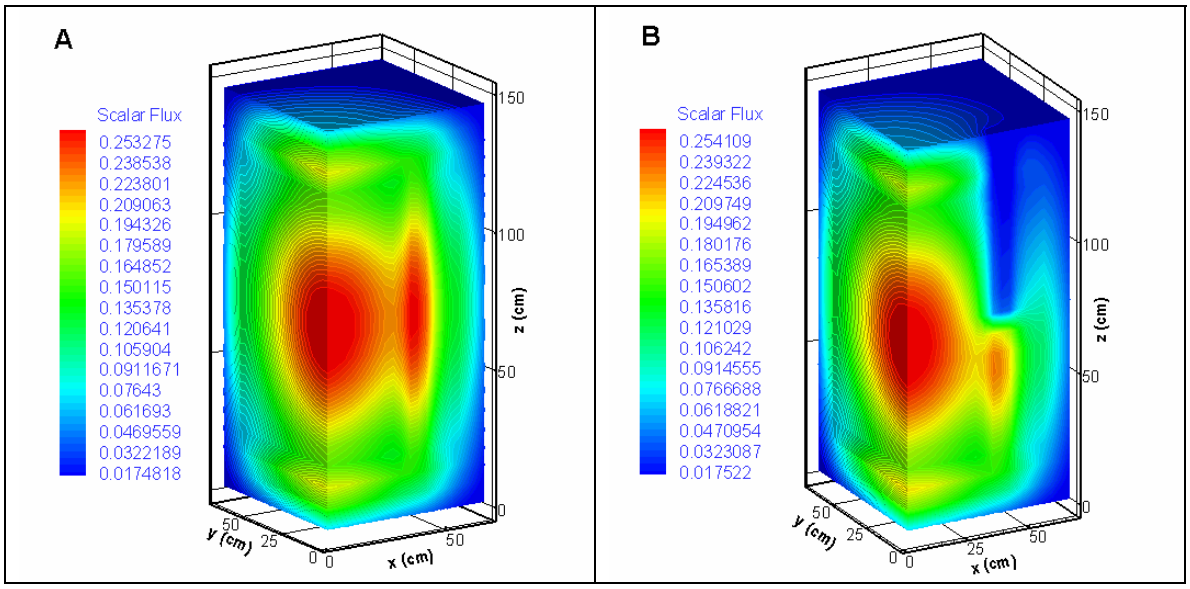

Figure 6-29. Scalar flux distribution in energy group 4: A) Case 1; B) Case 2.

6.4.8 The MOX 2-D Fuel Assembly Benchmark Problem

The MOX 2-D Fuel Assembly benchmark problem[44] has been proposed by NEA/OECD to test the current capabilities of radiation transport codes to perform whole-core calculations without spatial homogenization. For this benchmark both 2-D and 3-D versions of the problem were developed and accurate Monte Carlo solutions were obtained. The benchmark problem is the sixteen assembly (quarter core symmetry) C5 MOX fuel assembly problem proposed by Cavarec.[46] The 2-D mesh distribution is shown in Figure 6-30.

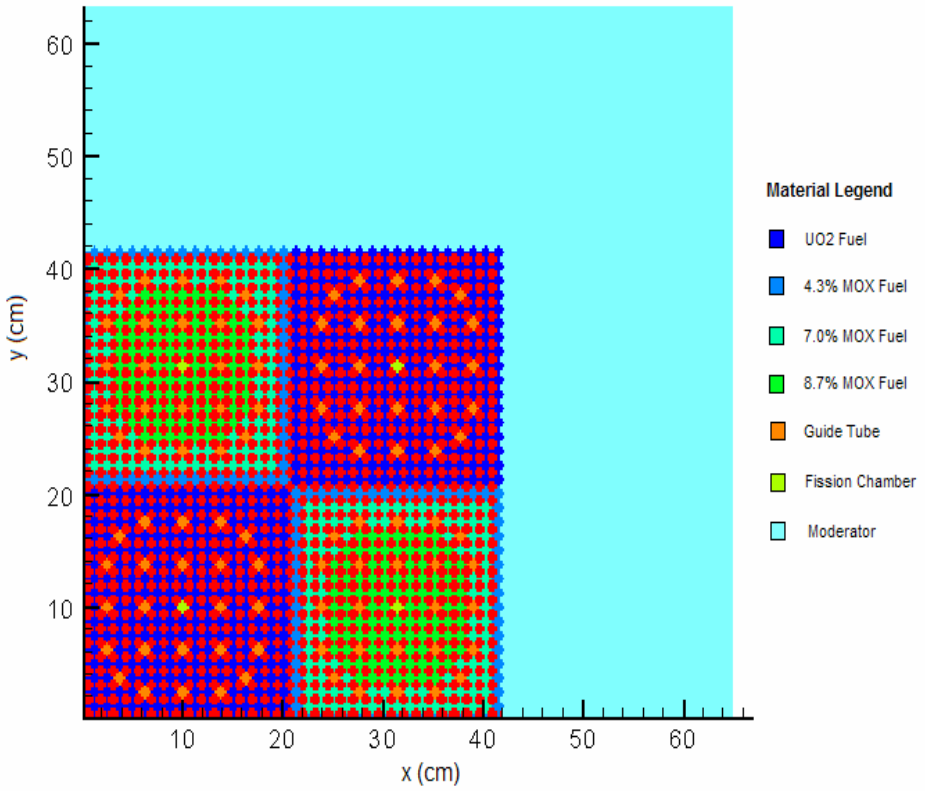

Figure 6-30. Mesh distribution of the MOX 2-D Fuel Assembly Benchmark problem.

The model consists of 81 coarse meshes, discretized with a total of 112,425 fine meshes. I have calculated the criticality eigenvalue and power distribution for this problem with the EP-SS_N method. Table 6-18 shows the criticality eigenvalues (k_{eff})

calculated with different EP-SS_N method and their relative error compared to the Refrence Monte Carlo solution.[44]

Table 6-18. Criticality eigenvalues and relative errors for the MOX 2-D benchmark problem.

Method	k_{eff}	Relative error (pcm)
Monte Carlo	1.86550	-
EP-SS_2	1.19335	573.0901
EP-SS_4	1.19017	305.0862
EP-SS_6	1.1907	349.7535

The EP-SS_4 method yields the most accurate solution in terms of the criticality eigenvalue. The increased accuracy obtained with the EP-SS_4 method compared to the diffusion method is due to the better representation of the transport effects due to heterogeneous regions with fuel-moderator interfaces. The accuracy obtained with the EP-SS_6 method slightly degrades due to the fact that the spatial mesh is not refined for increasing SS_N orders.

The power distribution, normalized over the number of fuel pins,[44] estimated for the inner UO_2 fuel assembly (see Figure 6-30) is 485.3, which differs by -1.5% compared to the MCNP reference solution (492.8±0.1%). For the MOX and the outer UO_2 fuel assemblies, I estimated a normalized power equal to 212.2 and 144.4, respectively. These results differ by ~0.3% and ~3.3% as compared to the Monte Carlo results (MOX: 211.7±0.18%, Outer UO_2: 139.8±0.20%), respectively. Note that the EP-SS_2 solution was obtained in 30 minutes running on 27 processors with spatial decomposition; the EP-SS_4 solution required 52.5 minutes on 18 processors with a hybrid domain decomposition (2-angle, 9-space), while the EP-SS_6 method took 86.3 minutes on 81 processors (3-angle, 27-space) . The EP-SS_2 and EP-SS_4 solutions were obtained on the PCPENII Cluster owned by the Nuclear & Radiological Department at the University of Florida. The EP-

SS_6 solution was obtained on the Zeta-Cluster (64 processors) and Kappa-Cluster (40 processors), part of the CARRIER Computational Lab Grid at the University of Florida. Figure 6-31 shows the scalar flux distribution for each energy group obtained with the EP-SS_4 method.

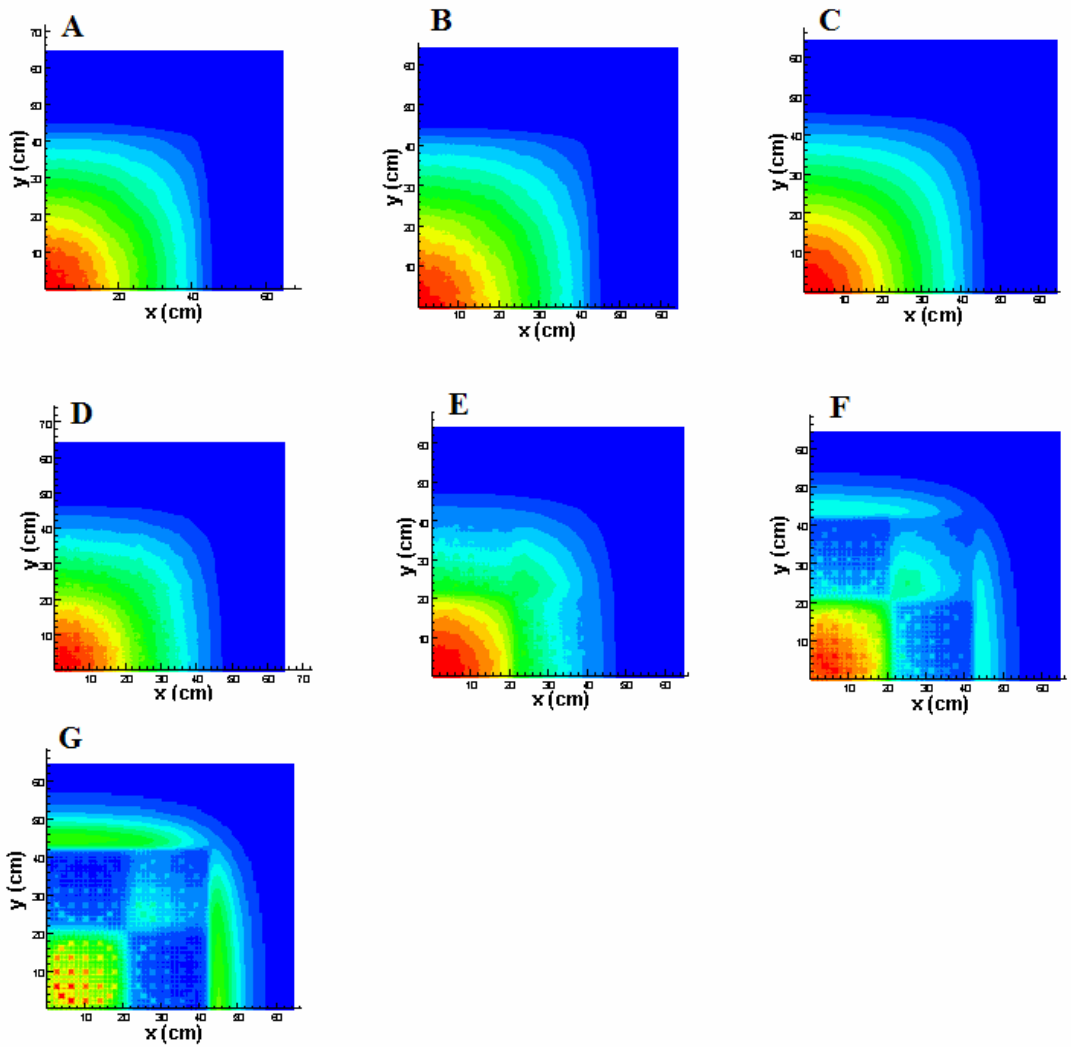

Figure 6-31. Scalar flux distribution for the 2-D MOX Fuel Assembly benchmark problem (EP-SS_4): A) Energy group 1; B) Energy group 2; C) Energy group 3; D) Energy group 4; E) Energy group 5; F) Energy group 6; G) Energy group 7.

Figure 6-32 shows the normalized pin power distribution obtained with the EP-SS_4 method.

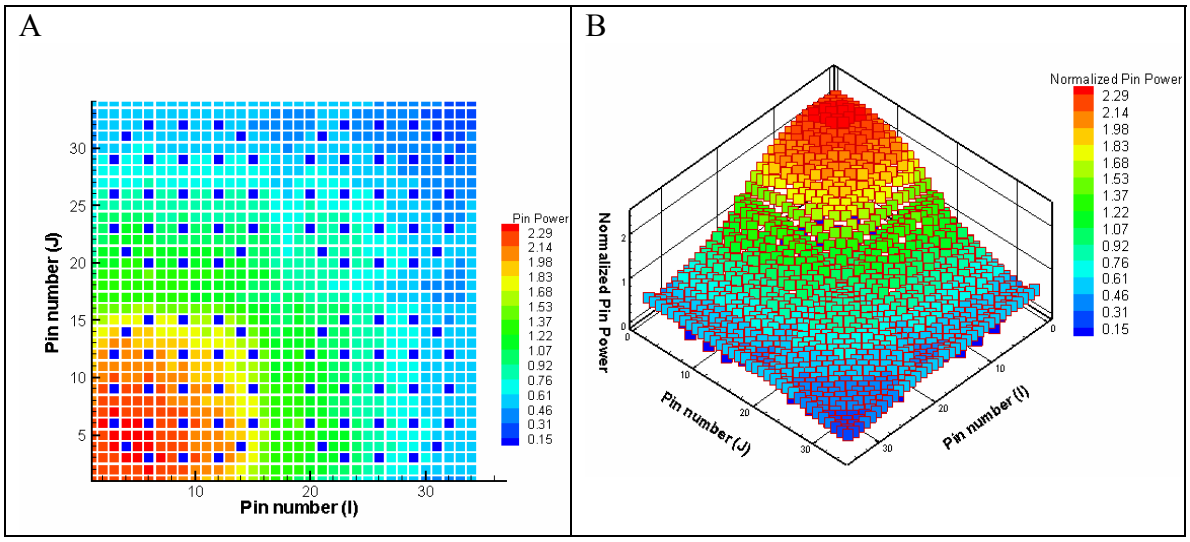

Figure 6-32. Normalized pin power distribution for the 2-D MOX Fuel Assembly benchmark problem (EP-SS_4): A) 2-D view; B) 3-D view.

CHAPTER 7
PARALLEL ALGORITHMS FOR SOLVING THE EP-SS_N EQUATIONS ON DISTRIBUTED MEMORY ARCHITECTURES

This chapter describes the parallel algorithms developed for the PENSSn code in distributed-memory architectures. I will describe the domain decomposition strategies developed, including spatial, angular and hybrid (spatial/angular) decompositions.

The parallel performance of PENSSn for a test problem, based on the speed-up, parallel efficiency and parallel fraction of the code is measured. Further, the parallel efficiency of the Krylov subspace based iterative solvers, and a methodology to improve their performance are discussed.

Finally, I will present the parallel performance obtained with PENSSn for the solution of the MOX 2-D Fuel Assembly Benchmark problem discussed in Chapter 6.

7.1 Parallel Algorithms for the PENSSn Code

PENSSn is designed to run on distributed memory architectures, where each processor is an independent unit with its own memory bank. This type of architecture is composed usually of PC-workstations linked together via a network backbone. The interconnection scheme among the processors is fundamental for distributed memory architectures because it affects, in part, the performance of the system. For cluster-type architectures, the processors are connected using a switch, which allows data transfer among the units.

For this type of system, the limited bandwidth available for processor intercommunication can be a limiting factor. Current network switches are capable of

1/10 GBit/sec bandwidth. Therefore, the parallel algorithm must minimize the communication time in order to yield an acceptable parallel performance.

PENSSn is written in Fortran-90 and it is parallelized with the MPI (Message Passing Interface) libraries.[27] This approach guarantees full portability of the code on a large number of platforms. The code solves the multigroup EP-SS_N equations with anisotropic scattering of arbitrary order for fixed source and criticality problems.

Three decomposition strategies have been implemented: spatial, angular and hybrid (spatial/angular) domain decompositions. The basic philosophy of this approach is to decompose part of the phase space on the processors, through a mapping function which defines the parallel virtual topology.

The mapping function or parallel vector, assigns portions of the domain to the processors; hence the calculation is performed locally by each processor on the allocated sub-domain. Note that on each processor only part of the domain is allocated in memory; this type of approach is defined as parallel memory, and it allows solving large problems which would be impossible to solve on a single workstation.

The main advantages of a parallel algorithm can be summarized in parallel tasking and memory partitioning. The first aspect relates to the computation time reduction achievable with a parallel computer; in an ideal situation, where no communication time is considered, *p* processors would solve the problem *p*-times faster than a single unit. In the remainder of this chapter, I will show that in practice, this level of performance is not achieved.

Memory partitioning allows the subdivision of the problem in RAM memory, hence, allowing the treatment of large simulation models. This aspect also eliminates the

need for scratch files on hard drives; the overall performance benefits from this aspect due to faster access of memory banks compared to hard drives.

7.2 Domain Decomposition Strategies

In order to parallelize the EP-SS_N equations, we partition the spatial domain into a number of coarse meshes and allocate them to different processors. Similarly, the angular domain is partitioned by allocating individual angles or groups of angles to each processor. The hybrid spatial and angular domain decomposition allows for simultaneous processing of spatial and angular sub-domains. Once the system is partitioned and the parallel vector is specified, the PENSSn code proceeds to sequentially allocate different sub-domains onto different processors, generating the so-called virtual topology.

7.2.1 Angular Domain Decomposition

The angular domain is partitioned based on a decomposition vector, which assigns the angles or group of angles to independent processors. Each processor locally solves angular fluxes for a subset of the total angular domain. After an inner iteration is completed, the moments of the even-parity angular flux are calculated using collective operations of the MPI library to minimize the communication overhead and to maintain data parallelism. In the PENSSn code, a subroutine is dedicated for the angular integration of the even-parity angular fluxes on the parallel environment, yielding total quantities such as scalar flux, currents, etc. The collective operation *MPI_ALLREDUCE* is used for this purpose;[27] note that when angular integration is performed, the values of the total quantities are also updated on each processor. Hence, this subroutine represents also a synchronization point.

7.2.2 Spatial Domain Decomposition

The spatial domain is partitioned into coarse meshes, as discussed in Chapter 5; each coarse mesh is then sequentially allocated to the processors through a decomposition vector. Every processor solves for the even-parity angular fluxes only on its assigned spatial sub-domain. The synchronization algorithm consists of a master/slave algorithm and a scheduling array, which contains information related to the allocation of the phase-space on every processor. The master/slave algorithm consists of a paired MPI_SEND/MPI_RECEIVE between two processors which share a coarse mesh interface. The scheduling array toggles each processor between send and receive modes, and it provides information on which portion of the phase-space has to be transferred. Note that before the sending processor initiates the communication phase, the projection algorithm, described in Chapter 5, is invoked. When every processor has updated the interface values on each coarse mesh, the calculation is continued. As for the angular decomposition algorithm, this point represents a synchronization phase.

7.2.3 Hybrid Domain Decomposition

The hybrid domain decomposition is a combination of spatial and angular decompositions. The hybrid decomposition takes advantage of both speed-up and memory partitioning offered by the angular and spatial decomposition, respectively. This decomposition strategy is based on the same algorithms described in the previous sections. Figure 7-1 shows an example of hybrid domain decomposition.

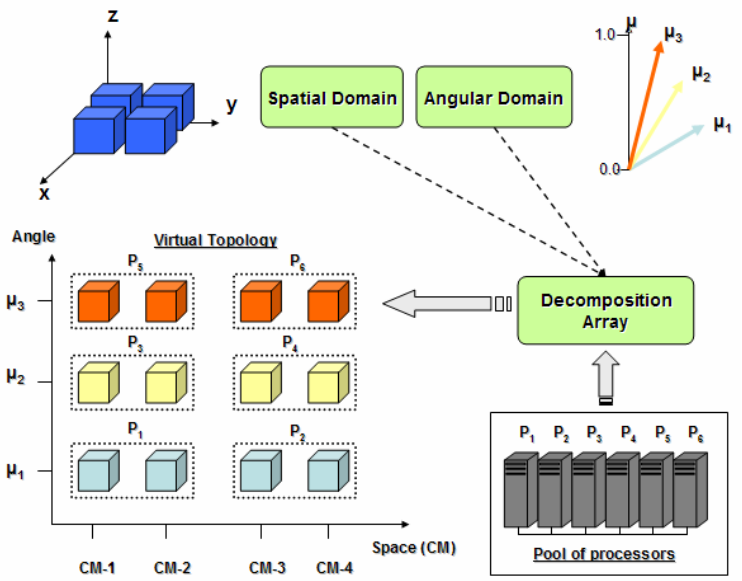

Figure 7-1. Hybrid decomposition for an EP-SS_6 calculation (3 directions) for a system partitioned with 4 coarse meshes on 6 processors.

7.3 Parallel Performance of the PENSSn Code

The parallel performance of PENSSn is assessed using a test problem composed of 64 coarse meshes; each coarse mesh is discretized with 4,000 fine meshes for a total of 256,000 fine meshes. The problem is characterized by a homogeneous material with one-group P_0 cross-sections; the total cross-section is equal to 1.0 cm^{-1}, while the scattering cross-section is equal to 0.5 cm^{-1}. A uniform distributed source is present in the system, emitting 1.0 particles/cm^3/sec. An SS_8 order is used for the calculations, which yields a total of 4 directions. Reflective boundary conditions are prescribed on boundary surfaces at x=0.0 cm, y=0.0 cm, z=0.0 cm, and vacuum boundary conditions are prescribed at x=24.0 cm, y=24.0 cm, z=24.0 cm. The convergence criterion for the angular flux is set to 1.0e-4, while it is set to 1.0e-6 for the Krylov solver.

Calculations have been performed on two different PC-Clusters: PCPENII at the Nuclear & Radiological Engineering Department and the Kappa Cluster at the Electrical

and Computer Engineering Department, part of the CARRIER Computational Lab Grid.

The specifications for the PCPENII Cluster are the following:

- 8 nodes (16 processors) Dual Intel Xeon processors with 2.4 GHz clock frequency, with hyper-threading
- 4 GB per node of DDR RAM memory on a 533 MHz system bus.
- 1 Gb/s full duplex Ethernet network architecture.
- 40 GB hard drives per each node.
- 512 KB L2 type cache memory for each processor.

The Kappa Cluster has the following technical specifications:

- 20 nodes (40 processors) Dual 2.4GHz Intel Xeon processors with 533MHz front-side bus with hyper-threading.
- Intel server motherboard with E7501 Chipset.
- On-board 1 Gb/s Ethernet.
- 1GB of Kingston Registered ECC DDR PC2100 (DDR266) RAM.
- 40GB IDE drive @ 7200 RPM.

The analysis of the parallel performance of PENSSn is based on the definition of speed-up, parallel efficiency and parallel fraction. The speed-up is the direct measure of the time reduction obtained due to parallel tasking; the mathematical definition of speed-up is given by

$$S_p = \frac{T_s}{T_p}, \tag{7.1}$$

where p is the number of processors, T_s is the wall-clock time for the serial run and T_p is the wall-clock time for the parallel run on p processors.

The parallel efficiency measures the performance of the domain decomposition algorithm. The definition of parallel efficiency is given by

$$\eta_p = \frac{S_p}{p}. \tag{7.2}$$

The speed-up and parallel efficiency are affected by communication time and idle time for each processor, by load-imbalance, and by the parallel fraction in the Amdahl's law.

Finally, using the Amdahl's law for expressing the theoretical speed-up, we can estimate the parallel fraction.

$$S_p = \frac{1}{(1-f_p) + \frac{f_p}{p} + \frac{T_c}{T_s}}, \tag{7.3}$$

where f_p is the parallelizable fraction of the code running on p processors and T_c is the parallel communication time. Eventually all these quantities are affected by the *load-imbalance*, which may be caused by the different amount of workload. Figure 7-2 shows the speed-up obtained for different decomposition strategies on the two PC Clusters.

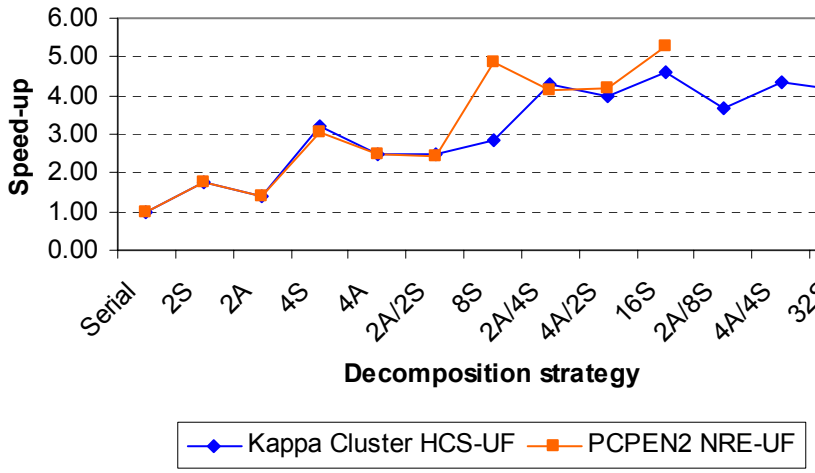

Figure 7-2. Speed-up obtained by running PENSSn on the Kappa and PCPENII Clusters.

In Figure 7-2, the "decomposition strategy" refers to the number of processors and the type of decomposition used; "S" refers to spatial decomposition and "A" refers to angular decomposition, and "/" identifies hybrid decompositions. Except for the 8-spatial domain decomposition, the speed-up is comparable for the two clusters up to 4 processors. The maximum speed-up achieved is 5.27 and 4.62, for the PCPENII and Kappa Cluster, respectively, for a spatial-decomposition strategy on 16 processors. Note that as the number of processors increases, the speed-up obtained does not increase as

well. This behavior is directly related to the concept of *granularity* and to the communication time. The granularity represents the amount of work-load available to each processor; a large grain size leads to a more efficient usage of the machines. In contrast, a small grain size leads to a large communication overhead and, hence, to lower parallel efficiencies. By increasing the number of processors for a fixed problem size, we effectively reduce the granularity with subsequent degradation of the speed-up and parallel efficiency as shown in Figure 7.3.

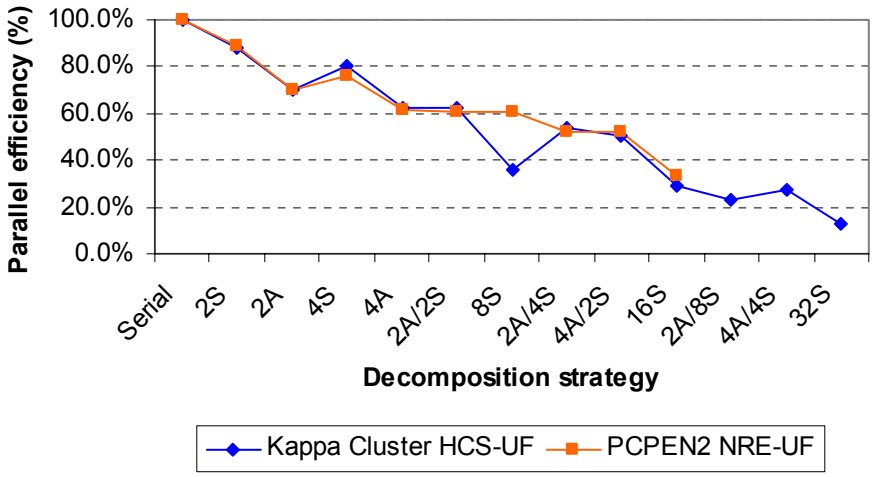

Figure 7-3. Parallel efficiency obtained by running PENSSn on the Kappa and PCPENII Clusters.

The spatial discretization of the test problem does not introduce any load imbalance per se; however, Figure 7-3 shows a difference in terms of parallel efficiency between the angular- and spatial-decomposition strategies on the same number of processors. This difference is due to load imbalance introduced by the Krylov iterative solver. Table 7-1 presents the data supporting the load imbalance generated by the Krylov solver.

Table 7-1. Data relative to the load imbalance generated by the Krylov solver.

Decomposition	Processor	Direction	Krylov iterations	Communication + Idle Time (sec)	Grain size
2A/1S	1	μ_1	9749	83.769	28.5%
		μ_2	32796		
	2	μ_3	48877	3.967	71.5%
		μ_4	57804		

The current strategy for partitioning the phase-space on each processor is based on a sequential allocation of sub-domains. However, processor 1 is characterized by a smaller grain size compared to processor 2; therefore, the idle time of processor 1 is much larger than processor 2. This is a clear example of load imbalance and its effect is observed in the relatively low speed-up of 1.41 and parallel efficiency of 70.3%. This behavior is not observed for the spatial decomposition where the entire angular domain is locally stored on each processor.

Table 7-1 shows also a measure of the grain size as the ratio between the total number of Krylov iterations required by each processor and the total number of Krylov iteration required by all the processors. Note that the load imbalance is clearly shown by the grain size calculated on each processor that is 28.5% and 71.5% on processors 1 and 2, respectively.

Theoretically, the load imbalance due to the Krylov solvers could be overcome by adopting an automatic load balancing algorithm. As discussed in Chapter 6, the spectral condition number for the matrix operators increases as the direction cosine approaches unity. Based on these results, the sequential allocation of the angular sub-domains is not optimal. The best angular decomposition algorithm is based on progressively pairing the directions which yield the highest and lowest spectral condition numbers.

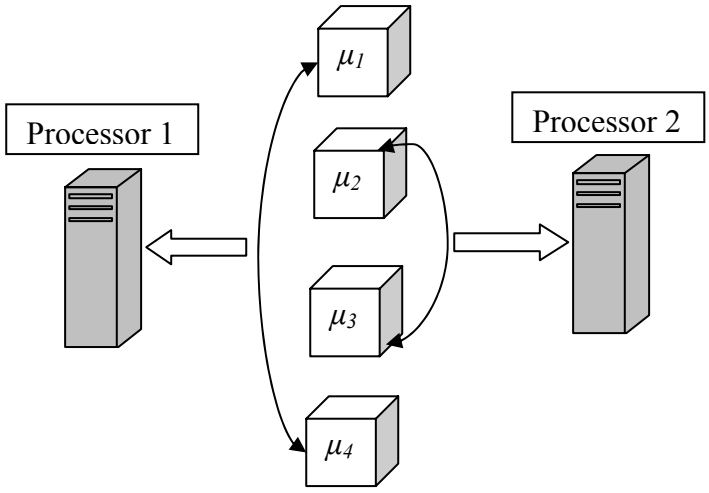

Figure 7-4. Angular domain decomposition based on the automatic load balancing algorithm.

Figure 7-4 shows the angular domain partitioning based on the automatic load balancing algorithm. If this algorithm is applied, the grain size would change to 45.3% and 54.7% for processor 1 and 2, respectively.

To complete the analysis of the parallel performance of PENSSn, I have calculated the parallel fraction of the code, by using the Amdahl's law in Eq. 7.3. Figure 7-5 shows the speed-up obtained on the PCPENII cluster compared to the theoretical speed-up predicted by the Amdhal's Law when the communication time is neglected.

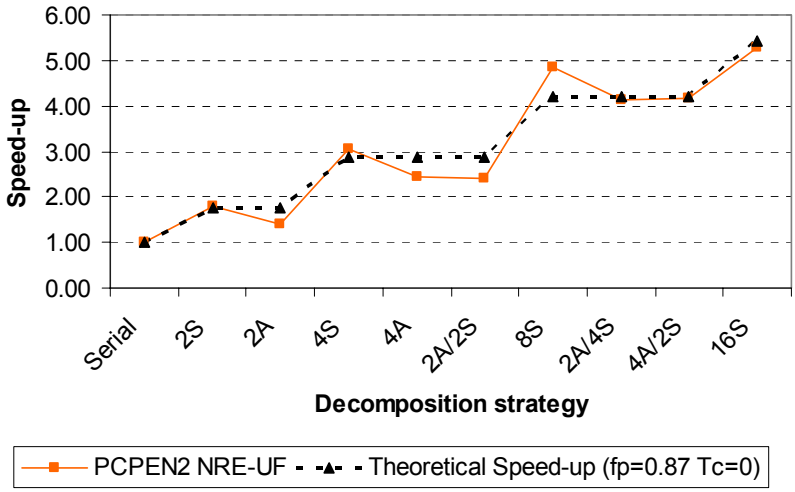

Figure 7-5. Parallel fraction obtained with the PENSSn code.

I obtained a 87% parallel fraction (f_p) with a maximum relative difference of 19% between the theoretical speed-up prediceted by the Amdahl's law and the speed-up observed on PCPENII. Tables 7-2 and 7-3 summarize the supporting data associated with the PCPENII and Kappa Clusters respectively.

Table 7-2. Parallel performance data obtained on PCPENII Cluster.

DD	# of processors	Parallel Vector	Wall-clock time (sec)	Speed-up	Efficiency	Memory MB/proc	Parallel Fraction
Serial	1	1A/1S	207.4	-	-	158.8	-
S	2	1A/2S	116.24	1.78	89.2%	73.2	88.0%
A	2	2A/1S	147.58	1.41	70.3%	79.9	61.6%
S	4	1A/4S	67.95	3.05	76.3%	43.1	89.8%
A	4	4A/1S	84.28	2.46	61.5%	53.2	84.1%
H	4	2A/2S	86.04	2.41	60.3%	46.4	78.4%
S	8	1A/8S	42.68	4.86	60.7%	28	90.8%
H	8	2A/4S	50.09	4.14	51.8%	29.7	86.9%
H	8	4A/2S	49.78	4.17	52.1%	33.1	87.1%
S	16	1A/16S	39.32	5.27	33.0%	20.5	86.5%

DD stands for Domain Decomposition (A – Angular, H – Hybrid, S – Spatial).

Table 7-3. Parallel performance data obtained on Kappa Cluster.

DD	# of processors	Parallel Vector	Wall-clock time (sec)	Speed-up	Efficiency	Memory MB/proc	Parallel Fraction
S	2	1A/2S	116.14	1.76	88.1%	73.2	86.9%
A	2	2A/1S	145.58	1.41	70.3%	79.9	61.7%
S	4	1A/4S	63.55	3.22	80.5%	43.1	92.1%
A	4	4A/1S	82.11	2.49	62.3%	53.2	83.6%
H	4	2A/2S	82.2	2.49	62.3%	46.4	80.2%
S	8	1A/8S	71.9	2.85	35.6%	28	74.3%
H	8	2A/4S	47.45	4.31	53.9%	29.7	88.0%
H	8	4A/2S	51.08	4.01	50.1%	33.1	86.0%
S	16	1A/16S	44.32	4.62	28.9%	20.5	83.6%
H	16	2A/8S	55.42	3.69	23.1%	21.4	77.9%
H	16	4A/4S	47.25	4.33	27.1%	23	82.3%
S	32	1A/32S	49.29	4.15	13.0%	16.8	78.4%
S	64	1A/64S	54.19	3.78	5.9%	14.9	74.8%

These results indicate that the PENSSn code is characterized by a relatively high parallel performance; also the PCPENII and Kappa Clusters yield almost the same performance. The main advantage of the PCPENII Cluster is the large amount of memory

available, which allows the simulation of large 3-D models without compromising the parallel performance due to communication overhead.

7.4 Parallel Performance of PENSSn Applied to the MOX 2-D Fuel Assembly Benchmark Problem

The MOX 2-D benchmark problem discussed in Chapter 6 is used to assess the parallel performance of PENSSn for a large criticality eigenvalue problem. The problem specifications are discussed in Chapter 6, and an SS_4 order is used for the performance testing. Table 7-4 presents the parallel performance data obtained on the PCPENII Cluster.

Table 7-4. Parallel performance data for the 2-D MOX Fuel Assembly Benchmark problem (PCPENII Cluster).

Decomposition Strategy	Number of processors	Parallel Vector	Wall-clock time (sec)	Speed-up	Efficiency	Memory MB/proc
Serial	1	1A/1S	511.0	1.0	-	341.6
Angular	2	2A/1S	263.7	1.9	95.0%	202.0
Spatial	3	1A/3S	198.7	2.6	86.6%	105.3
Hybrid	6	2A/3S	132.6	3.9	65.0%	71.7
Spatial	9	1A/9S	107.3	4.8	53.3%	39.5

Note that for this problem the angular decomposition yields the best speed-up, because for the low order SS_N methods the load-imbalance due to the Krylov solver is not so significant. For this problem, high order EP-SS_N methods provided a relatively accurate solution, both in terms of criticality eigenvalue and power distribution. As indicated, the hybrid decomposition is used mainly to increase the speed-up. Moreover, the PENSSn code yielded good parallel performance both in terms of speed-up and memory utilization.

CHAPTER 8
DEVELOPMENT OF A NEW SYNTHETIC ACCELERATION METHOD BASED ON THE EP-SS_N EQUATIONS

The inversion of the transport operator is obtained using the Source Iteration or Richardson iteration method. It is known that this iterative technique is very ineffective for problems with optically thick regions and scattering ratio close to unity. In these conditions, the spectral radius of the transport operator tends to unity and the convergence process becomes very slow.[3] Hence, it is necessary to develop acceleration schemes which can increase the rate of convergence.[17] In principle, synthetic acceleration schemes consist of two distinct operators: a higher order operator (e.g., S_N) and a lower order operator, usually a diffusion-like equation. The idea is to correct the solution of the diffusion-like equation using the transport solution, with subsequent acceleration of the convergence process.

In the late 1960s, Gelbard and Hageman developed a synthetic acceleration method based on the diffusion equation and the S_4 equations.[28] Later, Reed independently derived a similar synthetic acceleration scheme[40] and pointed out some limitations of the method derived by Gelbard and Hageman. The synthetic method developed by Reed has the advantage of being very effective for small mesh sizes, but it is unstable for mesh size greater than ~1 mfp. Later, Alcouffe independently derived the Diffusion Synthetic Acceleration (DSA) method. He addressed the issue of stability of the method and derived an unconditionally stable DSA algorithm.[29] Alcouffe pointed out that in order to obtain an unconditionally stable method, the diffusion equation must be derived from the

discretized version of the transport equation. In this way, the consistency between the two operators is preserved. However, Warsa, Wareing and Morel, recently observed a loss in the effectiveness of DSA schemes, especially for multi-dimensional heterogeneous problems.[31-32]

This chapter addresses the development of a new synthetic acceleration method based on the EP-SS_N method. Since the discretization of the EP-SS_N equations is not consistent with the discretization of the S_N method, the acceleration method is limited to mesh sizes up to ~1 mean free path. The first part of this chapter discusses the theory involved in developing the EP-SS_N synthetic acceleration method, and the second part presents the numerical results obtained for a test problem. The performance of the EP-SS_N synthetic acceleration algorithm will be compared with the Simplified Angular Multigrid (SAM) acceleration method.[39] In conclusion, I will point out strengths and weaknesses of the new method, and I will build the foundations for the FAST (Flux Acceleration Simplified Transport) preconditioning algorithm, discussed in Chapter 9.

8.1 The EP-SS_N Synthetic Acceleration Method

This section describes the general theory of a general synthetic acceleration method and, hence, its application to the solution of the S_N equations.

Solutions to many engineering problems of practical interest can be obtained via a balance equation written in operator form as follows

$$Tf = Kf + q\,. \tag{8.1}$$

In the case of the S_N equations, the operators T and K are defined as

$$T = \hat{\Omega}\cdot\vec{\nabla} + \sigma_t\,, \tag{8.2}$$

$$K = \int_0^\infty dE \int_{4\pi} d\Omega \sigma_s(\vec{r}, E' \to E, \hat{\Omega}'\cdot\hat{\Omega})\,. \tag{8.3}$$

Commonly, Eq. 8.1 is solved using the Richardson iterative method formulated as

$$Tf^{l+1} = Kf^{l} + q\,. \tag{8.4}$$

In Eq. 8.4, l is the iteration index; hence, by inverting the operator T in Eq. 8.4, we obtain the following

$$f^{l+1} = Mf^{l} + T^{-1}q\,, \tag{8.5}$$

where

$$M = T^{-1}K\,. \tag{8.6}$$

In Eq. 8.5 the operator M is usually referred to as *iteration operator*. In this derivation the spectral radius of the iteration operator is assumed to be less than unity; note that this is a realistic assumption for the problems encountered in engineering applications. To discuss the synthetic operator, I introduce the residual term given by

$$r^{l+1} = f^{l+1} - f^{l}\,. \tag{8.7}$$

Using the residual formulation, Eq. 8.5 reduces to

$$r^{i+1} = Mr^{i}\,. \tag{8.8}$$

The sum over the residuals following the l^{th} iteration is given by

$$\sum_{k=1}^{\infty} r^{l+k} = (M + M^{2} + ...)r^{l} = \frac{M}{I - M}r^{l}\,. \tag{8.9}$$

Where, I is the identity matrix. Hence, using the above formulation, the exact solution to Eq. 8.5 is obtained by

$$f = f^{l} + \sum_{k=1}^{\infty} r^{l+k}\,, \tag{8.10}$$

or

$$f = f^{l} + \frac{M}{I - M}r^{l}\,. \tag{8.11}$$

Note that using the definition of the iteration operator M (Eq. 8.6), Eq. 8.11 reduces to

$$f = f^l + (T - K)^{-1} K r^l . \tag{8.12}$$

Eq. 8.12 is just another formulation for Eq. 8.1 and its solution is as difficult. However, Eq. 8.12 separates the solution to Eq. 8.1 into two terms. The synthetic acceleration method is based on approximating the high order operator (e.g., transport) $(T - K)$ with a lower order operator W_L. The lower order operator must possess two fundamental properties in order for the synthetic method to be effective: 1) the lower order operator has to be a good approximation to the high order operator; and 2) it has to be easy to invert. The synthetic acceleration method formulated based on Eq. 8.12 can be written as

$$f^{l+1} = \tilde{f}^{l+1/2} + W_L^{-1} K r^l , \tag{8.13}$$

where

$$\tilde{f}^{l+1/2} = M f^l + T^{-1} q , \tag{8.13}$$

and

$$r^l = \tilde{f}^{l+1/2} - f^l . \tag{8.14}$$

The philosophy of the synthetic acceleration method described above consists in utilizing a lower order operator to project the residual term on the sub-space generated by the operator K. The projection operation with the low order operator is performed in a fraction of the time required for the solution of the higher order operator, thereby producing a significant speed-up of the iteration process with a consequent reduction of the numerical spectral radius.

The synthetic acceleration method discussed above can be readily applied to the S_N equations. The method is designed to accelerate the convergence of the inner iteration in

each energy group. In the following derivation we assume isotropic scattering and source for simplicity. In this particular case, the high order operator is the transport equation.

$$H\widetilde{\psi}_n^{l+1/2} = K_{S0}\phi^l + q\,, \tag{8.15}$$

where

$$H_n = \hat{\Omega}_n \cdot \vec{\nabla} + \sigma_t\,, \tag{8.16}$$

and

$$K_{S0} = \sigma_{s0}\,. \tag{8.17}$$

In the SI method, the scalar flux (ϕ^0) is initially guessed and substituted into Eq. 8.15; hence, Eq. 8.15 is solved and the value of the scalar flux at the new iteration is simply evaluated by

$$\phi^{l+1} = \sum_{n=1}^{N(N+2)} w_n \widetilde{\psi}_n^{l+1/2}\,, \tag{8.18}$$

and the iteration process is continued until convergence. The synthetic acceleration method described above substitutes the update of the scalar flux in Eq. 8.18, with the following expression

$$\phi^{l+1} = \widetilde{\phi}^{l+1/2} + p^l\,. \tag{8.19}$$

Note that the term indicated with the tilda symbol in Eq. 8.19 is obtained from the S_N method in Eq. 8.15. The projection of the residual term (p^l) on the scattering kernel is performed using the EP-SS_N equations as follows

$$\widetilde{p}_m^l = W_{L,m}^{-1} K_{S0} r^l\,. \tag{8.20}$$

The residual in Eq. 8.20 is calculated as the difference between the transport solution and the accelerated solution for the scalar flux

$$r^l = \widetilde{\phi}^{\,l+1/2} - \phi^l , \tag{8.21}$$

where

$$W_{L,m} = -\vec{\nabla} \cdot \left(\frac{\mu_m^2}{\sigma_t} \right) \vec{\nabla} + \sigma_t , \tag{8.22}$$

and

$$p^l = \sum_{m=1}^{(L+1)/2} w_m \widetilde{p}_{e,m}^{\,l} . \tag{8.23}$$

The S_N equations given in Eq. 8.15 are used to compute an uncollided scalar flux, which is then used to evaluate the residual with Eq. 8.21. The residual is then projected onto the scattering kernel via the EP-SS_N operator. Hence, the inversion of the EP-SS_N operator generates a projection of the residual onto the scattering kernel, which is then used to complete the calculation of the scalar flux at the new iteration using Eq. 8.19.

8.2 Spectral Analysis of the EP-SS_N Synthetic Acceleration Method

In this section, I will study the theoretical performance of the synthetic acceleration method based on the EP-SS_N equations. For this purpose I will analyze the spectrum of eigenvalues of the synthetic operator in the Fourier transformed space. The performance of iterative methods can be assessed by studying the spectral radius. For an infinite homogeneous medium, the spectral radius is equal to

$$\rho_\Lambda \equiv \underset{\omega}{MAX} \left| \Lambda(\omega) \right| , \tag{8.24}$$

where $\Lambda(\omega)$ represents the spectrum of eigenvalues in the transformed space. In general, the spectral radius is an indication of the error reduction for the iterative method at each iteration. Hence, if the spectral radius approaches unity the method presents very slow convergence.

The spectrum of eigenvalues for the EP-SS_N synthetic acceleration algorithm can be written as

$$\Lambda_N(\omega) = \frac{A(\omega) - A_N(\omega)}{1 - A_N(\omega)}, \tag{8.25}$$

where the transport operator in the transformed space is formulated as

$$A(\omega) = c\int_0^1 \frac{d\mu}{1+\mu^2\omega^2} = \frac{c}{\omega}\arctan(\omega), \tag{8.26}$$

and the lower order EP-SS_N operator is defined by

$$A_N(\omega) = c\sum_{\alpha=1}^{N} \frac{w_\alpha}{1+\mu_\alpha^2\omega^2}. \tag{8.27}$$

It is well known that the spectral radius of the unaccelerated transport operator in an infinite homogeneous medium is equal to

$$\rho_A \equiv \underset{\omega}{MAX}\left|A(\omega)\right| = \lim_{\omega\to 0}\frac{c}{\omega}\arctan(\omega) = c. \tag{8.28}$$

Therefore, as discussed in the beginning of this chapter, the unaccelerated source iteration method presents very slow convergence for scattering-dominated problems, and where the leakage probability is relatively small. Figure 8-1 shows the spectrum of eigenvalues for the unaccelerated source iteration and for the synthetic acceleration methods obtained with different SS_N orders.

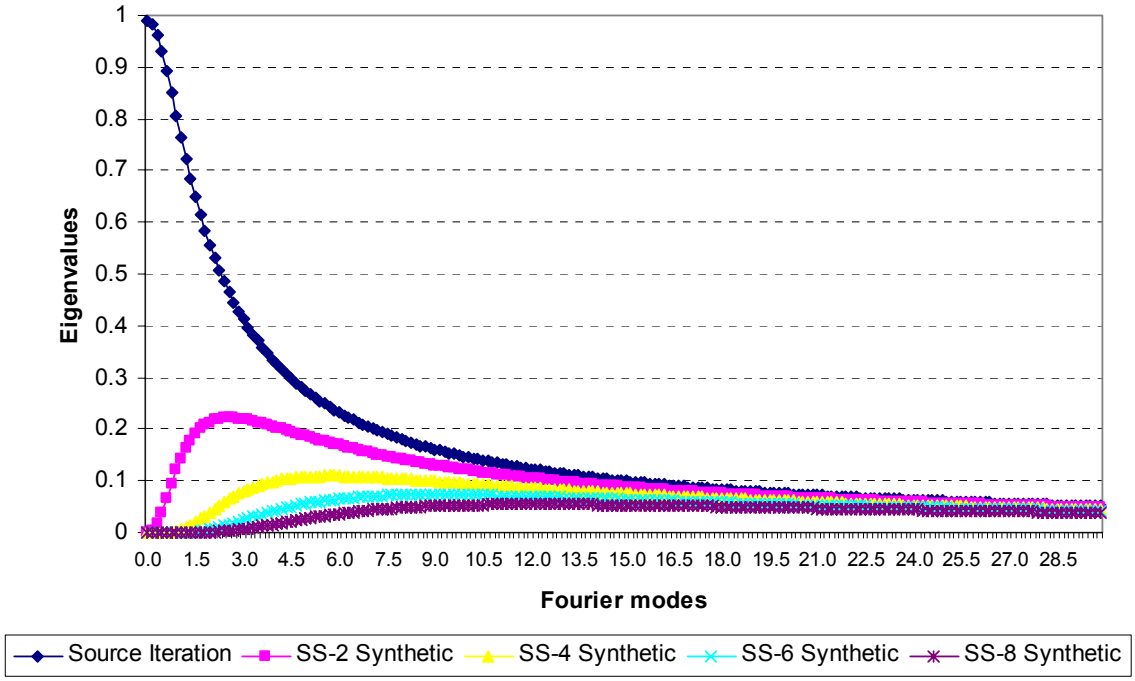

Figure 8-1. Spectrum of eigenvalues for the Source Iteration and Synthetic Methods based on different SS_N orders.

As indicated by Figure 8-1, the spectral radius of different EP-SS_N synthetic methods decreases with increasing order; this behavior is due to the fact that higher order EP-SSn methods resolve higher frequency modes of the spectrum, which are characterized by higher order angular dependencies. Also, the spectral radius obtained for the different methods are listed in Table 8-1.

Table 8-1. Spectral radius for the different iterative methods.

Method	Spectral Radius
Source Iteration	0.99
SS_2 Synthetic	0.221391
SS_4 Synthetic	0.109545
SS_6 Synthetic	0.072785
SS_8 Synthetic	0.054489

Based on these results, higher order EP-SS_N equations present significantly smaller spectral radii than diffusion based synthetic acceleration algorithms and, therefore, better

acceleration performance. However, the numerical results will show that in practice theoretical performance is not achieved.

8.3 Analysis of the Algorithm Stability Based on Spatial Mesh Size

In this section, I will analyze the stability of the EP-SS_N synthetic acceleration method with respect to the spatial mesh size. In this phase of the investigation, the discretization of the EP-SS_N formulation is not consistent with the transport operator; hence, the stability of the method depends on the size of the spatial mesh. The EP-SS_N acceleration method has been implemented into the PENTRAN Code System.[15]

For this analysis, I have considered a simple 3-D cube with a homogeneous material. The size of the cube is 10x10x10 cm^3, discretized with a 1.0 cm uniform mesh along the three axes. The total cross-section is varied in order to change the dimension of the system in terms of mean free paths (*mfp*), and the c-ratio is set equal to 0.99. The boundary conditions prescribed are reflective on the planes at x=0.0 cm, y=0.0 cm, z=0.0 cm and vacuum at x=10.0 cm, y=10.0 cm and z=10.0 cm. An isotropic source, with magnitude 1.0 [n/cm^3/sec] is uniformly distributed inside the system. The point-wise convergence tolerance for the scalar flux is set to 1.0e-5. Figure 8-2 shows the number of inner iteration required by the EP-SS_N synthetic methods as a function of the mesh size and different order of the lower-order EP-SS_N operator.

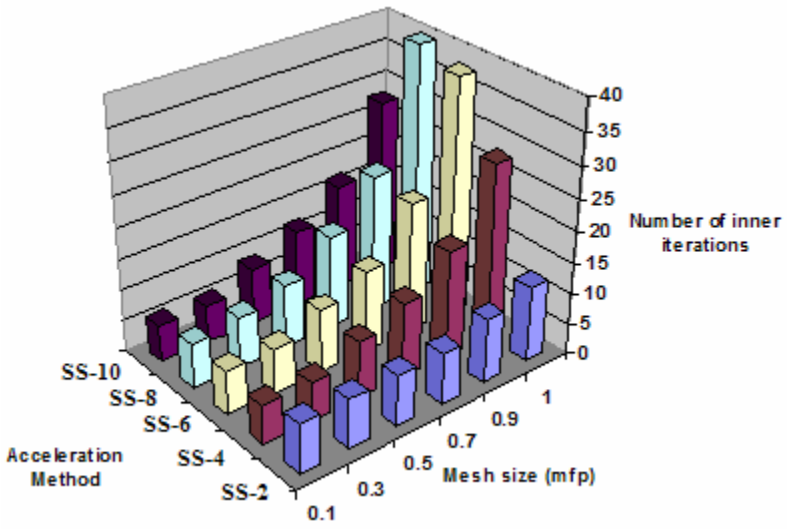

Figure 8-2. Number of inner iterations required by each acceleration method as a function of the mesh size.

Due to the inconsistent discretization of the transport and EP-SS_N operators, the synthetic acceleration method degrades in terms of performance as the size of the mesh increases, and for mesh sizes greater than 1.0 *mfp* the acceleration technique becomes unstable. Table 8-2 compares the EP-SS_N synthetic and the unaccelerated transport methods based on the number of inner iterations for a 1.0 *mfp* mesh size.

Table 8-2. Comparison of the number of inner iteration between EP-SS_N synthetic methods and unaccelerated transport.

Method	Inner iterations
EP-SS_2	12
EP-SS_4	28
EP-SS_6	38
EP-SS_8	40
Unaccelerated transport	262

The EP-SS_N synthetic methods reduce the number of inner iteration from ~6 to ~21 times with respect to the unaccelerated transport calculation. Note that as the SS_N order is increased, the acceleration performance is degraded; this behavior is due to the increasing number of inner iterations required to solve higher order EP-SS_N equations.

8.3.1 Comparison of the EP-SS_N Synthetic Acceleration with the Simplified Angular Multigrid Method

The EP-SS_N synthetic acceleration method is compared with the Simplified Angular Multigrid (SAM).[39] I have tested the effects of the scattering ratio and differencing schemes on the convergence rate. The test problem is a 10x10x10 cm^3 box with homogeneous medium. A vacuum boundary condition is prescribed on all surfaces. A fixed source of magnitude 1.0 particles/cm^3/s is placed in a region ranging from 4 to 6 cm along the *x*-, *y*-, and *z*-axes. The differencing schemes tested with the PENTRAN code are DZ, DTW and EDW. The problem is discretized with a 1.0 cm uniform mesh and an S_8 level-symmetric quadrature set is used in the calculations. The point-wise flux convergence tolerance is set to 1.0e-6.

Figures 8-3, 8-4, and 8-5 show the number of inner iterations required to converge as a function of the scattering ratio for the Source Iteration (SI), SAM and EP-SS_2 synthetic acceleration method, using the DZ, DTW, and EDW differencing schemes, respectively.

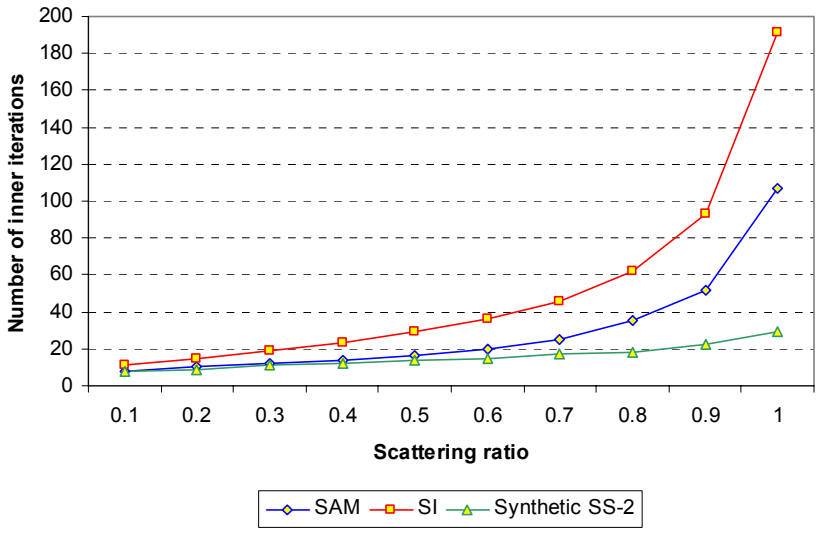

Figure 8-3. Number of inner iterations as a function of the scattering ratio (DZ differencing scheme).

The synthetic method improves the convergence rate by a factor of ~6.5 for a scattering ratio of 1.0. Note also that the performance of the synthetic method is not significantly affected by the scattering ratio.

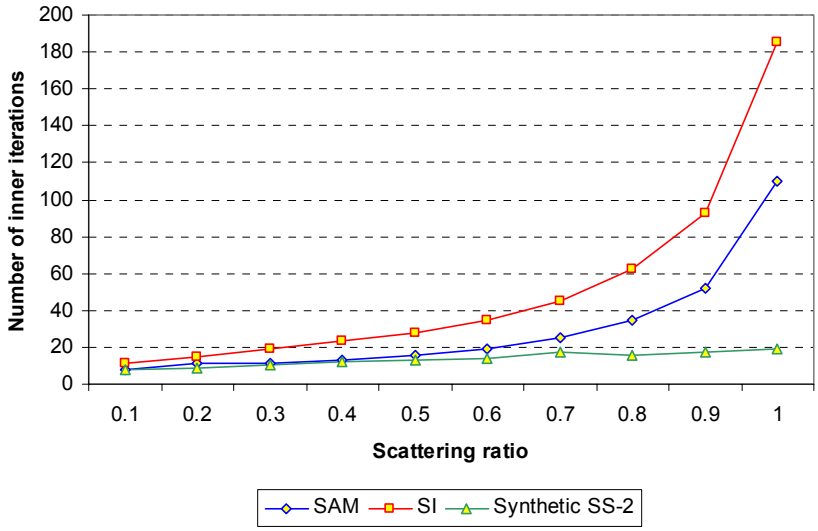

Figure 8-4. Number of inner iterations as a function of the scattering ratio (DTW differencing scheme).

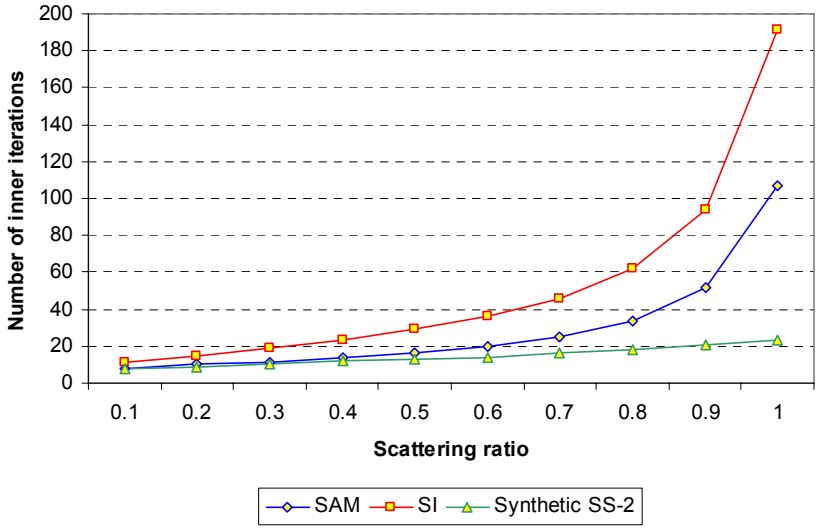

Figure 8-5. Number of inner iterations as a function of the scattering ratio (EDW differencing scheme).

As shown in Figures 8-4 and 8-5, the synthetic acceleration method improves the convergence rate compared to the SI and SAM methods. For the DTW and EDW

schemes the synthetic method reduces the number of inner iterations by a factor of ~10 and ~8, respectively. The inconsistent discretization of the operators does not yield significant instabilities in these cases; this is due to the fact that the fine-mesh size is adequate to yield a stable acceleration scheme.

Figure 8-6 shows a comparison of the number of inner iterations for the synthetic method with DZ, DTW, and EDW differencing schemes.

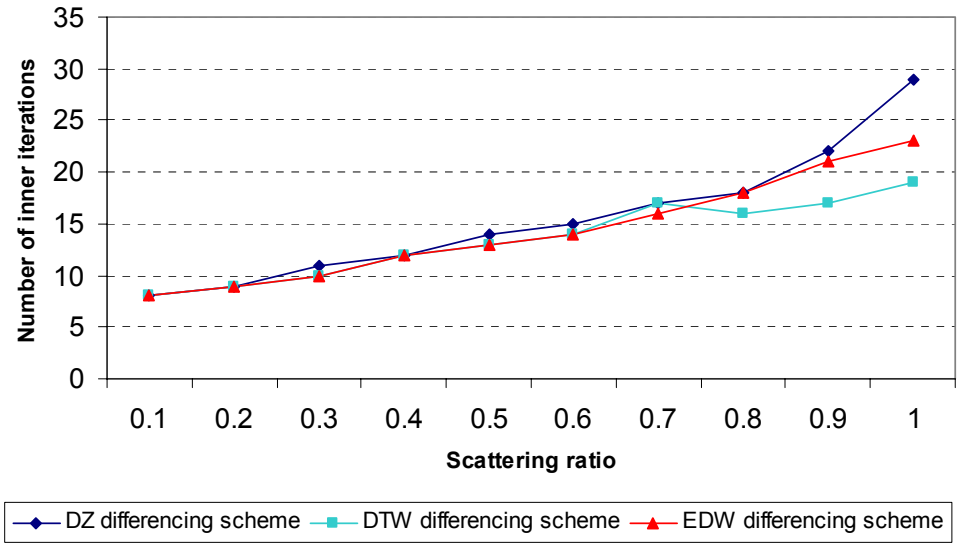

Figure 8-6. Number of inner iterations for the EP-SS_2 synthetic method obtained with DZ, DTW, and EDW differencing schemes.

All the differencing schemes perform similarly for scattering ratios up to 0.7; however, for scattering ratios greater than 0.8, the DTW differencing scheme yields the best convergence performance. The degraded performance of the DZ differencing scheme is due to the flux fix-up performed on the solution. The EDW differencing scheme degrades the performance of the synthetic method, because for scattering ratios close to unity, the physics of the problem is dominated by scattering processes, while the EDW differencing scheme predicts an exponential behavior of the particle flux.

8.4 Limitations of the EP-SS_N Synthetic Acceleration Method

Based on the analysis of the EP-SS_N synthetic acceleration method, I have identified the following limitations:

- Stability of the method is dependent on a mesh size smaller than 1.0 *mfp*.
- The method is affected by numerical oscillations for multidimensional problems with heterogeneous materials.
- Domain decomposition algorithms in parallel computing environments may worsen the performance of the synthetic method.

As previously discussed, these limitations are mainly due to the inconsistent discretization of the transport and EP-SS_N operators. However, if this condition is met, it does not necessarily imply unconditional stability of the method. Hence, for large heterogeneous multi-dimensional problem, this method is of limited applicability with current formulations.

To address this problem, I have decoupled the S_N and EP-SS_N methods by using the last one as a preconditioner. The philosophy behind this approach is to use the PENSSn code to obtain an initial solution in a fraction of the time required by the transport calculation; then the solution is introduced as an initial guess into the transport code. This approach has led to the development of the Flux Acceleration Simplified Transport (FAST©) system, which is a fully automated preconditioning system for the discrete ordinates method.

CHAPTER 9
FAST©: FLUX ACCELERATION SIMPLIFIED TRANSPORT PRECONDITIONER BASED ON THE EP-SS_N METHOD

In this chapter, I will discuss the development and implementation of FAST© in the PENTRAN Code System. FAST© is based on the PENSSn code and it is a fully automated system, which is integrated into PENTRAN-SSn. The FAST© algorithm is used to precondition and speed-up both criticality and fixed source calculations.

I will present the performance of the new system for both criticality and shielding calculations. Three problems with significant transport effects will be used to asses the capability of the algorithm. The first problem was introduced in Section 6.4.3, and it is characterized by regions of air and a considerable leakage probability. The second problem, introduced in Section 6.4.5, will demonstrate the effectiveness of the FAST© algorithm in dealing with fixed source problems characterized by anisotropic scattering. The third problem is a 3-D whole-core calculation based on the MOX 3-D Fuel Assembly Benchmark extension[47] proposed by OECD/NEA. For these test problems, I will discuss the speed-up obtained with FAST© compared to a conventional transport calculation without preconditioning.

9.1 Development and Implementation of FAST©

The FAST© code is derived from the PENSSn code described in Chapters 6 and 7; the code retains every feature of PENSSn, except for balance table generation and advanced input features which are part of the stand-alone code only. These modifications have been necessary in order to completely integrate the code into PENTRAN-SSn. The

system is fully automated and in order to invoke it, only an additional input card is required in a standard PENTRAN deck, as shown in Figure 9-1.

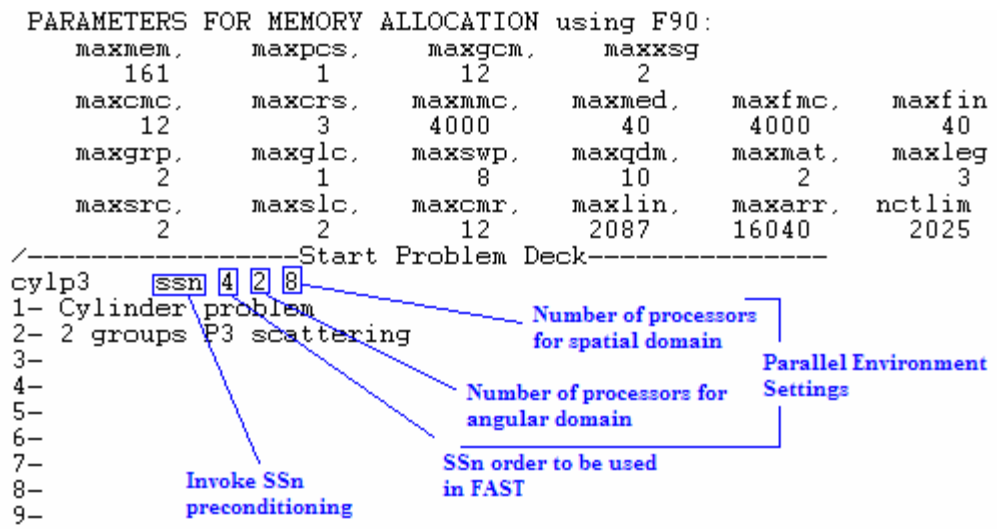

Figure 9-1. Card required in PENTRAN-SSn input deck to initiate SS_N preconditioning.

More details about the usage of the FAST© preconditioner can be found in the PENTRAN manual.[48] The SS_N card is used to instruct PENTRAN to generate the files needed by the FAST© algorithm such as problem definition, cross-sections and fixed source distribution. The FAST© algorithm prepares the initial solution for the S_N code and creates output files for scalar flux, currents, and criticality eigenvalue. The output files are dumped for each energy group and they are split into *file_a* for the first half of the spatial domain and *file_b* for the second half. This format is used to contain the file sizes under the limits handled by current operating systems. The files containing the initial solution are read successively by PENTRAN and used as initial guess for starting the calculation. Figure 9.2 shows the flowchart for PENTRAN-SSn system.

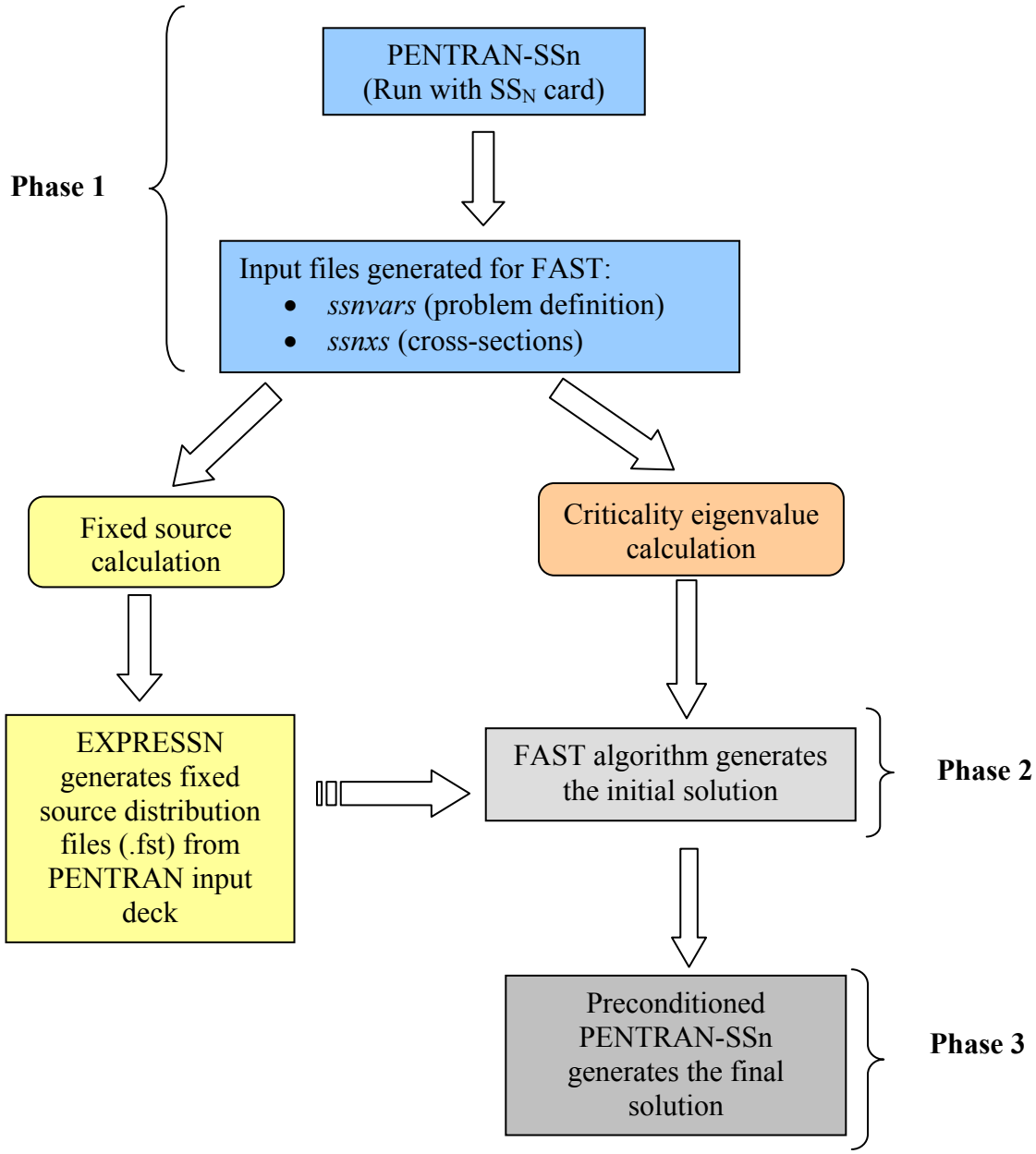

Figure 9-2. Flow-chart of the PENTRAN-SSn Code System.

As shown in Figure 9-2, the transport calculation is composed of three main phases: Phase 1 consists of processing the input deck and generating input files for the FAST© algorithm; in phase 2, the FAST© preconditioning algorithm generates the initial solution and dumps it to the output files; phase 3 completes the transport calculation by running the preconditioned PENTRAN-SSn code.

As I will show in the next sections, the FAST© preconditioner produces an accurate solution within a fraction of the computation time required by a standard transport calculation; hence, by starting the transport calculation with this initial solution, convergence is rapidly achieved. The acceleration performance can also be explained in terms of the Fourier transform of the transport operator. The Maclaurin expansion of the transport operator in the frequency domain for $\lambda \approx 0$ is given by

$$A(\lambda,\mu) = \frac{1}{1+i\lambda\mu} = 1 - i\lambda\mu + o(\lambda^2). \tag{9.1}$$

Note that $\lambda \approx 0$ modes correspond to low-frequency Fourier eigenmodes and, hence, to long wavelengths; these are error modes that span large optical distances and have weak spatial gradients. The low-frequency Fourier modes formulated in Eq. 9.1 have also weak angular dependencies. These modes of the spectrum are rapidly resolved by diffusive-like algorithms such as EP-SS_N, while the standard SI method efficiently suppresses the error modes with strong spatial and angular variations, where $\lambda >> 0$. Here, the preconditioning phase quickly resolves the low-frequency error modes, while the successive transport calculation resolves the remaining high-frequency modes.

9.2 Testing the Performance of the FAST© Preconditioning Algorithm

In this section, I will present the results obtained with the FAST© preconditioner and PENTRAN-SSn for a criticality eigenvalue, a fixed source, and the MOX 3-D Fuel Assembly benchmark problems.

9.2.1 Criticality Eigenvalue Problem

The objective of this test is to verify the performance of FAST© and PENTRAN-SSn for a criticality eigenvalue problem characterized by significant transport effects. The problem considered is described in Section 6.4.3. The system is characterized by air

regions where the EP-SS_N method is also affected by numerical difficulties. In order to remedy this issue, I have utilized the Density Factor Multiplier (DFM) described in Section 6.4.3.

I have calculated the ratio of the total number of inner iterations required to solve the problem with the preconditioned PENTRAN-SSn and with the non-preconditioned PENTRAN. Figure 9-3 presents this ratio for different EP-SS_N methods with DFM disabled and enabled.

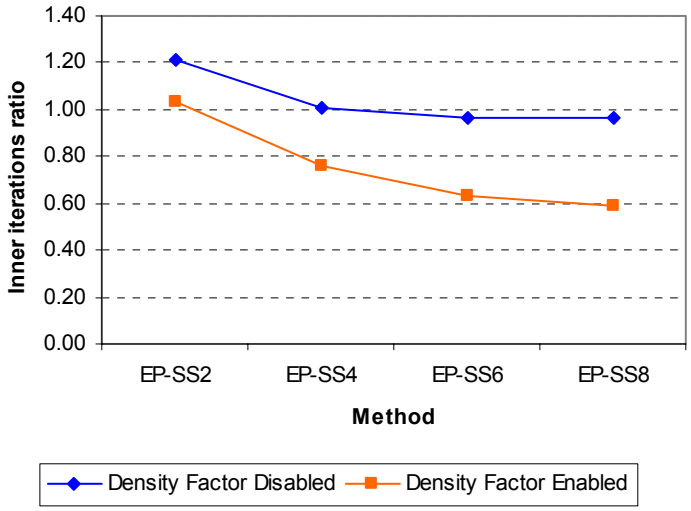

Figure 9-3. Ratio of total number of inner iterations required to solve the problem with preconditioned PENTRAN-SSn and non-preconditioned PENTRAN.

Two aspects are clearly apparent in Figure 9-3; in first instance, the acceleration performance obtained with DFM enabled is obviously superior to the case where DFM is disabled. Also, by enabling DFM, the convergence of the EP-SS_N method is significantly improved, as well as the accuracy of the solution, as shown in Table 9-1. Note that by disabling DFM, all the EP-SS_N methods did not converge within the maximum number of outer iterations set to 50.

Table 9-1. Criticality eigenvalues obtained with the preconditioned PENTRAN-SSn code for different EP-SS_N orders.

	DFM Disabled		DFM Enabled	
Method	FAST©	PENTRAN-SSn	FAST©	PENTRAN-SSn
EP-SS_2	0.69437	0.95212	0.77621	0.95213
EP-SS_4	0.83097	0.95212	0.90544	0.95212
EP-SS_6	0.8464	0.95213	0.92138	0.95212
EP-SS_8	0.84894	0.95213	0.92436	0.95215

Figure 9-3 shows also that higher order EP-SS_N methods improve the acceleration performance of PENTRAN-SSn. This behavior is expected since the air regions and the significant boundary leakage introduce substantial angular dependency; hence, as predicted high order EP-SS_N methods yield a more accurate solution and better acceleration performance.

9.2.2 Fixed Source Problem

The purpose of this test problem is to evaluate the performance of the FAST© algorithm in accelerating fixed source calculations. The problem configuration is described in section 6.4.5. I have preconditioned PENTRAN-SSn using the EP-SS_4 and EP-SS_6 methods. Figures 9-4 and 9-5 show the relative change in flux value as a function of the inner iteration number for energy groups 1 and 2.

In group 1, the non-preconditioned calculation presents a slight error reduction in the first 8 iterations, while PENTRAN-SSn logarithmically reduces the relative error from the first iteration. This behavior is due to the capability of the EP-SS_N method of resolving the low-frequency error modes and, hence, to bypass the plateau region where the SI method alone is experiencing difficulties in reducing the error.

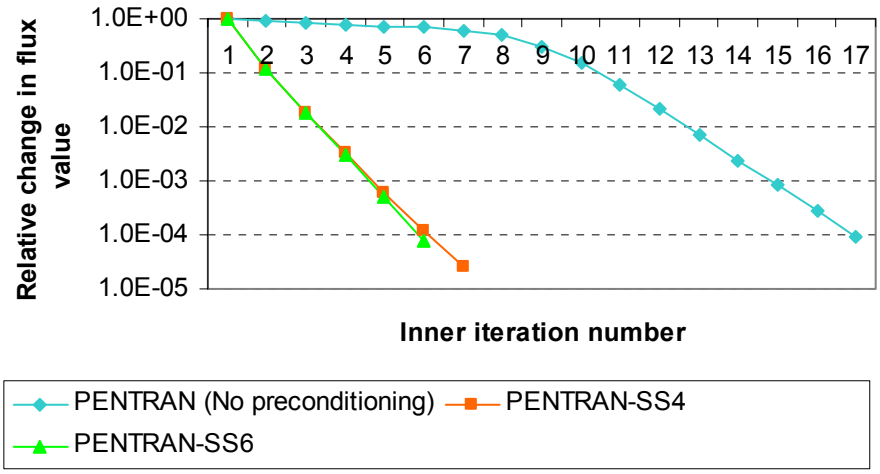

Figure 9-4. Relative change in flux value in group 1.

Figure 9-5 shows a similar behavior for energy group 2. Due to the small size of this problem the speed-up obtained with PENTRAN-SSn is roughly equal to a factor of 2.

It is also worth noticing that the EP-SS_4 method is sufficient to provide an accurate solution for an efficient preconditioning; the EP-SS_6 method does not yield significant benefits and, actually, it requires more computation time. This behavior can be explained by considering that the EP-SS_6 method does not introduce a far more accurate solution compared to EP-SS_4; however, the computational time required by FAST© for EP-SS_6 increases by ~60%, therefore explaining the behavior discussed above.

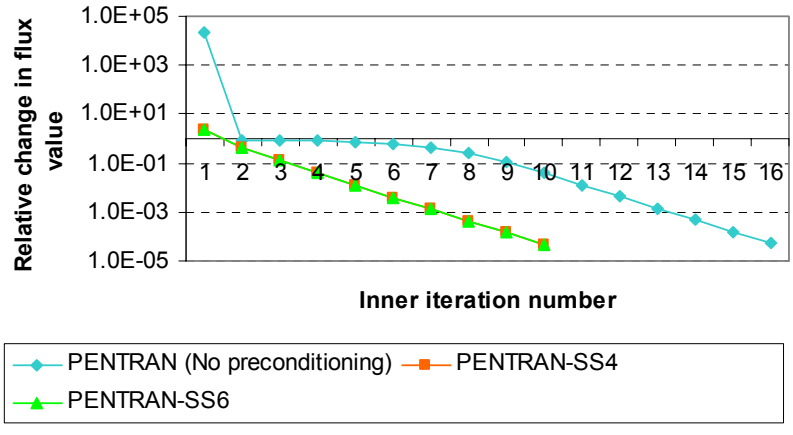

Figure 9-5. Relative change in flux value in group 2.

9.3 The MOX 3-D Fuel Assembly Benchmark Problem

The MOX 3-D Fuel Assembly Benchmark problem extension[47] has been proposed by OECD/NEA to test the capability of current transport methods and codes in dealing with whole-core simulations without spatial homogenization. The benchmark geometry is the sixteen assembly (quarter core symmetry) C5 MOX fuel assembly proposed by Cavarec.[46] Each fuel assembly consists of a 17x17 lattice of square pin cells. The side length of each pin cell is 1.26 cm and all the fuel pins and guide tubes have a 0.54 cm radius. The benchmark extension has introduced three different configurations of the reactor core:

- Unrodded configuration.
- Rodded-A configuration.
- Rodded-B configuration.

The unrodded configuration does not include any control rod in the model; in the second configuration Rodded-A, a control rod cluster is inserted 1/3 of the way into the inner UO_2 assembly (refer to Figure 6-29 for a model view on the *x*-*y* plane). In the Rodded-B configuration, the control rod clusters are inserted 2/3 of the way into the inner UO_2 assembly and 1/3 of the way into both MOX assemblies.

The seven-group, transport corrected, isotropic scattering cross-sections for each material were obtained using the collision probability code DRAGON,[49] which uses the WIMS-AECL 69-group library; these cross-sections include up-scattering processes also. These seven-group cross-sections proved the most difficult to solve and thus, were chosen to enhance transport difficulties of heterogeneous problems.

This problem presents significant transport effects due to the heterogeneous configuration, and highly angular dependent flux on the fuel-moderator interface. The

model is discretized with 946,080 fine meshes, and an S_6 level symmetric quadrature set is used. The PENTRAN Code System required a memory allocation of 1381.2 MB/Processor with an 8-space and 2-angle domain decomposition strategy. An EP-SS_4 method is used for the FAST© preconditioner, requiring 548.6 MB/Processor with a 16-space domain decomposition strategy. In the PENTRAN code, the point-wise flux convergence tolerance was chosen equal to 1.0e-3; while the convergence tolerance on the criticality eigenvalue was set equal to 1.0e-5. The FAST© preconditioner used the same convergence criteria specified for PENTRAN. Both preconditioned and non-preconditioned transport calculations ran on the 16-processors PCPENII Cluster.

In the following sections, I will present the results and performance obtained with PENTRAN-SSn using the FAST© preconditioner for the aforementioned three configurations.

9.3.1 MOX 3-D Unrodded Configuration

In this case, the model of the reactor core does not contain any control rod; therefore, the FAST© preconditioner yields the most accurate solution. Table 9-2 presents the results obtained with PENTRAN and PENTRAN-SSn as compared to a Monte Carlo reference solution.

Table 9-2. Results obtained for the MOX 3-D in the Unrodded configuration.

Method	Criticality eigenvalue	Relative error (pcm)	Inner iterations	Total Time	Time ratio	Inner it. ratio
Monte Carlo	1.14308±0.0026	-	-	-	-	-
PENTRAN	1.1466	307.94	2755	2.97d	1	1
PENTRAN-SSn	1.14477	147.85	413	15.2h	4.7	6.67
FAST©	1.15891	1384.85	-	6.9h	-	-

The inner iterations and time ratios presented in Table 9-2 are calculated using the values obtained with non-preconditioned PENTRAN and PENTRAN-SSn. Since no

synthetic acceleration is employed within PENTRAN-SSn, the values of time ratio and inner iterations ratio are similar, as expected.

For this case, I observed an improvement in the accuracy of the criticality eigenvalue due to the preconditioned solution. It is worth mentioning that improvements in accuracy have been observed also for the pin-power distributions.

Figure 9-6 shows that for this case the accurate critical flux distribution and eigenvalue provided by FAST© yield a non-oscillatory behavior of the criticality eigenvalue for PENTRAN-SSn, leading to a very steep error reduction as shown in Figure 9-7.

Figure 9-7 includes also a separate chart that shows the behavior of the criticality eigenvalue relative error for PENTRAN-SSn.

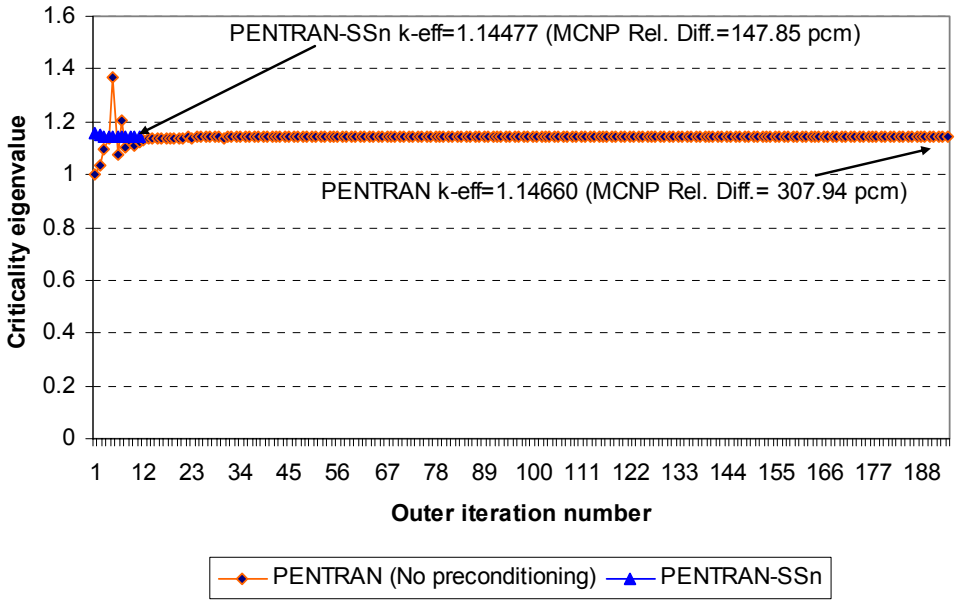

Figure 9-6. Behavior of the criticality eigenvalue as a function of the outer iterations.

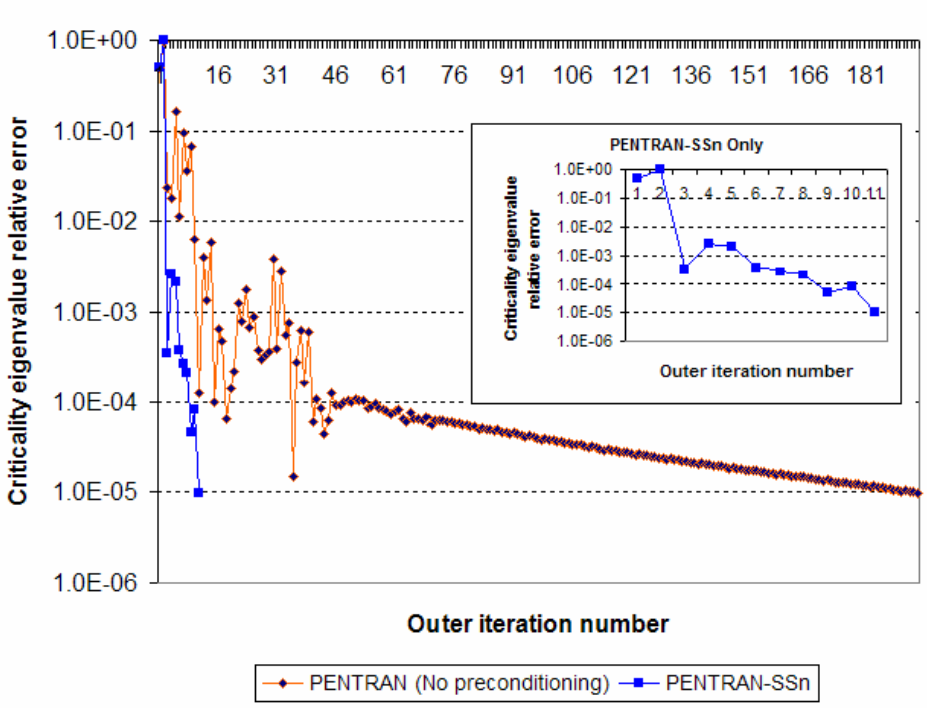

Figure 9-7. Convergence behavior of the criticality eigenvalue.

Figure 9-8 shows that the total computation time could be further reduced by terminating the preconditioning phase earlier in the calculation process, since the variation on the solution obtained with FAST© is not significant.

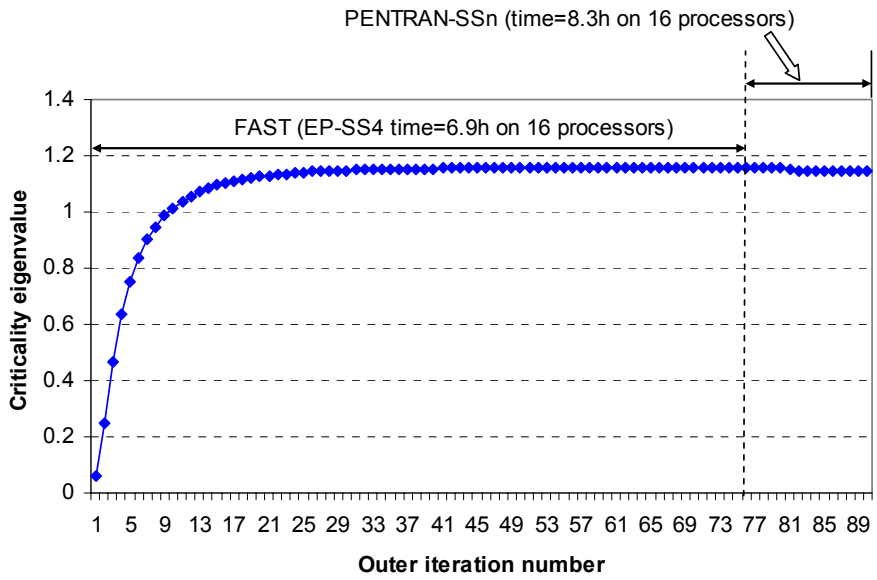

Figure 9-8. Preconditioning and transport calculation phases with relative computation time required.

9.3.2 MOX 3-D Rodded-A Configuration

The results of PENTRAN and PENTRAN-SSn, as compared to the reference Monte Carlo solution, are shown in Table 9-3.

Table 9-3. Results obtained for the MOX 3-D Rodded-A configuration.

Method	Criticality eigenvalue	Relative error (pcm)	Inner iterations	Total Time	Time ratio	Inner it. ratio
Monte Carlo	1.12806±0.0027					
PENTRAN	1.12753	-46.98	2714	3.4d	1.0	1.0
PENTRAN-SSn	1.12890	74.46	468	15.9h	5.1	5.8
FAST©	1.14582	1574.38	-	7h	-	-

For this case PENTRAN-SSn yield a speed-up of 5.1 compared to the non-preconditioned transport calculation. Note also that PENTRAN-SSn overestimates the criticality eigenvalue, which is a conservative solution from an engineering point of view.

Figure 9-9 shows the behavior of the criticality eigenvalue as a function of the outer iterations; note that PENTRAN-SSn reduces the necessary number of outer iterations by a factor of ~17 as compared to non-preconditioned PENTRAN.

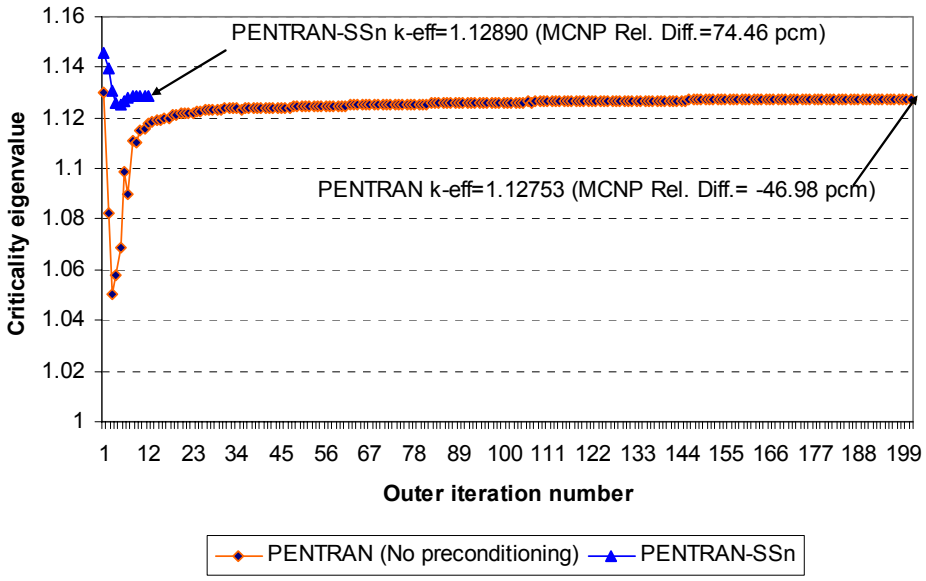

Figure 9-9. Behavior of the criticality eigenvalue as a function of the outer iterations.

Figure 9-10 shows the behavior of the relative error as a function of the outer iterations; this figure includes a chart which shows the convergence behavior for PENTRAN-SSn only.

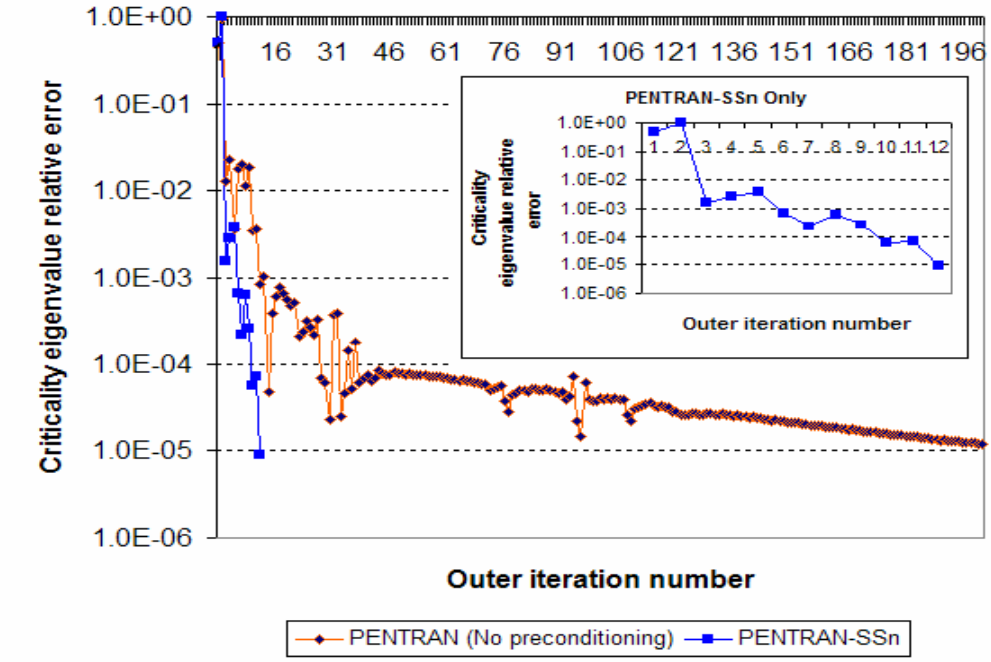

Figure 9-10. Convergence behavior of the criticality eigenvalue.

As clearly shown in Figure 9-10, the relatively accurate critical flux and criticality eigenvalue provided by FAST©, produce a steep error reduction of ~3 outer iterations for each order of magnitude. As previously discussed, the EP-SS_N method is very efficient in resolving the low-frequency error modes while the transport calculation corrects the preconditioned solution by resolving high-frequency error modes.

Figure 9-11 presents the behavior of the criticality eigenvalue calculation during the preconditioning and transport calculation phases. Note that the total computation time of 15.9 hours could be further reduced by earlier termination of the preconditioning phase, because from about the 40th iteration the relative change in the criticality eigenvalue is small.

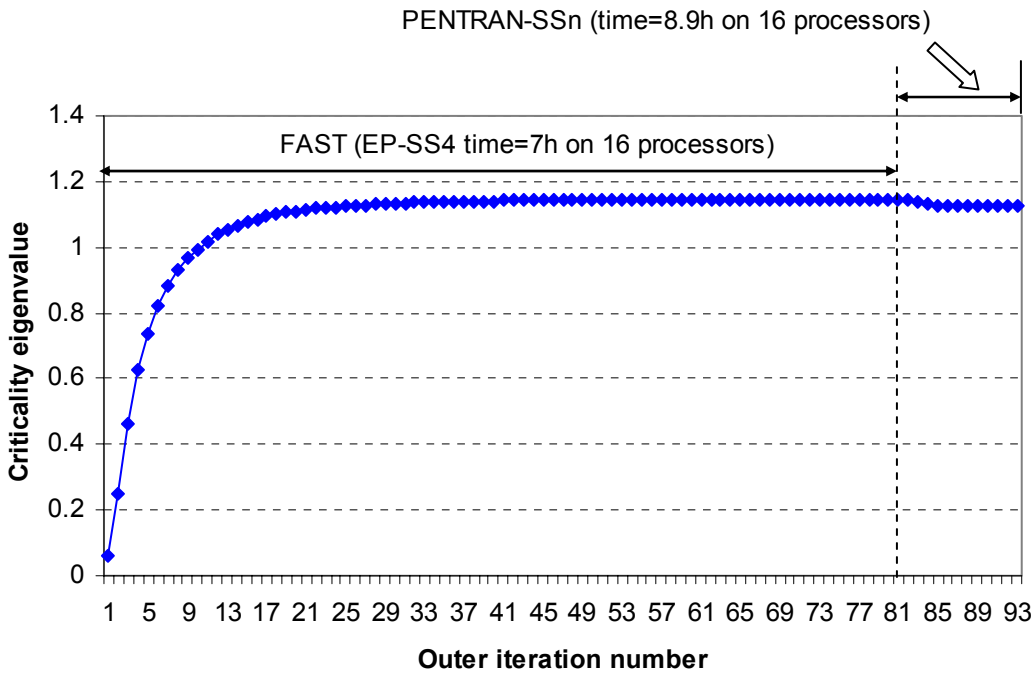

Figure 9-11. Preconditioning and transport calculation phases with relative computation time required.

9.3.3 MOX 3-D Rodded-B Configuration

For this configuration the control rods are inserted at different positions inside the reactor core. Table 9-4 presents PENTRAN and PENTRAN-SSn results as compared to the reference Monte Carlo solution.

Table 9-4. Results obtained for the MOX 3-D Rodded-B configuration.

Method	Criticality eigenvalue	Relative error (pcm)	Inner iterations	Total Time	Time ratio	Inner it. ratio
Monte Carlo	1.07777±0.0027	-	-	-	-	-
PENTRAN	1.06772	-932.48	1352	1.9d	1	1
PENTRAN-SSn	1.07356	-390.62	526	17.7h	2.6	2.57
FAST©	1.09553	1647.85	-	8.2h	-	-

The control rods introduce strong angular dependencies in the particle flux distribution; hence, as expected the FAST© preconditioner yields a less accurate solution compared to the Rodded-A or Unrodded cases. However, the accuracy is sufficient to

accelerate the transport calculation by a factor of ~2.6. Figure 9-12 shows the behavior of the criticality eigenvalue as a function of the outer iteration number.

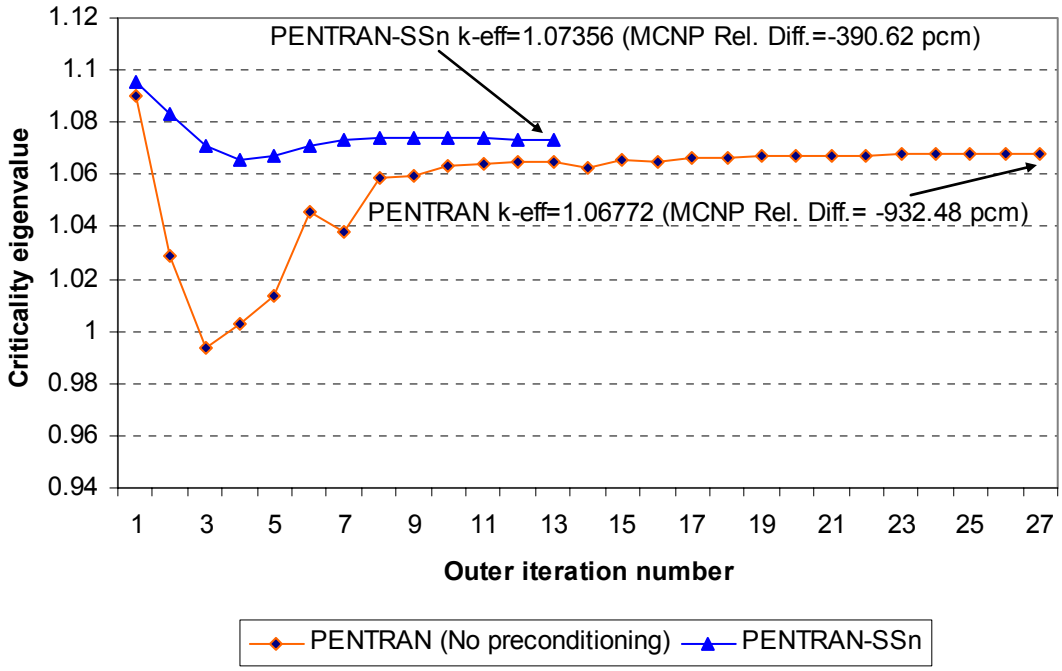

Figure 9-12. Behavior of the criticality eigenvalue as a function of the outer iterations.

PENTRAN-SSn achieve the converged solution in half of the outer iterations required by the non-preconditioned PENTRAN; moreover, for this case the FAST© algorithm increases the accuracy of the solution compared to Monte Carlo. The improved accuracy is due to the fact that the solution provided by FAST© reduces the numerical diffusion phenomenon caused by the relatively coarse discretization along the *z*-axis.

Figure 9-13 presents the convergence behavior of the criticality eigenvalue obtained with PENTRAN-SSn and PENTRAN.

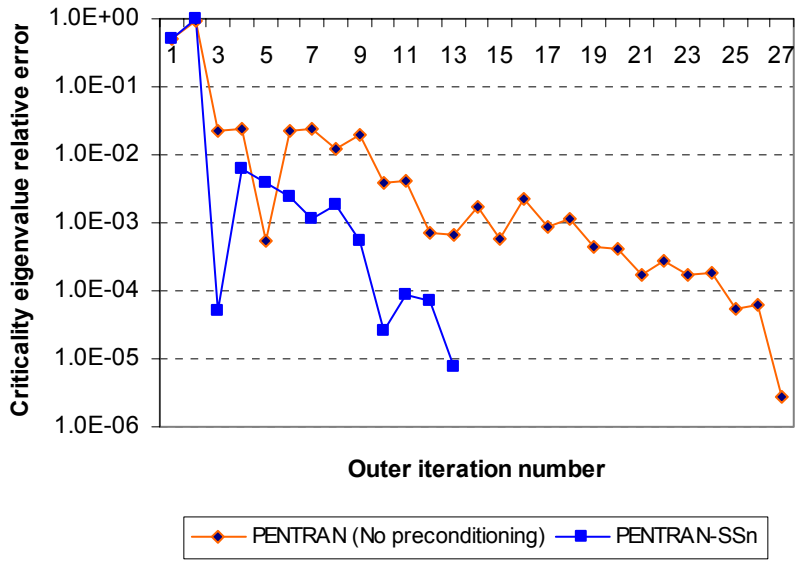

Figure 9-13. Convergence behavior of the criticality eigenvalue.

The relative error drops sharply in the first iterations due to the preconditioned solution. Figure 9-14 presents the behavior of the criticality eigenvalue in the preconditioning and transport calculation phases; again, the total computational time could be further reduced by stopping the FAST© preconditioner at an early stage in the calculation process.

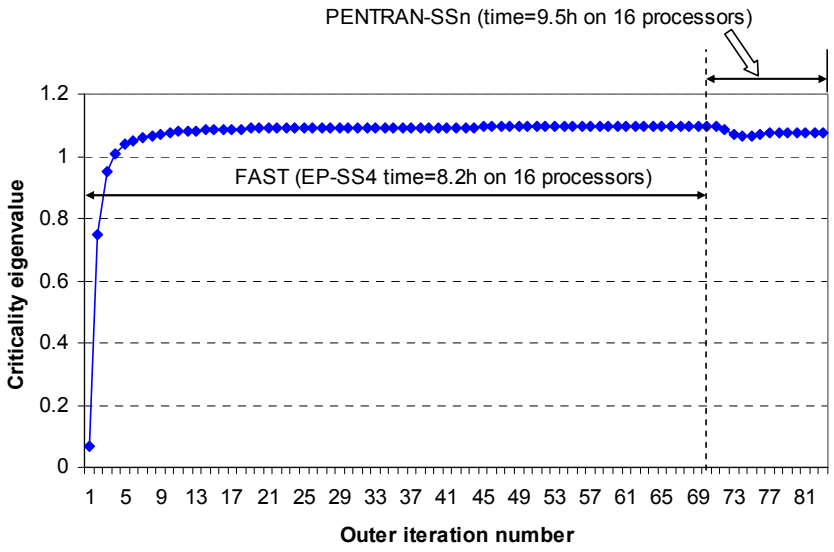

Figure 9-14. Preconditioning and transport calculation phases with relative computation time required.

In conclusion, the FAST© preconditioning algorithm integrated into the PENTRAN-SSn code has been proven very effective in accelerating the transport calculations for a large whole-core 3-D model. The computational time is reduced by a factor of 3 to 5, depending on the problem, as compared to non-preconditioned calculations. Moreover, I have also observed a slight improvement in the accuracy of the criticality eigenvalue and power distribution.

CHAPTER 10
SUMMARY, CONCLUSION, AND FUTURE WORK

In this research work, I have developed advanced quadrature sets including the P_N-EW and the P_N-T_N, along with a new biasing technique named Regional Angular Refinement (RAR). These quadrature sets do not present negative weights for any S_N order and they are suitable for problems characterized by highly angular dependent fluxes and/or sources. Based on the results obtained, the P_N-T_N quadrature set yields very accurate results and it is currently implemented in the PENTRAN Code System. The RAR technique has been proven very effective in dealing with highly angular dependent sources; the quadrature sets biased with RAR lessen the ray-effects, therefore yielding accurate results in a fraction of the time required by a standard quadrature set.

These new quadrature set generation techniques are very suitable for the simulation of medical physics applications and devices, where large regions of air require advanced quadrature sets in order to remove ray-effects from the solution.

I have investigated the Even-Parity Simplified S_N (EP-SS_N) formulation and it has been proven to be very accurate for a wide range of problems, including fixed source and criticality calculations. Therefore, I developed a new 3-D code, named PENSSn (Parallel Environment Neutral-particle Simplified Sn) which is based on the EP-SS_N equations.

The code is designed for parallel computing environments and it solves the EP-SS_N equations with anisotropic scattering of arbitrary order, for fixed source and criticality problems. The code has been benchmarked using realistic 2-D and 3-D problems, including a small LWR and an FBR model, and the MOX 2-D/3-D Fuel Assembly

Benchmark problem. The code yields very accurate results within the limitations of the method. In summary, the main limiting factors of the EP-SS_N methodology are the following:

- Optically thin media.
- Low density or void-like regions.
- Strong spatial/angular flux variations.
- Systems characterized by highly peaked anisotropic scattering.

The parallel capabilities of PENSSn have been tested on the PCPENII cluster (Nuclear and Radiological Engineering) and the Zeta cluster (Electrical and Computer Engineering High Performance Computing Lab) at the University of Florida; the code present a parallel fraction of ~87% and the parallel performance achieved follows the predictions of Amdahl's Law.

To speed-up the convergence of the S_N method, I have developed a new synthetic acceleration method based on the EP-SS_N equations; however, I found limited applicability for this method, due to instabilities which appear for mesh sizes greater than 1 *mfp* and for highly heterogeneous problems.

Further, I have developed a new preconditioning algorithm based on the EP-SS_N equations, for the acceleration of the S_N method. The Flux Acceleration Simplified Transport (FAST©) preconditioner is a fully automated system based on the kernel of the PENSSn code. FAST© is currently implemented in the PENTRAN-SSn transport code. This approach is very effective for accelerating large radiation transport problems in parallel computing environments, such as the MOX 3-D Fuel Assembly Benchmark problem. For this problem, the FAST© preconditioner has reduced the total computation time by a factor ranging between ~2.6 and ~5.1 compared to a standard non-accelerated transport calculation.

In conclusion this research work has culminated in the development of new methodologies that enhance the accuracy and feasibility of transport calculations for large realistic models. The new quadrature sets developed will improve the accuracy of dose calculations for medical physics applications.

Due to its accuracy and limited execution times, the PENSSn code is an ideal candidate for core physics, core design and certain shielding applications. A high performance has been obtained using Krylov subspace iterative solvers, which in the future may become a standard method for solving radiation transport problems. Moreover, a new formulation of the EP-SS_N equation has been developed which proved to be a key aspect for the preconditioning/acceleration algorithm designed for the S_N method.

Finally, the new FAST© preconditioner, integrated into the PENTRAN-SSn Code System, represents a leap forward in computational physics; large 3-D radiation transport calculations for core or shielding design can now be performed within a fraction of the computation time required in the past.

The methods described in this dissertation can be further enhanced and developed by studying the following issues:

- The calculation of the point-weights for the P_N-T_N quadrature set could be improved by solving the linear system of equations obtainable from the even- and odd-moment conditions of the direction cosines.
- A selection method for the biasing region in the RAR technique should be developed based on the physics of the problem.
- An automatic load-balancing algorithm for the Krylov solvers should be developed, following the ideas described in Chapter 8. This new algorithm may significantly improve the parallel performance of the angular domain decomposition strategy.
- Memory usage optimization and fine-tuning of the domain decomposition algorithms in the PENSSn code.
- Extension of the PENSSn code with time-dependent capabilities.
- The PENSSn code will be reviewed for QA.

- The new synthetic acceleration method based on the EP-SS_N equations could be investigated further; a consistent discretization between the EP-SS_N and S_N discretized operators may yield a stable algorithm for a wider range of problems.
- An optimization study on the FAST© preconditioner should be undertaken, in order to identify the necessary level of accuracy which yields the best performance in terms of speed-up.

APPENDIX A
EXPANSION OF THE SCATTERING TERM IN SPHERICAL HARMONICS

The angular dependency of the scattering cross section can be approximated by expanding the function using a complete basis of polynomial functions: spherical harmonics.

$$\sigma_{s,g'\to g}(\vec{r},\mu_0) \cong \sum_{l=0}^{L}(2l+1)\sigma_{sl,g'\to g}(\vec{r})P_l(\mu_0), \tag{A.1}$$

where

$$\sigma_{sl,g'\to g}(\vec{r}) = \int_{-1}^{1}\frac{d\mu_0}{2}\sigma_{s,g'\to g}(\vec{r},\mu_0)P(\mu_0). \tag{A.2}$$

The scattering cross section is assumed to be dependent only on the cosine of the scattering angle, i.e. $\mu_0 = \hat{\Omega}'\cdot\hat{\Omega}$, where $\hat{\Omega}$ and $\hat{\Omega}'$ are the directions of the particle before and after the scattering process. Note that this assumption implies the fact that the probability of scattering into the direction $\hat{\Omega}'$ and energy group *g'* does not depend on the initial direction of the particle. For very low energy ranges, i.e. "cold neutrons", this assumption is only approximate.

In Eqs. A.1 and A.2 the angular variable in normalized on the unit sphere as follows

$$\int d\Omega = \int_{-1}^{1}\frac{d\mu}{2}\int_{0}^{2\pi}\frac{d\varphi}{2\pi} = 1. \tag{A.3}$$

The direction vector $\hat{\Omega}$ is defined by the polar ($0 < \theta < \pi$) and azimuthal ($0 < \varphi < 2\pi$) angles. Hence the direction cosine between the directions $\hat{\Omega}$ and $\hat{\Omega}'$ is given by

$$\mu_0 = \mu\mu' + (1-\mu^2)^{1/2}(1-\mu'^2)^{1/2}\cos(\varphi-\varphi'). \tag{A.4}$$

The Legendre addition theorem states the following in terms of orthogonal spherical surface harmonics

$$P_l(\mu_0) = \frac{1}{(2l+1)}\sum_{k=-l}^{l} Y^*_{l,k}(\theta',\varphi')Y_{l,k}(\theta,\varphi). \tag{A.5}$$

The spherical harmonics functions (Y) are defined in terms of the Associated Legendre polynomials

$$Y_{l,k}(\theta,\varphi) = \sqrt{(2l+1)\frac{(l-k)!}{(l+k)!}}P_l^k(\mu)\exp(ik\varphi), \tag{A.6}$$

$$Y_{l,-k}(\theta,\varphi) = (-1)^k Y^*_{l,-k}(\theta,\varphi). \tag{A.7}$$

By using Eqs. A.5, A.6 and A.7, the Legendre polynomials are rewritten as follows

$$P_l(\mu_0) = P_l(\mu)P_l(\mu') + 2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)P_l^k(\mu')\cos\left[k(\varphi-\varphi')\right]. \tag{A.8}$$

So that the complete scattering kernel expanded in terms of spherical harmonics becomes

$$\begin{aligned}&\sum_{g'=1}^{G}\sum_{l=0}^{L}(2l+1)\sigma_{sl,g'\to g}(x,y,z)\{P_l(\mu)\phi_{l,g'}(x,y,z) + \\ &2\sum_{k=1}^{l}\frac{(l-k)!}{(l+k)!}P_l^k(\mu)[\phi^k_{Cl,g'}(x,y,z)\cos(k\varphi) + \phi^k_{Sl,g'}(x,y,z)\sin(k\varphi)]\}.\end{aligned} \tag{A.9}$$

APPENDIX B
PERFORMANCE OF THE NEW EP-SS_N FORMULATION

The new formulation of the EP-SS_N equations discussed in Chapter 4 is tested for the small FBR problem described in Chapter 6. The new formulation derived is useful to accelerate the solution of the EP-SS_N equations via the source iteration method. The small FBR model is simulated with the new formulation EP-SS_N method and its performance is compared to the standard EP-SS_N formulation.

Table B-1 shows the number of iterations required to converge and the relative computation time for different EP-SS_N methods derived with the standard formulation.

Table B-1. Performance data for the standard EP-SS_N formulation.

Method	Krylov iterations	Inner iterations	Outer iterations	Computation time (sec)
EP-SS_2	375882	1237	91	48.3
EP-SS_4	635326	1038	74	79.6
EP-SS_6	884580	953	67	110.1
EP-SS_8	1137518	907	63	144.7

Table B-2 shows the number of iterations required by the new EP-SS_N formulation for solving the small FBR benchmark problem.

Table B-2. Performance data for the new EP-SS_N formulation.

Method	Krylov iterations	Inner iterations	Outer iterations	Computation time (sec)
EP-SS_2	194071	343	17	23.2
EP-SS_4	385411	517	31	47.8
EP-SS_6	614960	589	37	75.9
EP-SS_8	865288	636	41	107.6

The new EP-SS_4 formulation reduces by more than 50% the number of inner iterations and computation time, compared to the standard formulation. This behavior is due to the reduction in terms of spectral radius achieved by the new EP-SS_N formulation.

However, the new formulation increases the spectral condition number of the matrix operators; this behavior is detected in the ratio between Krylov iterations and inner iterations, shown in Table B-3. The new EP-SS_N formulation presents higher values for this ratio, meaning that the matrix operators are characterized by larger spectral condition numbers.

Table B-3. Ratio between Krylov iterations and inner iterations.

Method	Standard EP-SS_N	Modified EP-SS_N
EP-SS_2	304	566
EP-SS_4	612	745
EP-SS_6	928	1044
EP-SS_8	1254	1361

I have observed also a degradation of the performance of the new EP-SS_N formulation for higher SS_N orders. Table B-4 compares the inner iterations ratio and time ratio between the standard and new EP-SS_N formulations; note that the speed-up decreases for increasing SS_N orders.

Table B-4. Inner iterations and time ratios for different SS_N orders.

Method	Inner iterations ratio	Time ratio
EP-SS_2	3.6	2.1
EP-SS_4	2.0	1.7
EP-SS_6	1.6	1.5
EP-SS_8	1.4	1.3

The speed-up degradation can be explained by observing that the direction dependent removal cross section in Eq. B.1 depends on the weights of the quadrature set. For high order quadrature sets, the value of the weight is decreased accordingly. Due to this aspect, the removal cross section is less affected by the scattering term as the quadrature set order increases, therefore leading to a degradation of the method. This

argument explains also the behavior observed for the transport equation, verified using the S_N formulation, where no significant benefits are observed.

$$\sigma_{m,g}^{R}(\vec{r}) = \sigma_{t,g}(\vec{r}) - \sum_{\substack{l=0,2..\\ even}}^{L-1} (2l+1) P_l(\mu_m)^2 w_m \sigma_{sl,g\to g}(\vec{r}) . \tag{B.1}$$

Moreover, note that in Table B-4 the reduction in terms of inner iterations does not match necessarily the reduction in computation time; clearly, this is due to the larger number of Krylov iterations required by the new EP-SS_N formulation compared to the standard formulation.

LIST OF REFERENCES

1. Bell G.I. and Glasstone S., *Nuclear Reactor Theory*, Robert E. Krieger Publishing CO. Inc., Malabar, FL, USA, 1985.

2. Carlson B.G., *Transport Theory: Discrete Ordinates Quadrature over the Unit Sphere*, Los Alamos Scientific Laboratory Report, LA-4554, 1970.

3. Lewis E.E. and Miller W.F. Jr., *Neutron Transport*, American Nuclear Society, La Grange Park, IL, 1993.

4. Carlson B.G. and Lathrop K.D., *Discrete Ordinates Angular Quadrature of the Neutron Transport Equation*, Los Alamos Scientific Laboratory Report, LA-3186, 1965.

5. Lathrop K.D., "Remedies for Ray Effects," Nuclear Science and Engineering, **Vol. 45**, pp. 255-268, 1971.

6. Fletcher J. K., "The Solution of the Multigroup Neutron Transport Equation Using Spherical Harmonics," Nuclear Science and Engineering, **Vol. 84**, pp. 33-46, 1983.

7. Carlson B.G., *Tables of Equal Weight Quadrature EQ_N Over the Unit Sphere*, Los Alamos Scientific Laboratory Report, LA-4734, 1971.

8. Carew J.F. and Zamonsky G., "Uniform Positive-Weight Quadratures for Discrete Ordinate Transport Calculations," Nuclear Science and Engineering, **Vol. 131**, pp.199-207, 1999.

9. Brown J.F. and Haghighat A., "A PENTRAN Model for a Medical Computed Tomography (CT) Device," *Proceedings of Radiation Protection for our National Priorities (RPSD 2000),* Spokane, Washington, September 17-21, 2000, on CD-ROM, American Nuclear Society, Inc., Lagrange Park, IL, 2000.

10. Sjoden G. E. and Haghighat A., "PENTRAN – Parallel Environment Neutral-particle TRANsport in 3-D Cartesian Geometry," *Proceedings of the Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications*, **Vol. 1**, pp. 232-234, Saratoga Springs, NY, October 6-10, 1997.

11. Longoni G. et al., "Investigation of New Quadrature Sets for Discrete Ordinates Method with Application to Non-conventional Problems," *Trans. Am. Nucl. Soc.*, **Vol. 84**, pp. 224-226, 2001.

12. Longoni G. and Haghighat A., "Development of New Quadrature Sets with the Ordinate Splitting Technique," *Proceedings of the ANS International Meeting on Mathematical Methods for Nuclear Applications (M&C 2001)*, Salt Lake City, UT, September 9-13, 2001, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2001.

13. Longoni G. and Haghighat A., "Simulation of a CT-Scan Device with PENTRAN Using the New Regional Angular Refinement Technique," *Proceedings of the 12th Biennial RPSD Topical Meeting of the Radiation Protection and Shielding Division of the American Nuclear Society*, Santa Fe, NM, April 14-18, 2002, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2002.

14. Longoni G. and Haghighat A., "Development of the Regional Angular Refinement and Its Application to the CT-Scan Device," *Trans. Am. Nucl. Soc.*, **Vol. 86**, pp. 246-248, 2002.

15. Longoni G. and Haghighat A., "Development and Application of the Regional Angular Refinement Technique and its Application to Non-conventional Problems," *Proceedings of PHYSOR 2002 ANS Topical Meeting - International Conference on the New Frontiers of Nuclear Technology: Reactor Physics, Safety and High-Performance Computing,* Seoul, Korea, October 7-10, 2002, on CD-ROM, American Nuclear Society, Inc., Lagrange Park, IL, 2002.

16. Kucukboyaci V. et al., "PENTRAN Modeling for Design and Optimization of the Spherical-Shell Transmission Experiments," *Trans. Am. Nucl. Soc.*, **Vol. 84**, pp. 156-159, 2001.

17. Adams M. L. and Larsen E. W., "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," Progress in Nuclear Energy, **Vol. 40**, n. 1, 2002.

18. Gelbard E., Davis J., and Pearson J., "Iterative Solutions to the P_l and Double-P_l Equations," Nuclear Science and Engineering, **Vol. 5**, pp. 36-44, 1959.

19. Ferziger J. H. and Milovan P., *Computational Methods for Fluid Dynamics Second Edition*, Springer-Verlag, Berlin Heidelberg, Germany, 1999.

20. Golub G. and Ortega J.M., *Scientific Computing An Introduction with Parallel Computing*, Academic Press, San Diego, CA, 1993.

21. Lewis E. E. and Palmiotti G., "Simplified Spherical Harmonics in the Variational Nodal Method," Nuclear Science and Engineering, **Vol. 126**, pp. 48-58, 1997.

22. Brantley P.S. and Larsen E.W., "The Simplified P_3 Approximation," Nuclear Science and Engineering, **Vol. 134**, pp. 1-21, 2000.

23. Longoni G. and Haghighat A., "The Even-Parity Simplified S_N Equations Applied to a MOX Fuel Assembly Benchmark Problem on Distributed Memory Environments," *PHYSOR 2004 – The Physics of Fuel Cycles and Advanced Nuclear Systems: Global Developments*, Chicago, IL, April 25-29, 2004, on CD-ROM, American Nuclear Society, Inc., Lagrange Park, IL, 2004.

24. Gamino R.G., "Three-Dimensional Nodal Transport Using the Simplified P_L Method," *Proceedings of the International Topical Meeting Advances in Mathematics, Computations, and Reactor Physics,* Pittsburgh, PA, April 28-May 2, 1991, on CD-ROM, American Nuclear Society, Inc., Lagrange Park, IL, 1991.

25. Longoni G., Haghighat A., and Sjoden G., "Development and Application of the Multigroup Simplified P_3 (SP_3) Equations in a Distributed Memory Environment," *Proceedings of PHYSOR 2002 ANS Topical Meeting - International Conference on the New Frontiers of Nuclear Technology: Reactor Physics, Safety and High-Performance Computing*, Seoul, Korea, October 7-10, 2002, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2002.

26. Longoni G. and Haghighat A., "Development and Applications of the SP_L Methodology for a Criticality Eigenvalue Benchmark Problem," *Proceedings of the ANS Topical Meeting on Nuclear Mathematical and Computational Sciences: A Century In Review – A Century Anew (M&C 2003)*, Gatlinburg, TN, April 6-11, 2003, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2003.

27. Gropp W., Lusk E., and Skjellum A., *Using MPI Portable Parallel Programming with the Message Passing Interface*, The MIT Press, Cambridge, Massachussetts, 1999.

28. Gelbard E. M. and Hageman L. A., "The Synthetic Method as Applied to the S_N Equations," Nuclear Science and Engineering, **Vol. 37**, pp. 288-298, 1969.

29. Alcouffe R. E., "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations," Nuclear Science and Engineering, **Vol. 64**, pp. 344-355, 1977.

30. Chang J. and Adams M., "Analysis of Transport Synthetic Acceleration for Highly Heterogeneous Problems," *Proceedings of the ANS Topical Meeting on Nuclear Mathematical and Computational Sciences: A Century In Review – A Century Anew (M&C 2003)*, Gatlinburg, TN, April 6-11, 2003, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2003.

31. Warsa J. S., Wareing T. A. and Morel J. E., "On the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional S_N Calculations in the Presence of Material Discontinuities," *Proceedings of the ANS Topical Meeting on Nuclear Mathematical and Computational Sciences: A Century In Review – A Century Anew (M&C 2003)*, Gatlinburg, TN, April 6-11, 2003, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2003.

32. Warsa J. S., Wareing T. A. and Morel J. E., "Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional S_N Calculations in Problems with Material Discontinuities," Nuclear Science and Engineering, **Vol. 147**, pp. 218-248, 2004.

33. Longoni G. and Haghighat A., "A New Synthetic Acceleration Technique based on the Simplified Even-Parity S_N Equations," accepted for publication on Transport Theory and Statistical Physics, 2004.

34. Sjoden G. and Haghighat A., "The Exponential Directional Weighted (EDW) S_N Differencing Scheme in 3-D Cartesian Geometry," *Proceedings of the Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications*, **Vol. II**, pp. 1267-1276, Saratoga Springs, NY, October 6-10, 1997.

35. Lathrop K., "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," Journal of Computational Physics, **Vol. 4**, pp. 475-498, 1969.

36. Petrovic B. and Haghighat A., "Analysis of Inherent Oscillations in Multidimensional S_N Solutions of the Neutron Transport Equation," Nuclear Science and Engineering, **Vol. 124**, pp. 31-62, 1996.

37. Sjoden G. E., "PENTRAN: A Parallel 3-D S_N Transport Code With Complete Phase Space Decomposition, Adaptive Differencing, and Iterative Solution Methods," Ph.D. Thesis in Nuclear Engineering, Penn State University, 1997.

38. Nakamura S., *Computational Methods in Engineering and Science*, Wiley, New York, 1977.

39. Kucukboyaci V.N. and Haghighat A., "Angular Multigrid Acceleration for Parallel Sn Method with Application to Shielding Problems," *Proceedings of PHYSOR 2000 - ANS International Topical Meeting on Advances in Reactor Physics and Mathematics and Computation into the Next Millennium*, Pittsburgh, PA, May 7-12, 2000, on CD-ROM, American Nuclear Society, Inc., La Grange Park, IL, 2000.

40. Reed W. H., "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Nuclear Science and Engineering, **Vol. 45**, pp. 245-254, 1971.

41. Larsen E. W., "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations. Part I: Theory," Nuclear Science and Engineering, **Vol. 82**, pp. 47-63 1982.

42. Kobayashi K., Sugimura N. and Nagaya Y., *3-D Radiation Transport Benchmark Problems and Results for Simple Geometries with Void Regions*, OECD/NEA report, ISBN 92-64-18274-8, Issy-les-Molineaux, France, November 2000.

43. Takeda T. and Ikeda H., *3-D Neutron Transport Benchmarks NEACRP-L-330*, OECD/NEA, Osaka University, Japan, March 1991.

44. *Benchmark on Deterministic Transport Calculations Without Spatial Homogenization*, OECD/NEA report, ISBN 92-64-02139-6, Issy-les-Molineaux, France, 2003.

45. Haghighat A., *Manual of PENMSH Version 5 – A Cartesian-based 3-D Mesh Generator*, University of Florida, Florida, June, 2004.

46. Cavarec C., *The OECD/NEA Benchmark Calculations of Power Distributions within Assemblies*, Electricité de France, France, September 1994.

47. *Benchmark Specification for Deterministic MOX Fuel Assembly Transport Calculations Without Spatial Homogenisation (3-D Extension C5G7 MOX)*, OECD/NEA report, Issy-les-Molineaux, France, April, 2003.

48. Sjoden G. and Haghighat A., *PENTRAN: Parallel Environment Neutral-particle TRANsport S_N in 3-D Cartesian Geometry – Users Guide to Version 9.30c*, University of Florida, Florida, May 2004.

49. Marleau G., A. Hébert, and R. Roy, *A User's Guide for DRAGON*, Ecole Polytechnique de Montréal, Canada, December 1997.

BIOGRAPHICAL SKETCH

Gianluca Longoni was born in Torino, Italy, on 31st of October, 1975; he is the son of Giancarlo and Annalisa, and has a brother, Daniele, who is an excellent student and prospective aerospace engineer. In 1994, Gianluca enrolled in the nuclear engineering program at Politecnico di Torino, located in Torino. He obtained the degree Laurea in Ingegneria Nucleare in March 2000; he performed his research work under the supervision of Piero Ravetto, and he developed a new 2-D radiation transport code based on the characteristics method in hexagonal geometry, for Accelerator Driven Systems (ADS).

He moved to the USA in August 2000 to pursue his Ph.D. with Professor Alireza Haghighat at Penn State University, Pennsylvania. In fall 2001 he moved to University of Florida in Gainesville, as Prof. Haghighat joined the Nuclear and Radiological Engineering Department; Gianluca continued his research work in Florida. Gianluca has presented his research work in a number of international conferences in the US, South Korea and Europe.

Gianluca is fond of basketball, having been a player in the pro-league in Italy. He is now a senior student in different martial art styles, including Iwama-Ryu aikido, Mudokwan taekwondo, and Iaido Japanese swordsmanship.