Orthogonal-Mesh, 3-D Sn with Embedded 2-D Method of Characteristics for Whole-Core, Pin-Resolved Reactor Analysis

by

Mitchell T.H. Young

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences and Scientific Computing)
in the University of Michigan
2016

Doctoral Committee:

Professor William R. Martin, Co-Chair
Adjunct Assistant Professor Benjamin Collins, ORNL, Co-Chair
Professor Thomas Downar
Professor Robert Krasny
Professor Edward Larsen


A + B = x
C - D = R + π

For all the people.

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CASL Consortium for the Advanced Simulation of Light Water Reactors

CDD Corrected Diamond Difference

CM Coarse Mesh

CMFD Coarse Mesh Finite Difference

DD Diamond Difference

FD Finite Difference

FM Fine Mesh

FSR Flat Source Region

IFBA Integral-Fuel Burnable Absorber

KBA Koch-Baker-Alcouffe

LD Linear Discontinuous

LWR Light Water Reactor

MoC Method of Characteristics

NEM Nodal Expansion Method

ORNL Oak Ridge National Laboratory

pcm per cent mille

SANM Semi-Analytic Nodal Method

SC Step Characteristics

SPn Simplified Pn

TL Transverse Leakage

WDD Weighted Diamond Difference

CHAPTER 1

Introduction

1.1 Motivation

Neutronic analysis of nuclear reactors is an important aspect of reactor analysis and design, as it governs many of the most important phenomena from an engineering perspective. Energy release rates, changes in nuclide inventory and transient behaviors are all driven by the distribution of neutrons in space, energy, angle and time, and how those neutrons interact with their background media. This behavior is described by the Boltzmann transport equation, which is described in detail on the following pages.

The reactor physics community has been using computers to assist in reactor core design and analysis since the beginning of the industry. The methods used have historically been limited by the computing resources that were available at the time, and have evolved considerably as computer speed and memory have increased. These computational techniques are numerous and varied. In the taxonomy of reactor analysis methods, the first bifurcation occurs between deterministic and stochastic, or Monte Carlo, methods. Monte Carlo methods randomly simulate the individual behavior of many simulated neutron "histories," yielding estimates of the average behavior. While Monte Carlo methods do not solve the transport equation *per se*, by simulating the underlying physics they obtain estimates of such solutions within some statistical uncertainty. On the other hand, deterministic methods endeavor to solve the transport equation directly. This work is concerned mostly

with deterministic methods, of which there are many. Each deterministic method employs a series of approximations and/or discretization schemes to the transport equation to yield a form that is suitable for solving on a computer. These methods have evolved greatly as available computing power has increased, with newer methods including more physics and increased fidelity.

1.2 Pin-Resolved Transport

Reactors take many forms with different fuel compositions and configurations, different coolants, and different operating characteristics. By far the most common type of reactor, especially in the commercial sector, are Light Water Reactors (LWRs). These tend to be large systems composed of many long, slender, cylindrical fuel rods, which are assembled into rectangular lattices and loaded into a roughly-cylindrical core. This is depicted roughly in Fig. 1.1. These rods are immersed in water, which serves as both a coolant and a mechanism for slowing neutrons down, improving their likelihood of initiating fission without becoming captured or leaking out of the system. Neutrons born from fission within a fuel rod tend to leak out of the rod and into the surrounding water, where they thermalize, ideally to reenter a fuel region and initiate a new fission. As a result, the behavior of neutrons varies strongly throughout a single pin cell, as well as through the larger reactor domain. Neutrons are also born from fission at high energies, on the order of 10^6 eV, and slow down to thermal energies on the order of a fraction of an electronvolt. To obtain an accurate understanding of neutron behavior in such systems requires a sufficiently fine treatment of the spatial and energy variables.

Older reactor analysis methods relied upon multi-stage simulations, where small subdomains were analyzed using fine meshes and energy grids, then homogenized and composed to solve the whole-core problem using lower-order methods. The errors accumulated from the series of homogenizations and simplifications not only increased the degree of un-

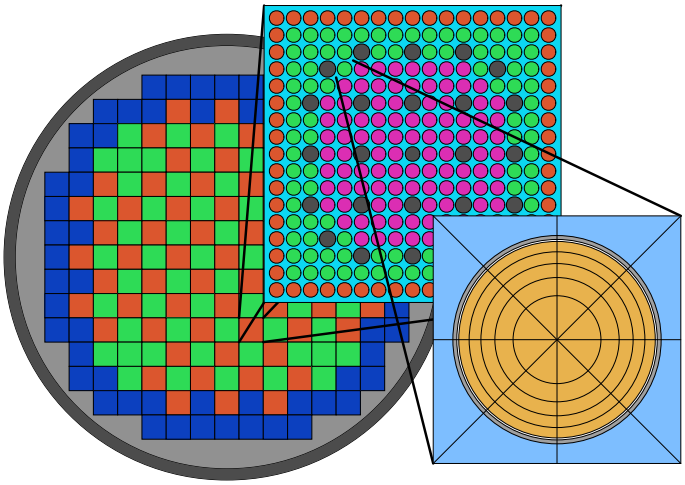


Figure 1.1: Typical power reactor core layout and mesh.

certainty in the final solutions, but also limited the range of situations to which these methods could be reasonably applied. Improvements to reactor analysis methods have enabled power uprates in existing reactors by reducing uncertainty, and have allowed core designers to use more advanced fuel technologies, such as Integral-Fuel Burnable Absorber (IFBA) pins and other burnable absorbers, which were previously too difficult to analyze. More accurate neutronics simulations also result in a better understanding of the state of the core under operation, improving operational safety and economy.

For the above reasons, it has become more popular to attempt direct solutions to the transport equation for entire reactor cores with pin resolved meshes. The Consortium for the Advanced Simulation of Light Water Reactors (CASL), a Department of Energy initiative has supported this work. Under reasonable discretization, this can lead to numbers of unknowns on the order of trillions (see Table 1.1). Methods are therefore sought that can solve the transport as quickly and efficiently as possible, and are capable of scaling to computation on very large, massively parallel computers.

When developing such methods, it is important to consider the nature of the computers upon which computation will take place. Throughout the history of supercomputing, many of the increases in performance (see Fig. 1.2) have come from adding parallelism.

Table 1.1: Typical number of unknowns for a whole-core, pin-resolved LWR calculation.

Energy Groups	47
Discrete Angles	144
Pin Cells	$\sim 56,000$
Axial Mesh	~ 400
Regions per Pin	64
Total Spatial Regions	$\sim 1,400,000,000$
Total Unknowns	$\sim 9.7 \times 10^{12}$

Recent years have also seen the addition of co-processing hardware and the introduction of more exotic processor architectures such as general-purpose GPUs, the Intel Xeon Phi, and the Sunway processors found in Sunway TaihuLight, the fastest computer in the world at the time of this writing. These co-processors have gained in popularity, since they offer much better floating point performance per cost per watt than traditional CPUs. While these highly-parallel, heterogeneous architectures have enabled supercomputers with astonishing floating-point performance, it tends to be much more difficult to develop efficient algorithms for these architectures. Many of the reactor analysis methods and codes in use today still do not perform well on these machines, if at all.

1.3 Method of Characteristics and 2-D/1-D

One class of methods in use today is the Method of Characteristics (MoC) and the 2-D/1-D iteration scheme. These have seen widespread use in reactor analysis codes such as CRX [25]. DeCART [16] and MPACT [8, 22, 32]. This approach is a type of "planar synthesis" method, which employs a series of two-dimensional transport sweepers[1] to treat each radial slab of a reactor. These sweepers are then coupled through some form of one-dimensional solver, which treats the axial dimension. Multiple iterations between the radial and axial sweepers are then used to converge the solution. The result is an approximate three-dimensional solution for the 3-D whole-core problem. The 2-D/1-D methods use 2-D MoC

[1] The term "sweeper" is used to refer to a technique for solving the transport equation by which angular flux is propagated from a boundary through the spatial domain.

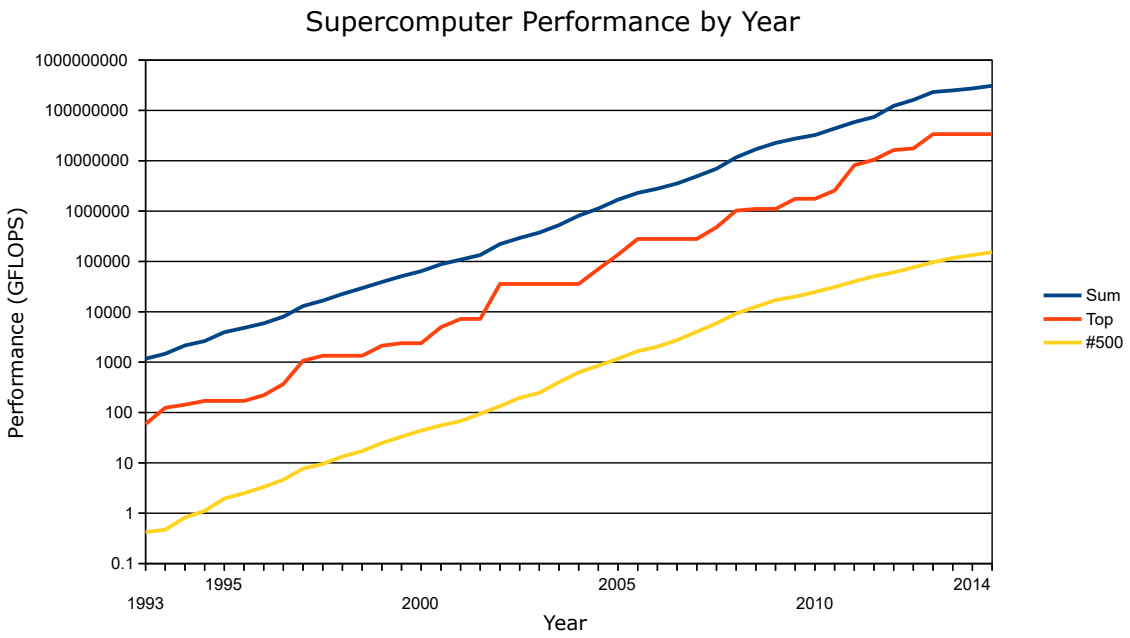


Figure 1.2: Supercomputer performance by year. Data collected from TOP500, showing performance of fastest and 500th fastest computers, along with total performance of all 500 fastest computers in the world.

sweepers to treat the radial directions, while myriad axial solvers have been used to couple them, including Fine Mesh (FM) diffusion, nodal diffusion, and more recently S_N methods. The MoC (described in more detail in Section 2.4.2) is particularly well suited for 2-D reactor applications because it is based upon ray tracing and can be applied to very general and efficient meshes. While MoC has been extended to 3-D, it has remained intractably expensive for whole-core applications [21].

While these 2-D/1-D methods have proved to be a great improvement over older methods, they still suffer in accuracy in certain cases that stress the assumption of separability between the axial and radial directions. This is typically the case in regions where strong material discontinuities occur in both the radial and axial directions. Such is the case surrounding the tips of control rods and other strong neutron absorbers. Higher-order representations of the transverse leakage have been recently developed [30] to improve accuracy considerably, but remain computationally expensive. Either way, MoC has been shown to

scale well on traditional massively parallel computers [23]. Work in the last several years has also shown that MoC has promise on massively parallel devices like GPUs [5, 6].

1.4 Orthogonal Mesh S_N

Another class of solution techniques involve formulating an approximate solution to the transport equation within each mesh region. This approach, typically referred to as S_N methods, may be applied to many different types of meshes, though S_N formulations become more complicated as the mesh becomes more general. When restricted to an orthogonal grid, S_N methods become incredibly efficient on a per-mesh element basis.

The orthogonal-mesh S_N method was used in the Denovo code [10, 11] at Oak Ridge National Laboratory, and permits the use of extremely efficient parallel wavefront propagation algorithms [20]. Denovo has been applied to reactor analysis problems and demonstrated impressive computational performance, scaling well to hundreds of thousands of processor cores [2, 15].

While the Cartesian grid approach exhibits impressive parallel performance, it tends to produce significant modeling error when used to approximate the circular fuel structures found in nuclear reactors. It was found that reduction of this error required a prohibitively fine mesh to resolve intra-pin behavior, and that reasonably accurate solutions using the S_N method on a Cartesian mesh alone is impractical [35].

In this work, a new 2-D/3-D method is investigated, which is intended to leverage both the computational efficiency of orthogonal-mesh S_N and the efficient mesh of the MoC to arrive at full-core, pin resolved reactor neutronics solutions.

1.5 About the Code Used in This Work

Most of the results presented in this work were obtained using one of two different pieces of software. The first, MPACT, is a MoC-based code developed jointly by the University

of Michigan and Oak Ridge National Laboratory (ORNL). The author is has contributed extensively to this project. It is an incredibly feature-rich, well-tested, production grade reactor analysis tool written in modern Fortran as part of the CASL initiative. While some early 2-D/3-D code was implemented in MPACT, the majority of the 2-D/3-D results presented here were obtained using MOCC [34], a different MoC code written in C++ almost solely by the author. The primary goals in the development of MOCC were to refine and simplify many aspects of MPACT, and to provide technical demonstrations of a number of software engineering techniques that make developing this kind of complicated scientific software much easier. In many ways, MOCC can be considered a surrogate to MPACT, as many aspects of its high-level architecture are inspired or derived in some way from MPACT.

MPACT was used to produce the 2-D/1-D results presented in Chapter 5, as well as the 2-D results that are presented in Section 3.3. MOCC was used to produce all other 2-D/3-D results. The mesh convergence studies discussed in Section 3.1 were produced with Denovo.

1.6 The Transport Equation

Neutron transport phenomena are governed by the Boltzmann transport equation. In the presence of multiplying (fissile) media, it is

$$\begin{aligned}
&\frac{1}{v}\frac{\partial\psi}{\partial t}(\mathbf{r},\hat{\mathbf{\Omega}},E,t)+\hat{\mathbf{\Omega}}\cdot\nabla\psi(\mathbf{r},\hat{\mathbf{\Omega}},E,t)+\Sigma_{\mathrm{t}}(\mathbf{r},E)\psi(\mathbf{r},\hat{\mathbf{\Omega}},E,t)\\
&\quad=\int_{4\pi}\int_{0}^{\infty}\Sigma_{\mathrm{s}}(\mathbf{r},\hat{\mathbf{\Omega}}'\cdot\hat{\mathbf{\Omega}},E'\rightarrow E)\psi(\mathbf{r},\hat{\mathbf{\Omega}}',E',t)dE'd\Omega'\\
&\qquad+\frac{\chi(\mathbf{r},E)}{4\pi}\int_{4\pi}\int_{0}^{\infty}(1-\beta(\mathbf{r},E'))\nu\Sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},\hat{\mathbf{\Omega}}',E',t)dE'd\Omega'\\
&\qquad+\sum_{k=1}^{N_{dg}}\frac{\chi_k(\mathbf{r},E)}{4\pi}C_k(\mathbf{r},t)\lambda_k+\frac{1}{4\pi}Q(\mathbf{r},E,t),
\end{aligned}\tag{1.1a}$$

$$\frac{\partial C_k}{\partial t}(\mathbf{r},t) = \int_{4\pi}\int_0^\infty \beta_k(\mathbf{r},E')\nu\Sigma_\mathrm{f}(\mathbf{r},E')\psi(\mathbf{r},\hat{\mathbf{\Omega}}',E',t)dE'd\Omega' - \lambda_k C_k(\mathbf{r},t), \tag{1.1b}$$

subject to the boundary and initial conditions

$$\psi(\mathbf{r},\hat{\mathbf{\Omega}},E,t) = \psi^b(\mathbf{r},\hat{\mathbf{\Omega}},E,t), \quad \mathbf{r}\in\partial V, \quad \hat{\mathbf{\Omega}}\cdot\mathbf{n}<0, \quad 0<E<\infty, \quad 0<t, \tag{1.1c}$$

and

$$\begin{aligned}\psi(\mathbf{r},\hat{\mathbf{\Omega}},E,0) &= \psi^0(\mathbf{r},\hat{\mathbf{\Omega}},E), \quad \mathbf{r}\in V, \quad \hat{\mathbf{\Omega}}\in 4\pi, \quad 0<E<\infty,\\ C_k(\mathbf{r},0) &= C_k^0(\mathbf{r}), \quad \mathbf{r}\in V.\end{aligned} \tag{1.1d}$$

The above equations describe the process by which neutrons change in population in time, move (or "stream") through space, interact with their background medium, and are introduced into the system as the result of fission events or generic sources. The terms on the left hand side of Eq. (1.1a) represent losses of neutrons from a particular region of phase space, while the right-hand-side terms represent sources of neutrons. Here $\mathbf{r}$ is a position vector describing a location in space, $\hat{\mathbf{\Omega}}$ is a unit vector describing a direction in the unit sphere. The neutron angular flux, $\psi(\mathbf{r},\hat{\mathbf{\Omega}},E,t)$, is the variable for which solutions are desired. Neutron flux is defined as the product of the neutron density and velocity, or the rate at which neutrons trace out path length through space, and is typically presented in units of $\frac{\text{neutrons}}{\text{eV}\cdot\text{cm}^2\cdot\text{str}\cdot\text{s}}$.

The terms Σ_f, Σ_s and Σ_t represent the macroscopic cross section for fission, scattering and all collisions, respectively. More precisely, each represents the probability per unit path length that a neutron traveling at location $\mathbf{r}$ and energy E will undergo the subscripted reaction. While in reality, the composition of the background medium changes in time, affecting the cross sections, it is typically assumed that they vary slowly enough with respect to the flux that their variation is negligible. In the context of reactor analysis, calculations in which the cross sections and nuclide densities are allowed to vary are called "deple-

tion" calculations, which still typically assume roughly stationary cross sections for each time step. The factor $\nu(\mathbf{r}, E)$ represents the number of new neutrons that are released as the result of a fission event initiated by a neutron traveling with energy E, and $\chi(\mathbf{r}, E)$ is a probability distribution function describing the likelihood of a neutron born from fission having an energy within dE about E.

The term

$$\sum_{k=1}^{N_{dg}} \frac{\chi_k(\mathbf{r}, E)}{4\pi} C_k(\mathbf{r}, t)\lambda_k$$

represents the "delayed" neutron source. Not all neutrons born from fission are emitted immediately; instead they are bound within fission products, or "delayed neutron precursors," to be emitted later as by decay. Since it is intractable to consider all delayed neutron precursors individually, they are instead lumped into N_{dg} delayed neutron groups, each with their own concentrations, C_k, decay constants, λ_k, and yields, β_k. Typically, six delayed neutron groups are used [27]. The complement of the total delayed neutron yield,

$$\beta = \sum_{k=1}^{N_{dg}} \beta_k, \tag{1.2}$$

is used in Eq. (1.1a) as the prompt fission neutron yield. Eqs. (1.1b) describe the rate of change of the delayed neutron precursor densities as new precursors are produced by fission and removed by decay.

It is often useful to employ the definition of the angularly-independent *scalar flux*,

$$\phi(\mathbf{r}, E, t) \equiv \int_{4\pi} \psi(\mathbf{r}, \hat{\mathbf{\Omega}}, E, t) d\Omega, \tag{1.3}$$

which is useful in simplifying the isotropic terms in the transport equation.

1.6.1 Eigenvalue Form of the Transport Equation

Equations (1.1) are very general, but difficult to solve in their original form. In particular, when not analyzing transient behavior of a reactor, an analyst is usually more interested in the quasi-steady state of a reactor. To simplify such analyses, the time dependence is removed by assuming that the delayed neutron precursor densities are in equilibrium and applying a scaling factor, $1/k$, to the fission source term and removing the inhomogeneous source, Q, to arrive at

$$\begin{aligned}
&\hat{\mathbf{\Omega}}\cdot\nabla\psi(\mathbf{r},\hat{\mathbf{\Omega}}',E)+\Sigma_{\mathrm{t}}\psi(\mathbf{r},\hat{\mathbf{\Omega}},E)\\
&\quad=\int_{4\pi}\int_{0}^{\infty}\Sigma_{\mathrm{s}}(\mathbf{r},\hat{\mathbf{\Omega}}'\cdot\hat{\mathbf{\Omega}},E'\rightarrow E)\psi(\mathbf{r},\hat{\mathbf{\Omega}}',E)dE'd\Omega'\\
&\quad+\frac{1}{k}\frac{\chi(\mathbf{r},E)}{4\pi}\int_{0}^{\infty}\nu\Sigma_{\mathrm{f}}(\mathbf{r},E')\phi(\mathbf{r},E')dE',
\end{aligned}\tag{1.4a}$$

subject to the boundary condition

$$\psi(\mathbf{r},\hat{\mathbf{\Omega}},E)=0,\quad \mathbf{r}\in\partial V,\quad \hat{\mathbf{\Omega}}\cdot\mathbf{n}<0,\quad 0<E<\infty.\tag{1.4b}$$

Equation (1.4a) is called the eigenvalue form of the transport equation. The value k is the multiplication factor of the system, where a value of unity means that neutrons introduced into the system are in natural balance with neutrons lost to leakage or absorption. As k deviates from unity, the fission source is being artificially scaled to enforce balance. In a super-critical system, where $k>1$, the neutron population would be increasing in time, thus requiring that the fission source be reduced. In a sub-critical system, where $k<1$, the fission source is scaled up to achieve balance. The value of k, often referred to as the eigenvalue, for a particular reactor configuration is very important from a design perspective and is therefore one of the key results from a reactor neutronics simulation.

1.6.2 Representing the Scattering Integral

In its original form, the scattering integral in the transport equation is easy to understand conceptually, but difficult to manipulate mathematically. It is therefore common to represent the scattering cross section as a series of Legendre polynomials. Legendre polynomials form an orthogonal set of functions defined on the interval $-1 \leq \mu \leq 1$, making them well suited for representing the scattering cross section, which depends angularly on the scattering cosine, μ. Employing an infinite series expansion of the scattering source gives the representation

$$\Sigma_s(\mathbf{r},\mu) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \Sigma_{s,n}(\mathbf{r}) P_n(\mu), \tag{1.5}$$

where $P_n(\mu)$ is is the nth-order Legendre polynomial. $\Sigma_{s,n}(\mathbf{r})$ is the nth Legendre moment of scattering cross section, defined as

$$\Sigma_{s,n}(\mathbf{r}) = 2\pi \int_{-1}^{1} P_n(\mu') \Sigma_s(\mathbf{r},\mu') d\mu'. \tag{1.6}$$

The infinite series represented above is exact, though in practice the series is truncated at some order N:

$$\Sigma_s(\mathbf{r},\mu) \approx \sum_{n=0}^{N} \frac{2n+1}{4\pi} \Sigma_{s,n}(\mathbf{r}) P_n(\mu), \tag{1.7}$$

For example, the case of $N = 0$ assumes isotropic scattering, while $N = 1$ assumes linearly anisotropic scattering. The required P_n order to achieve accurate solutions depends upon the nature of the system being simulated; the target nuclide and neutron energy determine the scattering behavior, which may vary between isotropic and highly-anisotropic. All of the methods in this work could be extended to treat anisotropic scattering; this work is restricted to isotropic scattering for simplicity.

1.7 Document Outline

The remainder of this document is devoted to describing a new 2-D/3-D method for performing 3-D, full-core, pin-resolved reactor analysis and exploring its effectiveness. The document is divided into this chapter and five others.

Chapter 2 introduces a number of discretizations to the transport equation that will be necessary to describe the 2-D/3-D method. This includes the multi-group and discrete ordinates approximations, which treat the energy and angular variables, respectively. The Method of Characteristics and S_N methods for discretizing the spatial domain are both described, as they are important components of the 2-D/3-D method and many 2-D/1-D methods. Since the 2-D/3-D and 2-D/1-D methods are so similar, the 2-D/1-D method is also described for context, and to better compare and contrast the two methods. Finally, since it has become such an important component of many reactor analysis codes, the Coarse Mesh Finite Difference (CMFD) acceleration technique is described.

Chapter 3 introduces the Corrected Diamond Difference concept and applies it to a number of 2-D scenarios. It demonstrates that Corrected Diamond Difference (CDD) is capable of producing equivalent solutions on a coarsened, orthogonal mesh to the fine-mesh MoC used to calculate correction factor and cross section data.

The 2-D/3-D method is introduced in Chapter 4, where the CDD equations are extended to 3-D problems. Two coupling methods are discussed: one in which correction factors and cross sections are calculated by multiple separate 2-D MoC eigenvalue calculations and stitched together into a 3-D problem. The other couples a stack of MoC sweepers and the 3-D S_N sweeper using a transverse leakage term, similarly to 2-D/1-D.

In Chapter 5, the methods described in Chapter 4 are applied to the C5G7 benchmark, and the results are discussed, analyzed, and compared to a pair of 2-D/1-D methods for context. The various strengths and weaknesses of the 2-D/3-D method are presented.

Chapter 6 contains a brief summary of the work, final conclusions, and a healthy list of topics for possible future work.

CHAPTER 2

Discretization of the Transport Equation and Solution Techniques

While methods exist to solve the transport equation as a continuous function in space and/or angle, this work is concerned mostly with methods that discretize phase space entirely. This section discusses the various discretization schemes used to treat the energy, angle and spatial domains to arrive at the forms of the transport equation that are relevant to this work. Particular attention is paid to the S_N and MoC methods.

2.1 Multi-Group Approximation

The energy domain is discretized using the multi-group approximation, which applies a one-dimensional grid to the energy domain. Each term in the continuous-energy transport equation is then represented as energy-averaged quantities over N_G energy "groups," which are bounded by $N_G + 1$ energy bounds, $E_0 = E_{max} > E_1 > \cdots > E_g > \cdots > E_{N_G-1} > E_{N_G} = E_{min}$. Conventionally, the energy groups are ordered in descending energy, with the range $E_1 < E < E_0$ being the highest energy group and $E_G < E < E_{G-1}$ being the lowest. Throughout this text, G is used to refer to the set of all energy groups. The following group-wise quantities are also defined as the integral of their respective energy group intervals:

$$\psi_g(\mathbf{r},\hat{\mathbf{\Omega}}) = \int_{E_g}^{E_{g-1}} \psi(\mathbf{r},\hat{\mathbf{\Omega}},E)dE, \tag{2.1a}$$

$$\chi_g(\mathbf{r}) = \int_{E_g}^{E_{g-1}} \chi(\mathbf{r},E)dE. \tag{2.1b}$$

The cross-section data must also be treated in a similar way, with the added the constraint that it is desired to preserve the reaction rates of the continuous-energy transport equation. The definition of the cross sections are therefore flux-weighted such that the product of the group-wise flux and group-wise cross section yields the integral of the continuous-energy reaction rate. For a generic reaction, b, the group-wise cross section is defined as

$$\Sigma_{b,g}(\mathbf{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_{b,g}(\mathbf{r},E) \int_{4\pi} \psi(\mathbf{r},\hat{\mathbf{\Omega}},E)d\Omega dE}{\int_{E_g}^{E_{g-1}} \int_{4\pi} \psi(\mathbf{r},\hat{\mathbf{\Omega}},E)d\Omega dE}. \tag{2.2}$$

The scattering cross section requires special attention, and is defined as

$$\Sigma_{s,g'g}(\mathbf{r},\hat{\mathbf{\Omega}}'\cdot\hat{\mathbf{\Omega}}) = \frac{\int_{E_g}^{E_{g-1}} \int_{E'_g}^{E_{g'-1}} \Sigma_s(\mathbf{r},E'\rightarrow E,\hat{\mathbf{\Omega}}'\cdot\hat{\mathbf{\Omega}})\phi(\mathbf{r},E')dE'dE}{\int_{E_{g'}}^{E_{g'-1}} \phi(\mathbf{r},E')dE'} \tag{2.3}$$

These spectrum-weighted reaction cross sections of course rely upon *a priori* knowledge of the continuous-energy flux spectrum. Typically, a spectrum is assumed to facilitate the generation of a cross section library, which contains suitable group constants.

Integrating Eq. (1.4a) over the energy range $E_{N_G} < E < E_{N_G-1}$ and employing the definitions above, the multi-group form of the k-eigenvalue transport equation is obtained:

$$\begin{aligned}\hat{\mathbf{\Omega}}\cdot\nabla\psi_g(\mathbf{r},\hat{\mathbf{\Omega}}) + \Sigma_{\mathrm{t},g}\psi_g(\mathbf{r},\hat{\mathbf{\Omega}}) &= \sum_{g'\in G}\int_{4\pi}\Sigma_{\mathrm{s,g',g}}(\mathbf{r},\hat{\mathbf{\Omega}}'\cdot\hat{\mathbf{\Omega}})\psi_{g'}(\mathbf{r},\hat{\mathbf{\Omega}}')d\Omega' \\ &+ \frac{1}{k}\frac{\chi_g(\mathbf{r})}{4\pi}\sum_{g'\in G}\nu\Sigma_{\mathrm{f},g'}(\mathbf{r})\phi_{g'}(\mathbf{r}).\end{aligned} \tag{2.4}$$

2.2 Angular Discretization

As with other components of phase space, there are numerous approaches to treating the angular variable. This section describes two such approaches, which are particularly relevant to this work: diffusion and discrete ordinates.

2.2.1 Diffusion Approximation

The diffusion approximation is not actually an angular discretization method, but rather assumes that the angular flux is well-approximated as the linear function in angle,

$$\psi(\mathbf{r}, \hat{\mathbf{\Omega}}, E) \approx \frac{1}{4\pi}\left(\phi(\mathbf{r}, E) + 3\mathbf{J}\cdot\hat{\mathbf{\Omega}}\right). \tag{2.5}$$

With linearly-anisotropic scattering, the multi-group form of the transport equation can be written as

$$\begin{aligned}\psi_g(\mathbf{r}, \hat{\mathbf{\Omega}}) + \hat{\mathbf{\Omega}}\cdot\nabla\psi_g(\mathbf{r}, \hat{\mathbf{\Omega}}) + \Sigma_{t,g}(\mathbf{r})\psi_g(\mathbf{r}, \hat{\mathbf{\Omega}}) = \\ \frac{1}{4\pi}\sum_{g'\in G}\left(\Sigma_{s0,g'g}(\mathbf{r})\phi_{g'}(\mathbf{r}) + 3\Sigma_{s1,g'g}(\mathbf{r})\left(\eta J_{x,g'}(\mathbf{r}) + \varepsilon J_{y,g'}(\mathbf{r}) + \mu J_{z,g'}(\mathbf{r})\right)\right) \\ + \frac{1}{k}\frac{\chi_g(\mathbf{r})}{4\pi}\left(\sum_{g'\in G}\nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r})\right).\end{aligned} \tag{2.6}$$

The J_x, J_y, and J_z terms are the x, y, and z components of the neutron current, or the first angular moments of the angular flux. Integrating Eq. (2.6) over all angles gives the zeroth angular moment of the transport equation,

$$\begin{aligned}\frac{\partial}{\partial x}J_{x,g}(\mathbf{r}) + \frac{\partial}{\partial y}J_{y,g}(\mathbf{r}) + \frac{\partial}{\partial z}J_{z,g}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r})\phi_g(\mathbf{r}) = \\ \sum_{g'\in G}\Sigma_{s0,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \frac{1}{k}\frac{\chi_g(\mathbf{r})}{4\pi}\sum_{g'\in G}\nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}).\end{aligned} \tag{2.7}$$

Similarly, multiplying Eq. (2.6) by angle and integrating over all angles gives the first angular moment of the transport equation:

$$\frac{1}{3}\frac{\partial \phi_g}{\partial x}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{x,g}(\mathbf{r}) = \sum_{g' \in G} \Sigma_{s1,g'g}(\mathbf{r}) J_{x,g'}(\mathbf{r}), \tag{2.8a}$$

$$\frac{1}{3}\frac{\partial \phi_g}{\partial y}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{y,g}(\mathbf{r}) = \sum_{g' \in G} \Sigma_{s1,g'g}(\mathbf{r}) J_{y,g'}(\mathbf{r}), \text{ and} \tag{2.8b}$$

$$\frac{1}{3}\frac{\partial \phi_g}{\partial z}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{z,g}(\mathbf{r}) = \sum_{g' \in G} \Sigma_{s1,g'g}(\mathbf{r}) J_{z,g'}(\mathbf{r}). \tag{2.8c}$$

To arrive at the neutron diffusion equation, a couple of approximations are made. First, the derivative of the neutron currents with respect to time are neglected. Second, the first-order scattering cross sections, $\Sigma_{s1,g'g}$, which would otherwise form a $G \times G$ matrix, are approximated by a diagonal matrix, $\bar{\Sigma}_{s1}$, with elements:

$$\bar{\Sigma}_{s1,g} = \sum_{g' \in G} \Sigma_{s1,g'g}. \tag{2.9}$$

Applying these approximations to Eqs. (2.8) gives

$$\frac{1}{3}\frac{\partial \phi_g}{\partial x}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{x,g}(\mathbf{r}) = \bar{\Sigma}_{s1,g}(\mathbf{r}) J_{x,g}(\mathbf{r}), \tag{2.10a}$$

$$\frac{1}{3}\frac{\partial \phi_g}{\partial y}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{y,g}(\mathbf{r}) = \bar{\Sigma}_{s1,g}(\mathbf{r}) J_{y,g}(\mathbf{r}), \text{ and} \tag{2.10b}$$

$$\frac{1}{3}\frac{\partial \phi_g}{\partial z}(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r}) J_{z,g}(\mathbf{r}) = \bar{\Sigma}_{s1,g}(\mathbf{r}) J_{z,g}(\mathbf{r}). \tag{2.10c}$$

Solving for current using the above equations gives

$$\mathbf{J}(\mathbf{r}) = -\frac{1}{3\left(\Sigma_{t,g}(\mathbf{r}) - \bar{\Sigma}_{s1,g}(\mathbf{r})\right)} \nabla \phi_g(\mathbf{r}). \tag{2.11}$$

The value $\frac{1}{3\left(\Sigma_{t,g}(\mathbf{r})-\bar{\Sigma}_{s1,g}(\mathbf{r})\right)}$ is often referred to as the diffusion coefficient, $D_g(\mathbf{r})$. Using this definition, the expression for the current becomes

$$\mathbf{J} = -D_g(\mathbf{r})\nabla\phi_g(\mathbf{r}), \tag{2.12}$$

which is Fick's Law applied to neutron diffusion.

Using Eq. (2.12) to represent the streaming term in Eq. (2.7) yields the neutron diffusion equation:

$$\begin{aligned} &-\nabla\cdot D_g(\mathbf{r})\nabla\phi_g(\mathbf{r}) + \Sigma_{t,g}(\mathbf{r})\phi_g(\mathbf{r}) \\ &\quad = \sum_{g'\in G}\Sigma_{s0,g'g}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \frac{1}{k}\frac{\chi_g(\mathbf{r})}{4\pi}\sum_{g'\in G}\nu\Sigma_{f,g'}(\mathbf{r})\phi_{g'}(\mathbf{r}). \end{aligned} \tag{2.13}$$

The diffusion equation functions as a good approximation under specific circumstances; namely, when the solution to the transport equation can be considered "diffusive." This usually holds in regions sufficiently far (on the order of several mean free paths) away from strong material discontinuities, boundary conditions, or sources, and when the background medium is not too strongly-absorbing.

2.2.2 Discrete Ordinates Approximation

The discrete ordinates approximation represents the continuous angular space with a discrete set of angles. Directions of neutron travel are represented by the unit vector $\hat{\mathbf{\Omega}}$, which can be localized on the unit sphere using a polar angle, θ, and azimuthal angle, φ as depicted in Fig. 2.1.

When working with these angles, as in the streaming term of the transport equation, it is useful to express them as unit vectors in Cartesian coordinates,

$$\hat{\mathbf{\Omega}} = \Omega_x\hat{\mathbf{i}} + \Omega_y\hat{\mathbf{j}} + \Omega_z\hat{\mathbf{k}}, \tag{2.14}$$

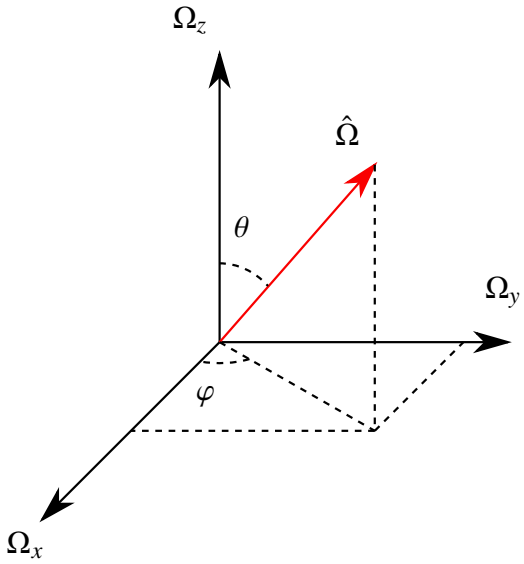


Figure 2.1: Representation of angles in the unit sphere.

where the components of $\hat{\mathbf{\Omega}}$ are expressed as

$$\Omega_x = \eta = \sqrt{1-\mu^2}\cos(\varphi) \tag{2.15a}$$

$$\Omega_y = \varepsilon = \sqrt{1-\mu^2}\sin(\varphi) \tag{2.15b}$$

$$\Omega_z = \mu = \cos(\theta). \tag{2.15c}$$

To discretize the angular variable a set of discrete angles is selected,

$$\hat{\mathbf{\Omega}}^n = \hat{\mathbf{\Omega}}(\theta^n, \varphi^n), \quad n = 1, 2, \ldots, N \tag{2.16}$$

along with corresponding weights, w_n. Such sets of angles and weights are referred to as angular quadrature sets. The angles, $\hat{\mathbf{\Omega}}^n$, and weights, w_n, are selected to result in accurate angular integrals approximated by quadrature,

$$\int_{4\pi} f(\hat{\mathbf{\Omega}}) d\Omega \approx \sum_{n=1}^{N} w_n f(\hat{\mathbf{\Omega}}^n). \tag{2.17}$$

The angular flux variable is then constrained to these discrete angles,

$$\psi^n(\mathbf{r}) = \psi(\mathbf{r}, \hat{\mathbf{\Omega}}^n). \tag{2.18}$$

Using this quadrature set, angular integrals in the transport equation and elsewhere are approximated by quadrature summation. For example, the scalar flux can be represented as

$$\phi(\mathbf{r}) = \int_{4\pi} \psi(\mathbf{r}, \hat{\mathbf{\Omega}}) d\Omega \approx \sum_{n=1}^{N} w_n \psi^n(\mathbf{r}). \tag{2.19}$$

Applying this treatment to Eq. (2.4) results in the discrete ordinates form of the multi-group transport equation,

$$\hat{\mathbf{\Omega}}^n \cdot \nabla \psi_g^n(\mathbf{r}) + \Sigma_{\mathrm{t},g}(\mathbf{r}) \psi_g^n(\mathbf{r}) = \sum_{g' \in G} \sum_{n' \in N} w_n \Sigma_{\mathrm{s,g',g}}^{n'n}(\mathbf{r}) \psi_g^n(\mathbf{r}) + \frac{1}{k} \frac{\chi_g(\mathbf{r})}{4\pi} \sum_{g' \in G} \nu \Sigma_{\mathrm{f},g'}(\mathbf{r}) \sum_{n \in N} \psi_{g'}^n(\mathbf{r}) w_n. \tag{2.20}$$

In the above, a discrete form of the scattering cross section, $\Sigma_{\mathrm{s,g',g}}^{n'n}$, is used, which is defined as

$$\Sigma_{\mathrm{s,g',g}}^{n'n}(\mathbf{r}) = \Sigma_{s,g'g}(\mathbf{r}, \hat{\mathbf{\Omega}}^{n'} \cdot \hat{\mathbf{\Omega}}^n). \tag{2.21}$$

Various quadrature sets have been developed, each with different properties. The primary requirement for any set is that the weights sum to 4π thereby preserving the integral of constant functions of angle on the unit sphere. In practice, the specific directions and corresponding weights in a particular quadrature set are chosen in such a way that they properly integrate polynomial functions on the unit sphere as accurately as possible.

Historically, S_N methods (described in Section 2.4.1) have used "level symmetric" quadratures, which have the property that the direction components maintain symmetry about the coordinate axes under 90° rotations, as can be seen in Fig. 2.2. Level-symmetric quadratures are named like S_N, where N is the "order" of the quadrature, which refers to the number of levels in the set of directions between two poles. For example, the quadrature

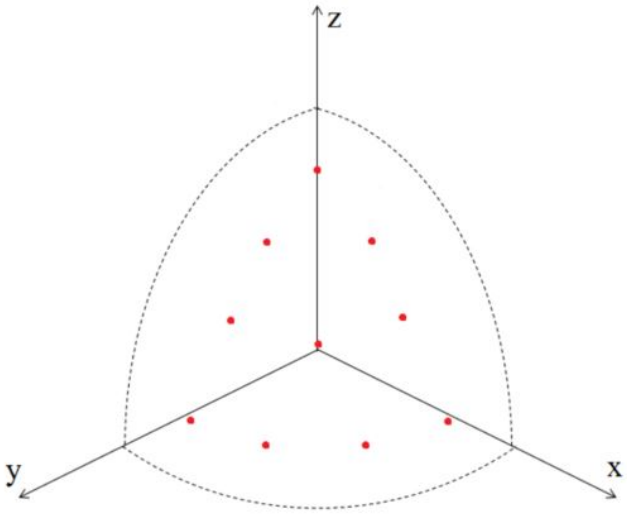


Figure 2.2: A typical level-symmetric quadrature, in this case S_8.

in Fig. 2.2 is S_8, with the four levels in the positive-z half-space being depicted. Level-symmetric quadratures aim to place no emphasis on any particular direction, which is why they tend to be useful for general 3-D problems. The S_N method, described below uses such quadratures so often that it has assumed their name.

In some situations, such as reactor simulations, it becomes beneficial to construct a quadrature that treats different directions in space differently. With reactors, containing many cylindrical structures with parallel axes, it is useful to align the pole of the angular coordinate system with the axes of the fuel pins, then sample the azimuthal angles more finely than the polar angles. To do this, "product quadratures" are used instead of level-symmetric quadratures. A product quadrature is formed by all combinations of the angles in two separate quadratures which treat the azimuthal and polar angles, respectively. Figure 2.3 depicts a product quadrature with four azimuthal angles and three polar angles per octant. The final weights of each angle in the product quadrature is the product of the weights from the corresponding angles in the source azimuthal and polar quadratures, hence the name. As an example, to calculate the scalar flux, $\phi(\mathbf{r})$, the following summation

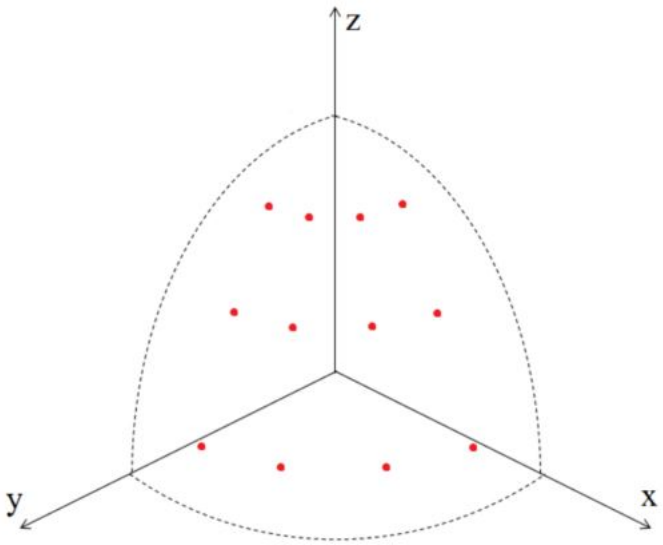


Figure 2.3: A typical product quadrature with azimuthal and three polar angles per octant.

would be used:

$$\phi(\mathbf{r}) = \sum_{n=1}^{N_{\text{polar}}} \sum_{m=1}^{N_{\text{azi}}} \psi(\mathbf{r}, \theta^n, \varphi^m) w_n w_m. \tag{2.22}$$

Typical azimuthal quadratures are evenly-spaced angles in $(0, 2\pi)$ with equal weights, often referred to as a Chebyshev quadrature. Polar quadratures are usually Gauss-Legendre or Yamamoto [33] quadratures. Since the final product quadrature weights must sum to 4π, the weights of the polar quadrature are normalized to sum to 2, while the azimuthal quadrature weights are normalized to sum to 2π, reflecting their respective domains.

Not only do product quadratures more efficiently discretize the angular variable in the context of reactor analysis, but they permit a number of computational optimizations. These optimizations usually leverage the fact that a 2-D solver may in certain contexts ignore the presence of multiple polar angles. For instance, in 2-D MoC, rays may be traced only for the azimuthal angles, saving time and memory, and the sweep algorithms can solve all polar angles in parallel as they sweep a single ray.

2.3 Source Iteration

Many methods of solving the transport equation rely upon the concept of *source iteration*. Source iteration is a technique by which the scattering and fission source terms of the transport equation, rather than being calculated implicitly, are converged iteratively. This allows the fission and scattering sources to be incorporated into an inhomogeneous source term, greatly simplifying the equation to be solved for each iteration. A number of sophisticated methods exist to achieve better approximations of the source terms [1, 11, 28]. However, a relatively straightforward approach was used in this work (in both MOCC and MPAC) and is described below.

Starting with the multi-group, discrete ordinates transport equation, the fission and scattering sources are referred to separately as F and S, respectively,

$$\hat{\mathbf{\Omega}}^n \cdot \nabla \psi_g^n + \Sigma_t \psi_g^n = \frac{1}{4\pi} S_g + \frac{1}{4\pi} \chi_g F. \tag{2.23}$$

The fission source is calculated using the current estimate of the system eigenvalue, k_{eff}, as

$$F = \frac{1}{k_{\text{eff}}} \sum_{g=1}^{G} \nu \Sigma_{f,g} \phi_g \tag{2.24}$$

When using the multi-group approximation, it is often useful to separate the scattering source into a self-scattering source, S_{self}, a down-scattering source, S_{down}, and an up-scattering source S_{up}[1].

$$\hat{\mathbf{\Omega}} \cdot \nabla \psi + \Sigma_t \psi = \frac{1}{4\pi} \left(S_{\text{self}} + S_{\text{up}} + S_{\text{down}} + \chi_g F \right). \tag{2.25}$$

A fixed-point iteration scheme is then used in which the in-scattering and fission sources are calculated using previous-iteration estimates of the flux.

[1] Self-scattering refers to neutrons that scatter, but do not change energy group. Down-scattering refers to neutrons that scatter into the current group from higher energy groups and up-scattering refers to neutrons scattering up from lower energy groups.

For a particular iteration, ℓ, it is assumed that the fission source and k_{eff} have been updated either by power iteration or some other iterative eigenvalue technique using the previous iteration flux. A loop is then performed over the energy groups, from high energy to low energy. The scattering sources S_{down} and S_{up} are usually handled in one of two ways. In the first, both in-scattering sources are obtained using the previous-iteration scalar flux, giving

$$\hat{\mathbf{\Omega}} \cdot \nabla \psi^{\ell} + \Sigma_{\text{t}} \psi^{\ell} = S^{\ell}_{\text{self}} + S^{\ell-1}_{\text{down}} + S^{\ell-1}_{\text{up}} + F^{\ell-1}, \tag{2.26}$$

which is essentially a Jacobi iteration on the scattering source. Alternatively, since the energy sweep loops from high to low energy, advantage may be taken of the current-iteration estimate of the flux for higher energy groups to get a more up-to-date down-scatter source:

$$\hat{\mathbf{\Omega}} \cdot \nabla \psi^{\ell} + \Sigma_{\text{t}} \psi^{\ell} = S^{\ell}_{\text{self}} + S^{\ell}_{\text{down}} + S^{\ell-1}_{\text{up}} + F^{\ell-1}. \tag{2.27}$$

This constitutes a Gauss-Seidel style iteration on the scattering source. In the bulk of the energy domain takes place overwhelmingly from higher energy groups to lower, since it only becomes possible for neutrons to to scatter up in energy when they are close to thermal energies. As a result, Gauss-Seidel iteration very nearly constitutes a direct inversion of the scattering operator, and tends to perform well. Sometimes in the thermal energy range, where up-scattering is more common, it is beneficial to loop over this range several times to help converge the up-scatter source without having to revisit the higher-energy groups. These are called "up-scatter iterations."

Within a given energy group, it is also common to perform multiple transport sweeps, updating the self-scattering source each time. There are typically referred to as "inner" iterations.

2.4 Spatial Discretization

While the discretization of the energy and angular variables are important, the multi-group and discrete ordinates approximations are very well understood. This work is concerned mostly with methods used for treating the spatial variable. The two spatial discretization methods most relevant to the 2-D/3-D method are S_N and the MoC, which are described below.

2.4.1 Discrete Ordinates (S_N) Method

The S_N method is a simple approach for solving the multi-group, discrete ordinates form of the transport equation discussed in Section 1.6. In this derivation of the S_N method, the geometry is restricted to a structured Cartesian grid. A single element of such a grid is depicted in Fig. 2.4. Any region on the mesh, $V_{i,j,k}$, is bounded by $x_{i-1/2} < x \leq x_{i+1/2}$, $y_{j-1/2} < y \leq y_{j+1/2}$ and $z_{k-1/2} < z \leq z_{k+1/2}$. The cross section, Σ_t and the source, Q are both assumed to be constant within each mesh region.

To simplify notation, a modified version of Eq. (2.20) is used, in which the fission and scattering sources are combined into an inhomogeneous source, $Q_g^n(\mathbf{r})$, leaving behind only the streaming and reaction loss terms thusly,

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y} + \mu^n \frac{\partial}{\partial z} + \Sigma_{t,g}(\mathbf{r})\right)\psi_g^n(\mathbf{r}) = Q_g^n(\mathbf{r}), \tag{2.28a}$$

$$Q_g^n(\mathbf{r}) = \frac{\chi_g}{4\pi k}(\mathbf{r}) \sum_{g' \in G} \nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + \sum_{g' \in G}\sum_{n' \in N} \Sigma_{s,g',g}(\mathbf{r}, \hat{\mathbf{\Omega}}^n \cdot \hat{\mathbf{\Omega}}^{n'}) w^{n'} \psi_{g'}^{n'}(\mathbf{r}), \tag{2.28b}$$

where η^n, ε^n and μ^n are the $x-$, $y-$ and $z-$components of the $n-$th discrete angle in our angular quadrature. To further simplify notation, the energy group index, g, is omitted. Since it is desired to find the cell-averaged flux for each cell, the equation is integrated over

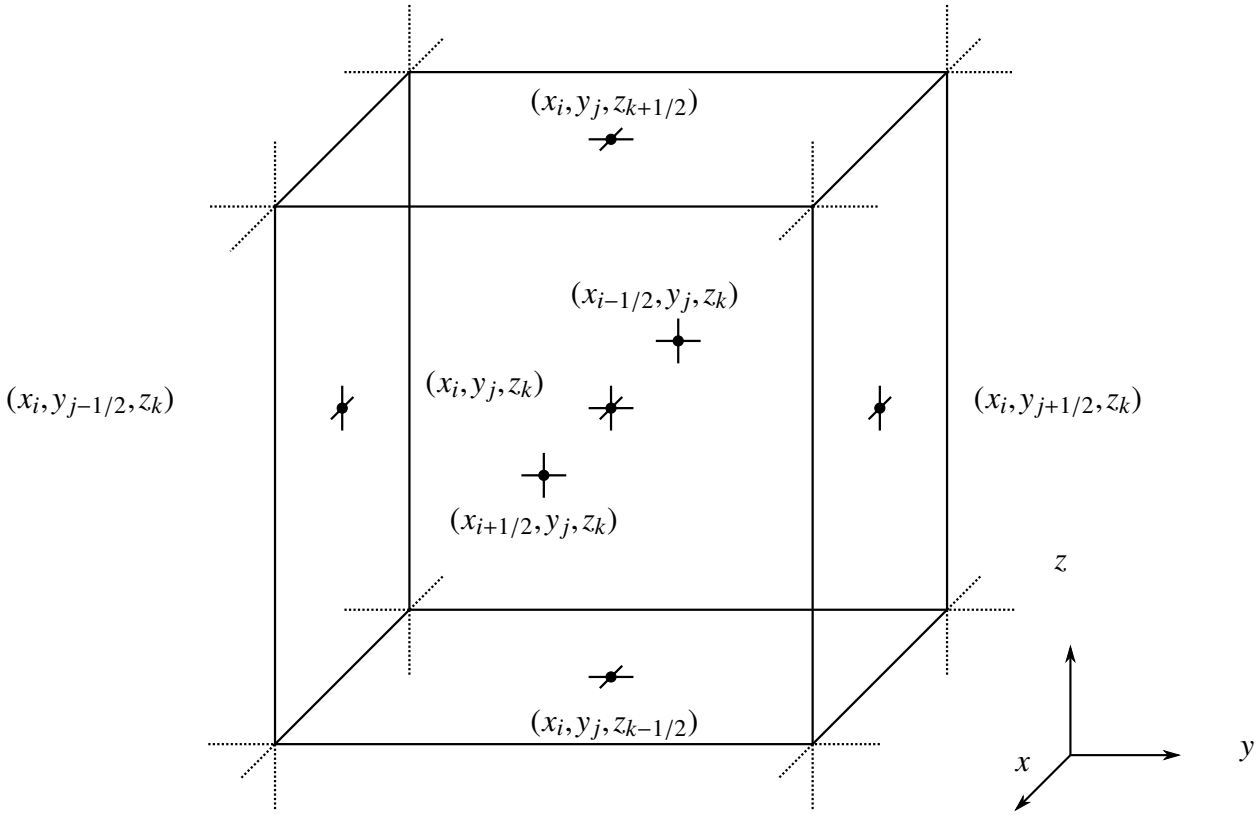


Figure 2.4: Element from a structured Cartesian mesh.

the volume of the mesh element, giving

$$\begin{aligned}
&\eta^n \int_k \int_j \left[\psi^n(x_{i\pm1/2},y,z) - \psi^n(x_{i\mp1/2},y,z)\right] dydz \\
&+ \varepsilon^n \int_k \int_i \left[\psi^n(x,y_{j\pm1/2},z) - \psi^n(x,y_{j\mp1/2},z)\right] dxdz \\
&+ \mu^n \int_j \int_i \left[\psi^n(x,y,z_{k\pm1/2}) - \psi^n(x,y,z_{k\mp1/2})\right] dxdy \\
&+ \Sigma_{t,ijk} \int_k \int_j \int_i \psi(x,y,z) dxdydz = \int_k \int_j \int_i Q(x,y,z) dxdydz.
\end{aligned} \tag{2.29}$$

Here the following shorthand was used:

$$\int_i (\cdot) dx = \int_{x_{i-1/2}}^{x_{i+1/2}} (\cdot) dx \qquad \int_j (\cdot) dy = \int_{y_{j-1/2}}^{y_{j+1/2}} (\cdot) dy \qquad \int_k (\cdot) dz = \int_{z_{k-1/2}}^{z_{k+1/2}} (\cdot) dz.$$

The material properties have also been assumed to be constant within a mesh region. Defining the node- and surface-averaged quantities

$$\begin{aligned}
\psi^n_{i\pm1/2,j,k} &= \frac{1}{\Delta y_j \Delta z_k} \int_j \int_k \psi_n(x_{i\pm1/2},y,z) dydz \\
\psi^n_{i,j\pm1/2,k} &= \frac{1}{\Delta x_i \Delta z_k} \int_i \int_k \psi_n(x,y_{i\pm1/2},z) dxdz \\
\psi^n_{i,j,k\pm1/2} &= \frac{1}{\Delta x_i \Delta y_j} \int_i \int_j \psi_n(x,y,z_{i\pm1/2}) dxdy \\
\bar{\psi}^n_{i,j,k} &= \frac{1}{\Delta x_i \Delta y_j \Delta z_k} \int_k \int_j \int_i \psi_n(x,y,z) dxdydz,
\end{aligned} \tag{2.30}$$

Eq. 2.29 can be divided by the node volume to yield

$$\begin{aligned}
&\frac{\eta^n}{\Delta x_i}\left(\psi^n_{i+1/2,j,k} - \psi^n_{i-1/2,j,k}\right) + \frac{\varepsilon^n}{\Delta y_j}\left(\psi^n_{i,j+1/2,k} - \psi^n_{i,j-1/2,k}\right) + \\
&\frac{\mu^n}{\Delta z_k}\left(\psi^n_{i,j,k+1/2} - \psi^n_{i,j,k-1/2}\right) + \Sigma_{t,ijk}\bar{\psi}^n_{i,j,k} = Q_{i,j,k}.
\end{aligned} \tag{2.31}$$

For a given angle and energy group, the result is a single equation with seven unknowns. Some auxiliary relationships must therefore be introduced between the node- and surface-

averaged fluxes to arrive at a closed set of equations.

Many differencing schemes have been developed throughout the years, some of which will be discussed in Chapter 4. A common and simple closure of particular relevance to this work are the *Diamond Difference (DD)* equations [24]. They specify the node-averaged flux as the average of the upwind and downwind surface flux for each direction:

$$\begin{aligned}
\bar{\psi}^n_{i,j,k} &= \frac{1}{2}\left(\psi^n_{i+1/2,j,k} + \psi^n_{i-1/2,j,k}\right), \\
\bar{\psi}^n_{i,j,k} &= \frac{1}{2}\left(\psi^n_{i,j+1/2,k} + \psi^n_{i,j-1/2,k}\right), \text{ and} \\
\bar{\psi}^n_{i,j,k} &= \frac{1}{2}\left(\psi^n_{i,j,k+1/2} + \psi^n_{i,j,k-1/2}\right).
\end{aligned} \tag{2.32}$$

The above is essentially the Crank-Nicolson method in space.

To perform a sweep along each direction in the angular quadrature, the procedure begins with the upwind[2] boundary condition and propagates the flux through the spatial domain until the downwind edges of the mesh are reached. The order in which the cells of the mesh are swept depends upon the octant in which the discrete direction lies. Introducing Eqs. 2.32 into Eq. 2.31, eliminating the downwind fluxes and solving for $\psi^n_{i,j,k}$ yields

$$\begin{aligned}
\bar{\psi}^n_{ijk} &= \frac{\frac{2\eta^n}{\Delta x_i}\psi^n_{i\mp1/2,j,k} + \frac{2\varepsilon^n}{\Delta y_j}\psi^n_{i,j\mp1/2,k} + \frac{2\mu^n}{\Delta z_k}\psi^n_{i,j,k\mp1/2} + Q_{ijk}}{\sigma_{ijk} + \frac{2\eta^n}{\Delta x_i} + \frac{2\varepsilon^n}{\Delta y_j} + \frac{2\mu^n}{\Delta z_k}}, \\
\eta^n &\gtrless 0, \quad \varepsilon^n \gtrless 0, \quad \mu^n \gtrless 0.
\end{aligned} \tag{2.33}$$

The angular component sign conditionals apply to the indices of the flux terms corresponding to the appropriate angular component. For instance, when $\eta > 0$, $\psi_{i\pm1/2,j,k} \equiv \psi_{i+1/2,j,k}$, and so on. Following the evaluation of the node-averaged flux using the upwind node surface fluxes, the diamond-difference equations (2.32) are invoked to calculate the downwind

[2]Upwind and downwind are used to specify directions with respect to the direction of neutron travel. When considering any particular direction, downwind denotes the direction in which the neutrons are traveling, and upwind denotes the direction from which they are traveling.

flux,

$$\begin{aligned}
\psi^n_{i\pm1/2,j,k} &= 2\bar{\psi}^n_{ijk} - \psi^n_{i\mp1/2,j,k}, \quad \eta^n \gtrless 0, \\
\psi^n_{i,j\pm1/2,k} &= 2\bar{\psi}^n_{ijk} - \psi^n_{i,j\mp1/2,k}, \quad \varepsilon^n \gtrless 0, \text{and} \\
\psi^n_{i,j,k\pm1/2} &= 2\bar{\psi}^n_{ijk} - \psi^n_{i,j,k\mp1/2}, \quad \mu^n \gtrless 0
\end{aligned} \tag{2.34}$$

at which point the sweep proceeds to the next mesh element. This process is depicted in Fig. 2.5.

When using an orthogonal mesh, parallel decomposition of the S_N method can be carried out using the Koch-Baker-Alcouffe (KBA) [20] algorithm. This algorithm was developed in the 1990's at Los Alamos National Laboratory, and more recently has been demonstrated in Denovo to achieve very impressive parallel performance on modern computing hardware. Figure 2.6 shows the behavior of KBA under weak and strong scaling studies. Weak scaling involves increasing the size or complexity of a problem to maintain the same amount of work per processor as the number of processors is increased. In strong scaling, the amount of overall work is kept constant while the number of processors is increased, leading to a smaller amount of work per processor. Inter-process communication is usually the dominant driver in the scaling performance of parallel algorithms. As a result, strong scaling usually deviates more from ideal scaling than weak scaling, since the reduction in local work results in proportionally more time spent in communication. Less-than-idea Weak scaling tends to arise from load balancing efficiency, and is typically less severe than strong scaling, as can be seen in Fig. 2.6.

While at the time of this writing, the KBA algorithm is not implemented in MOCC, it is described here because parallel performance will be an important aspect of 2-D/3-D methods in production environments. KBA operates by decomposing the diagonal plane along which flux is propagated through a 3-D mesh. This planar wavefront starts as a single cell at the farthest-upwind cell of the mesh, but grows as (1, 3, 6, 10, ...) to some maximum

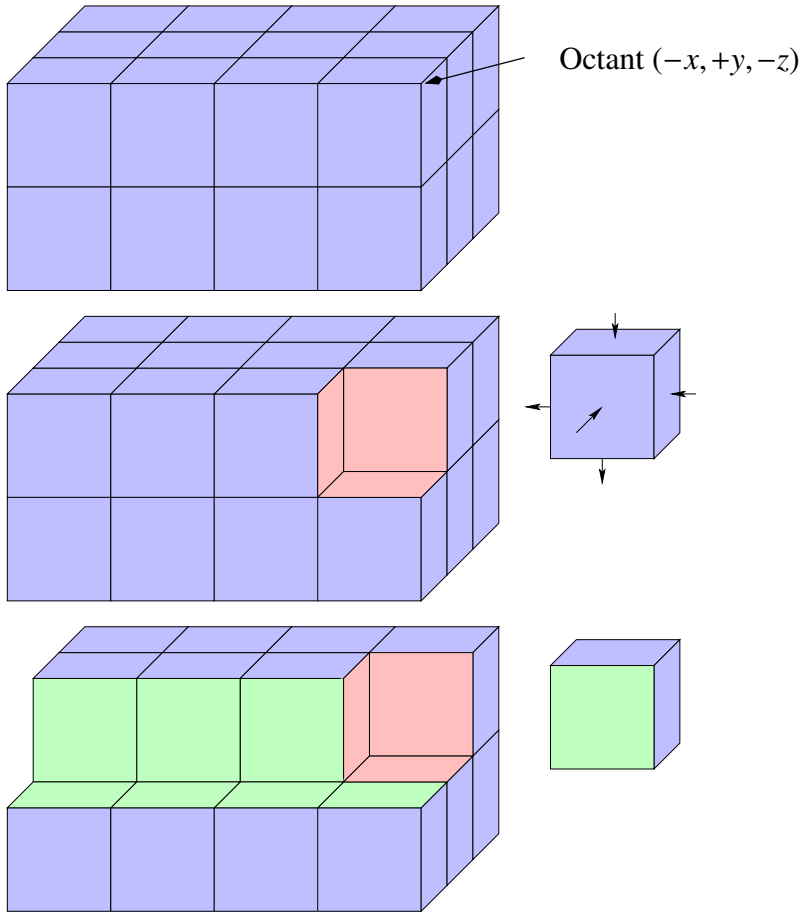
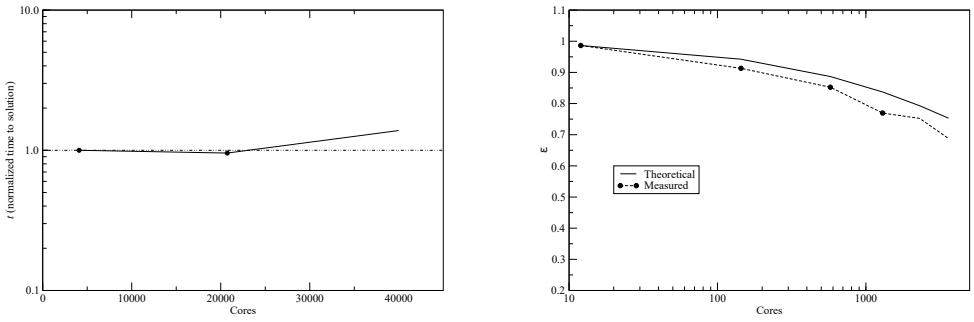


Figure 2.5: S_N sweep procedure for octant (-1, 1, -1) [10], used with author's permission. Blue surfaces contain boundary flux values, green surfaces have downwind flux calculated from sweeping a cell, and red surfaces have unknown flux, yet to be determined.



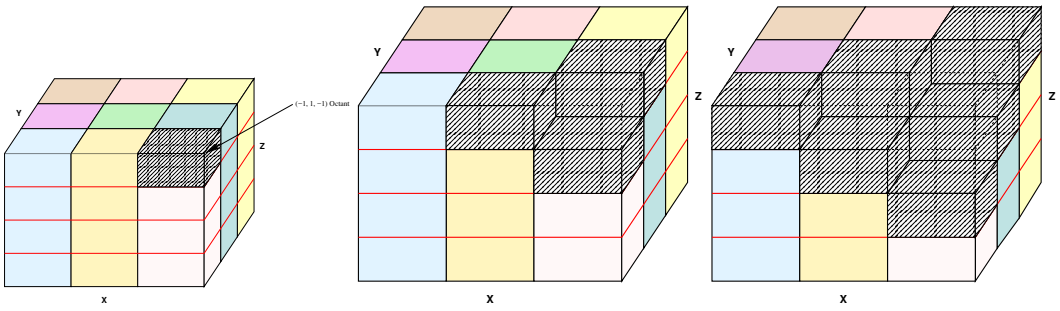
(a) Weak scaling.

(b) Strong scaling. ϵ is the scaling efficiency.

Figure 2.6: Scaling of the KBA algorithm in Denovo [10]. Used with author's permission. Strong scaling has an algorithm-dependent theoretical maximum efficiency, which accounts for the increased communication time incurred with increased parallelism. Less-than-ideal weak scaling arises from less efficient load balancing when more processors are used.

size in the interior of the mesh (this maximum plane size being dependent upon the mesh dimensions), before shrinking again at the most downwind corner of the mesh, as illustrated in Fig. 2.7.

Rather than decompose the entire 3-D wavefront, it can instead be projected onto a 2-D plane in which each processor treats cells at a different distance normal to the 2-D plane chosen. For example, if the 3-D wavefront is projected onto the X-Y plane, it can be used to decompose the mesh in space as shown in Fig. 2.8. For the depicted mesh, processor 2



(a) Wavefront 1.

(b) Wavefront 2.

(c) Wavefront 3.

Figure 2.7: KBA sweep plane progression.

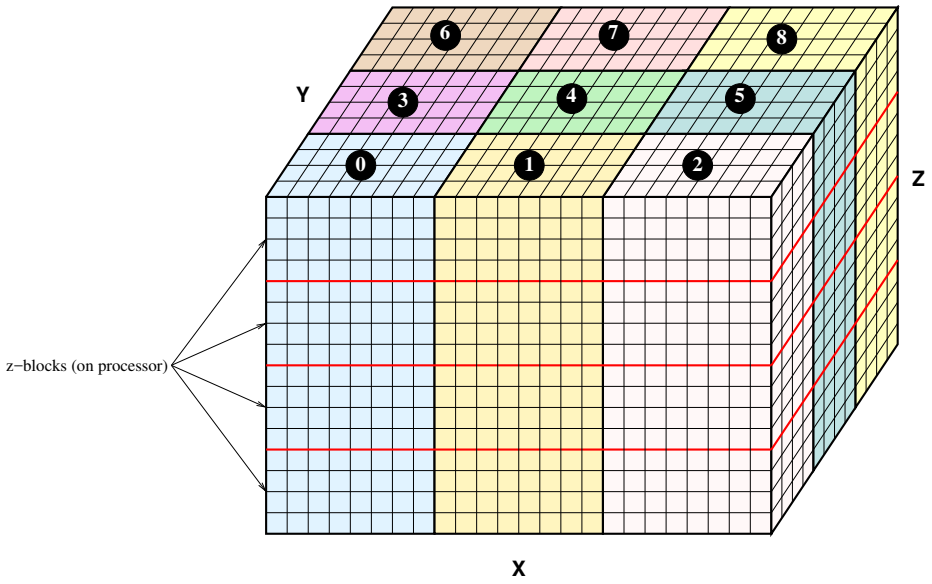


Figure 2.8: Spatial decomposition for KBA, using an X-Y projection. Top numbers correspond to processor index. Red lines indicate within-processor block boundaries, which create more wavefront steps and aid in parallelism.

would be the only active processor for the first wavefront step treating the hashed block of cells in Fig. 2.7a. By the third wavefront step (Fig. 2.7c), processors 0, 4, and 8 would be treating blocks at the top of the mesh, processors 1 and 5, blocks one lower, and processor 2 would be sweeping cells two blocks down. Using this scheme, communication need only take place for the x and y faces of the blocks, while the z faces of the blocks are kept within the processor.

Mapping the blocks in the wavefront to physical processors in this way minimizes the communication necessary. However, there are still periods when not all processors are in use. To improve utilization of the parallel processors, KBA uses an angular pipelining approach. Since the order in which cells must be swept (and hence the nature of the wavefront decomposition) depends only upon the octant in which an angle lies, all angles in given octant are swept together in series. This allows a processor to continue sweeping the next angle in the octant as soon as it has reached the end of the mesh for the current octant angle. This has the effect of stacking the multiple instances of the spatial domain, resulting in an effective mesh that is "taller" than it would otherwise be, increasing the amount of

time that all processors are active. This is depicted in Fig. 2.9, in which the blue borders denote the boundaries between the angles. It should be noted that with a fixed number of angles, weak scaling in space will not be ideal, since an appreciable number of processors will be unused while the sweep pipeline is being filled. This effect is apparent in Fig. 2.6a.

2.4.2 Method of Characteristics

The Method of Characteristics treats the multi-group, discrete ordinates form of the transport equation by considering characteristic curves along which the solution variable is allowed to change [13, 19]. This results in a series of ordinary differential equations which may be solved along each characteristic. Starting with Eq. (2.20), the coordinate transformation,

$$\mathbf{r} = \mathbf{r}_0 + s\hat{\mathbf{\Omega}} \rightarrow \begin{aligned} x(s) &= x_0 + \eta^n s \\ y(s) &= y_0 + \varepsilon^n s \\ z(s) &= z_0 + \mu^n s, \end{aligned}$$

is performed to obtain

$$\frac{d}{ds}\psi(s) + \Sigma_t(s)\psi(s) = Q(s). \tag{2.35}$$

Considering a single mesh region with constant total cross section and source, Eq. (2.35) has the analytic solution

$$\psi^n(s) = \psi^n_{in} e^{-\Sigma_t s} + \frac{Q}{\Sigma_t}\left(1 - e^{-\Sigma_t s}\right), \tag{2.36}$$

where ψ_{in} is the angular flux incident upon the region, or at $\mathbf{r}_0$. The approximate solution to the transport equation for each angle is obtained by solving the above equation along numerous characteristic curves, or rays, which pass through the reactor core.

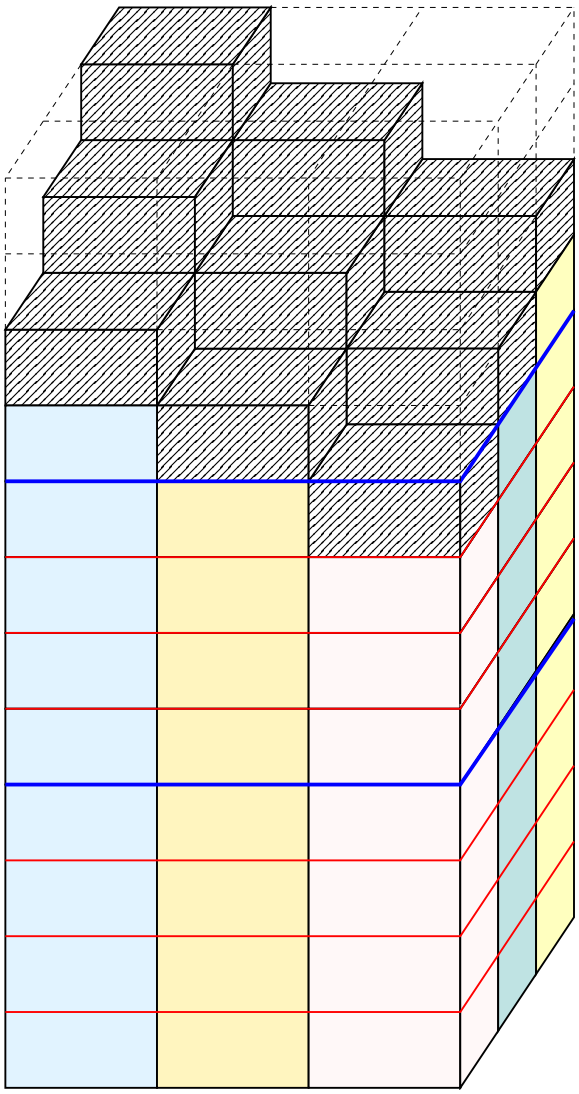


Figure 2.9: Angular pipelining approach used in KBA. Blue lines are “imaginary” boundaries between spatial domains for different angles in the same octant. This has the effect of vertically “stacking” multiple instances of the spatial domain.

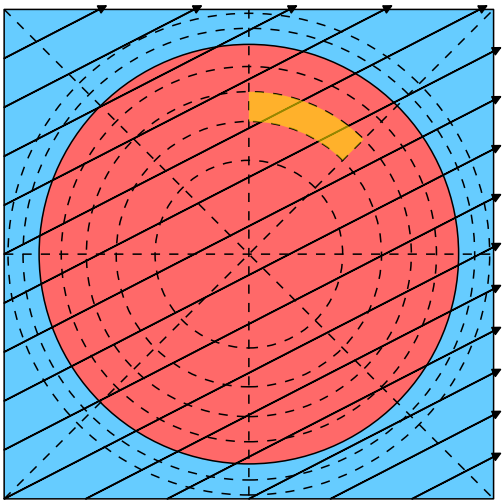


Figure 2.10: Typical MoC pin mesh with rays. Dotted lines denote mesh boundaries, while the solid circle represents a material discontinuity (also a mesh boundary). An FSR is a region of the mesh enclosed mesh boundaries. For example, the yellow highlighted region is an FSR, which is shown in more detail in Fig. 2.11.

Flat Source Region mesh and rays The spatial domain is typically divided into a mesh of Flat Source Regions (FSRs), as depicted in Fig. 2.10. In most cases, rays are spaced evenly with some nominal ray spacing, δ. For a given ray segment of length t traversing an FSR, that segment represents the approximate solution within a sub-volume of the FSR as shown in Fig. 2.11. Even though the mesh and rays are 2-D, they actually represent and interact with a prismatic, 3-D structure; therefore, the ray segment lengths must be scaled relative to the polar angle,

$$\ell = \frac{t}{\sqrt{1-\mu^2}}. \tag{2.37}$$

As depicted in Fig. 2.11, the volume of all ray segments in an FSR do not necessarily represent the exact volume of that region. Therefore in practice, an effective ray segment length is used instead, which preserves the total amount of material in each region by stretching or compressing the length of the segments. This is typically done in one of two

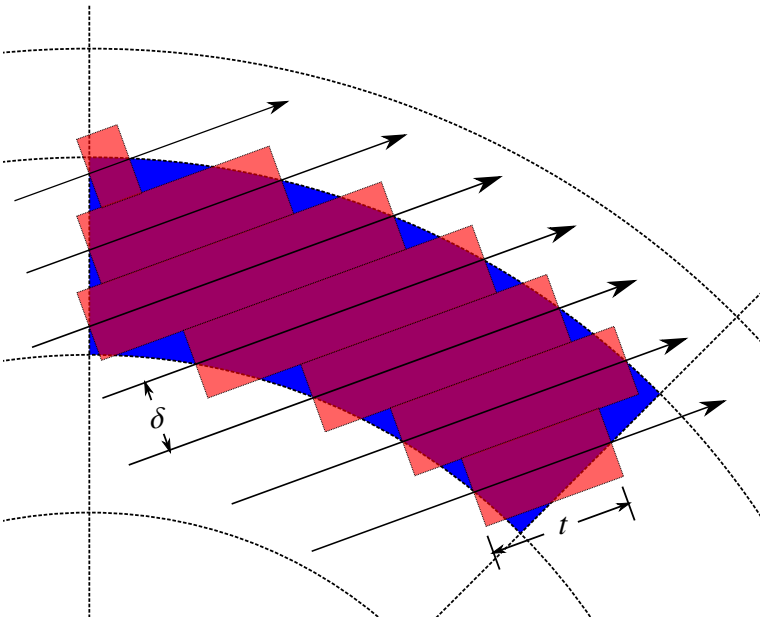


Figure 2.11: Typical FSR with rays and associated ray segments.

ways. The first preserves the total volume of all FSRs for each angle individually,

$$\ell^m_{\text{eff}} = \frac{\sum_{m \in \text{FSR}} \delta \ell^m}{V_{\text{FSR}}}. \tag{2.38}$$

The second preserves the angular integral of the FSR volumes,

$$\ell^m_{\text{eff}} = \frac{\sum_{m \in \text{FSR}} \sum_{n=1}^{N} w^n \delta \ell^m}{4\pi V_{\text{FSR}}}. \tag{2.39}$$

While it is essential that the total amount of material be preserved for each FSR, the modification of the segment lengths does introduce error in the representation of the geometry of the mesh regions, and thus it is desired to minimize the magnitude of the correction. For this reason the second approach is usually favored, since considering all angles and all segments together tends to yield a smaller overall modification.

Equation (2.36) can be used to develop expressions for the segment-average and outgo-

ing angular flux for a given ray, m

$$\psi_{out}^{m} = \psi_{in}^{m} e^{-\Sigma_{\mathrm{t}} \ell^{m}} + \frac{Q}{\Sigma_{\mathrm{t}}} \left(1 - e^{-\Sigma_{\mathrm{t}} \ell^{m}}\right) \tag{2.40}$$

$$\bar{\psi}^{m} = \frac{Q}{\Sigma_{\mathrm{t}}} + \frac{1}{\Sigma_{\mathrm{t}} \ell^{m}} \left(\psi_{in}^{m} - \frac{Q}{\Sigma_{\mathrm{t}}}\right) \left(1 - e^{-\Sigma_{\mathrm{t}} \ell^{m}}\right). \tag{2.41}$$

Ray modularization Some MoC codes such as MPACT and MOCC use a concept called *modular ray tracing* to reduce the memory burden of storing the ray data, as well as the time needed to trace them through the mesh. Modular ray tracing takes advantage of the highly periodic nature of the geometry of a nuclear reactor core, which permits the re-use of ray tracing data within appropriate subdomains. Depending upon the nature of the problem, different repeating geometries might be used, including pin cells, quarter assemblies or full assemblies. These repeated structures are referred to as "ray tracing modules."

In order to do this properly, several constraints must be placed on the nature of the rays and the angles along which they travel to ensure that a ray exiting one module lines up with another ray entering the neighboring module. A typical approach to enforcing this constraint involves performing the following steps for each angle in the quadrature. First an integral number of rays crossing each boundary of the border of the module is determined given the starting azimuthal angle, α,

$$\begin{aligned} \mathrm{N}_x &= \left\lceil \frac{H_x}{\delta \sin \alpha} \right\rceil \\ \mathrm{N}_y &= \left\lceil \frac{H_y}{\delta \cos \alpha} \right\rceil, \end{aligned} \tag{2.42}$$

where N_x and N_y are the number of rays crossing the $x-$ and $y-$normal faces of the module, respectively. H_x and H_y are the dimensions of the module. The use of the ceiling function ensures that the final ray spacing is smaller than the desired, nominal ray spacing. A new

azimuthal angle is then determined, given the configuration of the ray as

$$\alpha' = \arctan\left(\frac{H_y N_x}{H_x N_y}\right), \tag{2.43}$$

and a new, angle-specific ray spacing is determined by

$$\delta' = \cos(\alpha')\frac{H_y}{N_y}. \tag{2.44}$$

This was the approach used in the CACTUS code [13], an early implementation of the MoC. Depending upon the dimensions of the ray tracing module and the desired ray spacing, modularization can have a noticeable impact upon the final angular quadrature, and care must be taken to properly adjust quadrature weights accordingly.

It should be noted that even when not using modular ray tracing to save memory, it is still useful to treat the entire geometry as a single ray tracing module to aid in handling reflective boundary conditions; if the rays of reflected angles line up at the mesh boundary, reflective or albedo boundary conditions may be enforced simply by using the outgoing flux from the exiting ray as the incoming flux for the entering ray. This is the approach used in the MOCC code.

MoC sweep algorithm An MoC sweep loops over all angles, initializing incoming angular flux using an appropriate boundary condition, and propagates the angular flux along each ray, one segment at a time. Following traversal of the mesh, the outgoing flux is stored to possibly be used later as incoming flux for reflective, periodic, or albedo boundary conditions.

While it is possible to use Eqs. (2.40) and (2.41) in the ray sweep, alternative forms tend to be more efficient for computation. Instead of computing the segment-average angular flux using Eq. (2.41), the outgoing flux can be expressed as the incoming flux, plus some

incremental flux, $\Delta\psi^m$, and the segment-average flux in terms of this incremental flux,

$$\psi^m_{out} = \psi^m_{in} + \Delta\psi^m. \tag{2.45}$$

Using the above and Eq. (2.40), the incremental flux can be calculated as

$$\Delta\psi_m = \left(\frac{Q}{\Sigma_t} - \psi^m_{in}\right)\left(1 - e^{-\Sigma_t \ell^m}\right). \tag{2.46}$$

Referring back to Eq. (2.41), and inserting the expression for $\Delta\psi^m$ gives

$$\bar{\psi}^m = \frac{Q}{\Sigma_t} - \frac{1}{\Sigma_t \ell^m}\Delta\psi^m. \tag{2.47}$$

After common subexpression elimination, computing Eqs. (2.41) and (2.40) would require one exponential function evaluation, five floating-point multiplications, four floating-point additions or subtractions, and two floating-point divisions. In contrast, Eqs. (2.46) and (2.47) require three fewer floating-point multiplications. Since these are the most frequently-evaluated expressions in an MoC sweeper, this reduction in operations is quite valuable.

The scalar flux in an FSR is approximated as the sum over all segments and angles,

$$\phi_{\text{FSR}} = \frac{\sum\limits_{n \in N}\left(w_n \sum\limits_{m \in \text{FSR}} \delta\ell^{m,n}_{\text{eff}} \bar{\psi}^{m,n}\right)}{V_{\text{FSR}}}. \tag{2.48}$$

Further savings may be made by first applying the definition of $\bar{\psi}^m$ to the above summation,

$$\phi_{\text{FSR}} = \frac{\sum\limits_{n \in N}\left(w_n \delta^n \sum\limits_{m \in \text{FSR}} \ell^{m,n}_{\text{eff}} \frac{Q}{\Sigma_t} - \frac{\Delta\psi^m}{\Sigma_t}\right)}{V_{\text{FSR}}}. \tag{2.49}$$

Recognizing that $\frac{Q}{\Sigma_t}$ is constant over angle and ray segment for any FSR and the quadrature-weighted sum over all ℓ_{eff} for all angles is $4\pi V_{\text{FSR}}$ allows the term to be removed from the

summation. The $\frac{1}{\Sigma_t}$ factor may similarly be removed. After removing as many operations as possible from the scalar flux sum, what remains is

$$\phi_{\text{FSR}} = 4\pi V_{\text{FSR}} \frac{Q}{\Sigma_t} - \frac{\sum_{n \in N} \left(w_n \delta^n \sum_{m \in \text{FSR}} \Delta\psi^m \right)}{V_{\text{FSR}}}. \tag{2.50}$$

The final MoC sweep algorithm is presented in Algorithm 1 as it would typically be embedded in an eigenvalue solver.

```
Initialize;
Make initial guesses for fission source and k_eff;
while Fission source, eigenvalue not converged do
    for group ∈ G do
        Calculate energy group-dependent source from in-scattering and fission;
        Pre-calculate Q/Σ_t for the current group;
        for inner=0, N_inner do
            Calculate total source, including self-scattering;
            Reset running sum of intermediate scalar flux value: φ = 0;
            for Angles do
                for Rays do
                    Initialize ψ_in from boundary value;
                    for Ray segments do
                        Look up FSR index, i, corresponding to ray segment;
                        Calculate Δψ using Eq. (2.46);
                        φ_i = φ_i + w_n δ_n Δψ;
                        Update ψ_in using Eq. (2.45);
                    end
                    Store ψ_out boundary value;
                end
            end
            Adjust flux summation to recover actual scalar flux: φ = 4π Q/Σ_t − φ/(Σ_t V_FSR);
        end
    end
    Update fission source and k_eff;
    Check for convergence;
end
```

Algorithm 1: Method of Characteristics sweep algorithm

2.5 2-D/1-D Method

The 2-D/1-D method as considered in this work was first used in the CRX [25] and DeCART [7, 16] reactor analysis codes, and was used to extend the solutions from 2-D MoC sweepers to the full 3-D reactor geometry. It is currently the mainstay approach to solving 3-D problems in MPACT.

2-D/1-D is a planar synthesis method, which splits the 3-D reactor geometry into a stack of 2-D axial slabs. A detailed 2-D method is used to treat each of these slabs, which are coupled using a 1-D axial solver, as shown in Fig. 2.12. Traditionally, MoC is used to treat the 2-D planes, while a lower-fidelity, pin-homogenized method is used in the axial solvers. While the 2-D MoC treatment is fairly consistent between most 2-D/1-D implementations, a variety of methods have been used for treating the axial dimension. Nodal diffusion methods, such as Nodal Expansion Method (NEM) and Semi-Analytic Nodal Method (SANM) have been used for some time, while newer methods have begun employing nodal S_N or other transport-based methods [14, 30]. Communication between the radial and axial sweepers is carried out using the concept of *Transverse Leakage (TL)*, which applies an auxiliary source to each sweeper, accounting for neutron streaming in the dimension(s) orthogonal to the particular sweeper. The radial sweepers include an axial TL term, while the axial sweepers include a radial TL term. Each TL is obtained by integrating the transport equation over the dimensions that are not treated by a given sweeper.

2.5.1 Axially-Integrated Transport Equation

The radial (MoC) sweepers in the 2-D/1-D scheme solve the axially-integrated form of the transport equation. This equation is obtained by starting with the multi-group, discrete-ordinates form of the transport equation in Cartesian coordinates,

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y} + \mu^n \frac{\partial}{\partial z}\right) \psi_g^n(x,y,z) + \Sigma_{t,g}(x,y,z) = Q(x,y,z), \tag{2.51}$$

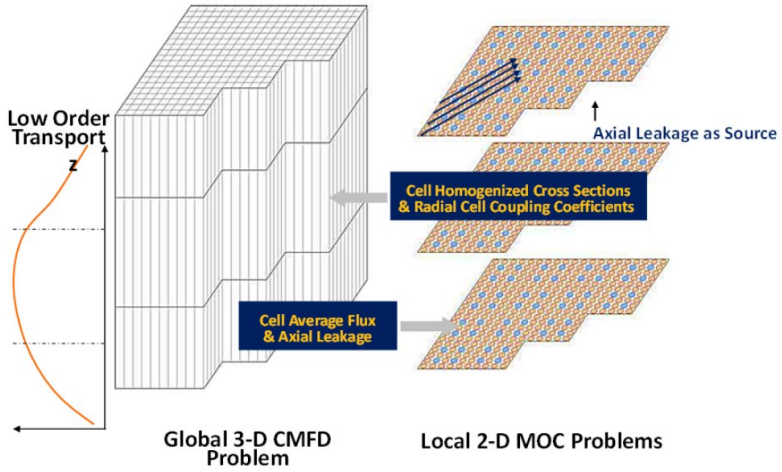


Figure 2.12: Illustration of the 2-D/1-D method.

where $Q(x,y,z)$ is the true neutron source, including fission and scattering sources,

$$Q(x,y,z) = \frac{\chi_g}{k} \sum_{g' \in G} \nu\Sigma_{\mathrm{f},g'}(x,y,z)\phi_{g'}(x,y,z) + \sum_{g' \in G} \Sigma_{\mathrm{s,g'g}}\phi_{g'}(x,y,z) \tag{2.52}$$

Operating on the above equation with

$$\frac{1}{h_z}\int_{z_B}^{z_T}(\cdot)\,dz \tag{2.53}$$

produces the axially-averaged transport equation,

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y}\right)\psi_g^n(x,y) + \Sigma_{\mathrm{t},g}\psi_g^n(x,y) + L_g^n(x,y) = Q_g^n(x,y), \tag{2.54a}$$

$$L_g^n(x,y) = \frac{\mu^n}{h_z}\left(\psi_{g,T}^n(x,y) - \psi_{g,B}^n(x,y)\right), \tag{2.54b}$$

where h_z is the height of the planar axial region and $\psi_{g,T}^n(x,y) = \psi_g^n(x,y,z_T)$ and $\psi_{g,B}^n(x,y) = \psi_g^n(x,y,z_B)$. $\psi_g^n(x,y)$ represents the axially-averaged, or radially-dependent angular flux for direction of travel n and group g, $\Sigma_{\mathrm{t}}(x,y)$ is the axially-averaged cross section, and so on.

The $L_g^n(x,y)$ term is called "axial transverse leakage," and accounts for the component of the streaming term in the axial direction. In 2-D/1-D schemes, it is typically moved to the right hand side and treated as a transverse leakage source, where the flux at the top and bottom of the axial slab are calculated using a 1-D axial solution. These schemes typically operate on a pin-homogenized mesh, and the plane-top and -bottom fluxes are approximated as piecewise constants,

$$\psi_{g,B}^n(x,y) \approx \psi_{g,B,i,j}^n \text{ and} \tag{2.55}$$

$$\psi_{g,T}^n(x,y) \approx \psi_{g,T,i,j}^n, \tag{2.56}$$

where i and j are x and y pin indices, respectively. As a result, a single TL source is usually applied to all FSRs within a given pin. This is often referred to as the "flat" TL approximation. Research has shown that this can be a large source of error in 2-D/1-D methods [31], and it is equally relevant for 2-D/3-D methods as well.

2.5.2 Radially-Integrated Equations

2-D/1-D methods choose to use either the transport or diffusion equations to treat the axial direction. Both are obtained using a similar approach as used above for the axially-integrated equations, except they are integrated over a radial region as

$$\frac{1}{h_x h_y} \int_{x_L}^{x_R} \int_{y_L}^{y_R} (\cdot) dy dx. \tag{2.57}$$

Applying this treatment to Eq. (2.51) yields

$$\mu^n \frac{d}{dz} \psi_g^n(z) + \Sigma t, g(z) + L_g^n(z) = Q(z), \tag{2.58a}$$

$$L_g^n(z) = \frac{1}{h_x h_y}\Big(\eta^n \int_{y_L}^{y_R} \psi_g^n(x_R, y, z) - \psi_g^n(x_L, y, z)dy + \tag{2.58b}$$

$$\varepsilon^n \int_{x_L}^{x_R} \psi_g^n(x, y_R, z) - \psi_g^n(x, y_L, z)dx\Big) \tag{2.58c}$$

It is also common in 2-D/1-D methods to use axial diffusion. The same treatment can be applied to the 3-D diffusion equation to get

$$-D_g \frac{\partial^2}{\partial z^2}\phi_g(z) + \Sigma_{r,g}\phi(z) + L_g(z) = \frac{\chi_g}{k}\sum_{g' \in G} \nu\Sigma_{\mathrm{f},g'}\phi_{g'}(z) + \sum_{g' \in G, g' \neq g} \Sigma_{\mathrm{s,g'g}}\phi_{g'} \tag{2.59a}$$

$$L_g(z) = \frac{1}{h_x}\left(J_x^R(z) - J_x^L(z)\right) + \frac{1}{h_y}\left(J_y^R(z) - J_y^L(z)\right). \tag{2.59b}$$

In the above, J represents the net current on the face and in the direction indicated, and are obtained from the 2-D radial sweeper. Use of the above equation with NEM and SANM nodal methods have historically been the most popular axial treatments in 2-D/1-D. However, 1-D transport-based methods, such as nodal Simplified Pn (SPn) are becoming more commonly used.

2.5.3 Iteration Scheme

While iteration schemes vary between implementations, the iteration process employed by MPACT is depicted in Fig. 2.13. Each outer iteration begins with a global CMFD solve, which updates the fission source and eigenvalue estimate. An axial solver is then used to determine the flux behavior along the axial direction for each pin. The resultant axial solution is used to calculate axial TLs for use in the radial sweepers. Each plane is then swept with the radial MoC sweepers, employing the TLs obtained in the previous step. At the end of the radial sweeps, new currents are calculated on the pin boundaries for use in the next CMFD iteration.

This iteration scheme has proved effective for many reactor analysis applications. However in some cases, especially with small axial slabs, the iteration scheme can become

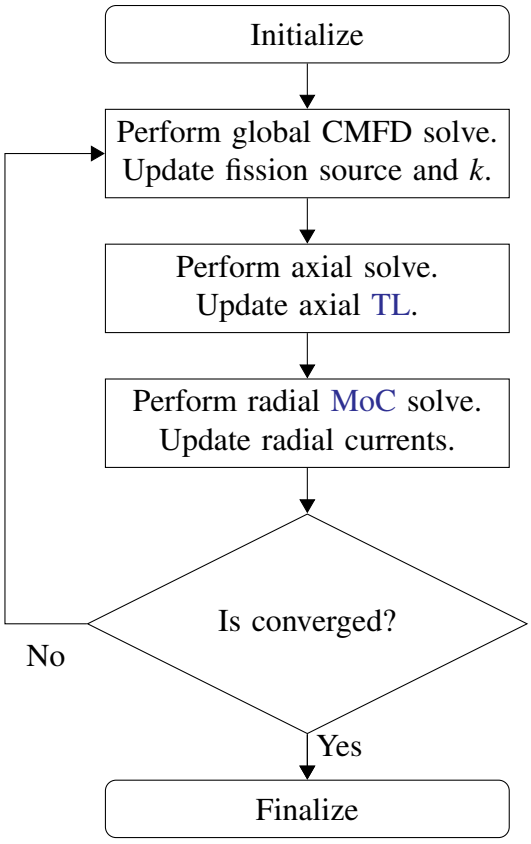


Figure 2.13: 2-D/1-D iteration scheme.

unstable, resulting in the divergence of the solution. This effect has been studied in recent work [17, 18]. The result of this analysis was the development of an optimal under-relaxation factor, which guarantees stability while minimizing the negative impact on the convergence rate of the solution.

2.6 Coarse Mesh Finite Difference

The CMFD method [28] is a finite differencing approach for solving the diffusion equation, described in Section 2.2.1. It is formulated in such a way that the Finite Difference (FD) diffusion calculation will produce an equivalent solution to a coupled transport calculation at convergence. This equivalence permits its use as an accelerator for the coupled transport solver.

In practice, CMFD is usually applied on a coarse mesh that is aligned with the pin mesh of a reactor. Each pin cell is homogenized to form a single region of an orthogonal grid. Within each region, the cross sections are assumed constant. In standard FD the interface current from region, i to its neighbor, j is expressed

$$J_{i\to j} = \tilde{D}_{ij}\left(\phi_i - \phi_j\right), \tag{2.60}$$

where $\tilde{D}_{ij}$ is the *surface diffusivity* of the interface between regions i and j, and is defined as

$$\tilde{D}_{ij} = \frac{2\frac{D_i}{h_i}\frac{D_j}{h_j}}{\frac{D_i}{h_i} + \frac{D_j}{h_j}}. \tag{2.61}$$

The CMFD method applies a modification to Eq. (2.60), which includes non-linear correction term, $\hat{D}_{ij}$:

$$J_{i\to j} = \tilde{D}_{ij}\left(\phi_i - \phi_j\right) + \hat{D}_{ij}\left(\phi_i + \phi_j\right). \tag{2.62}$$

Here, the value of $\hat{D}_{ij}$ is computed using the known values of the fine-mesh flux and and interface currents from the transport solver that is being accelerated, homogenized to the coarse mesh:

$$\hat{D}_{ij} = \frac{J^{FM}_{i\to j} - \tilde{D}_{ij}\left(\phi^{FM}_i - \phi^{FM}_j\right)}{\phi^{FM}_i + \phi^{FM}_j}. \tag{2.63}$$

In the computation of $\hat{D}_{ij}$, ϕ^{FM} and J^{FM} represent these homogenized values from the fine mesh transport solver. In the case of accelerating MoC calculations, values of the interface currents are computed with a specialized sweeping method that tallies the net current contributions of rays as they cross the boundaries of the coarse mesh. This special sweep procedure is more computationally expensive than the regular MoC sweeps, and therefore is only used on the last iteration prior to performing a CMFD acceleration step. Other necessary group constants are homogenized using the fine-mesh, transport-based scalar flux.

Using our expressions for cell interface currents, Eq. (2.62), the following balance equa-

tion for a 3-D, cuboid coarse mesh cell can be obtained:

$$
\begin{aligned}
&\left(\hat{D}^E_{i,g} - \tilde{D}^E_{i,g}\right) A_x \phi^E_{i,g} + \left(\hat{D}^W_{i,g} - \tilde{D}^W_{i,g}\right) A_x \phi^W_{i,g} + \\
&\left(\hat{D}^N_{i,g} - \tilde{D}^N_{i,g}\right) A_y \phi^N_{i,g} + \left(\hat{D}^S_{i,g} - \tilde{D}^S_{i,g}\right) A_y \phi^S_{i,g} + \\
&\left(\hat{D}^T_{i,g} - \tilde{D}^T_{i,g}\right) A_z \phi^T_{i,g} + \left(\hat{D}^B_{i,g} - \tilde{D}^B_{i,g}\right) A_z \phi^B_{i,g} + \\
&\left[\left(\tilde{D}^E_{i,g} + \hat{D}^E_{i,g} + \tilde{D}^W_{i,g} - \hat{D}^W_{i,g}\right) A_x + \left(\tilde{D}^N_{i,g} + \hat{D}^N_{i,g} + \tilde{D}^S_{i,g} - \hat{D}^S_{i,g}\right) A_y + \right. \\
&\left. \left(\tilde{D}^T_{i,g} + \hat{D}^T_{i,g} + \tilde{D}^B_{i,g} - \hat{D}^B_{i,g}\right) A_z + \Sigma_{t,i,g} V_i \right] \phi_{i,g} \\
&= \frac{\chi_g}{k} V_i \sum_{g' \in G} \nu \Sigma_{f,i,g'} \phi_{i,g'} + V_i \sum_{g' \in G} \Sigma_{s,i,g'g} \phi_{i,g'}, \quad i \in CM, \quad g \in G.
\end{aligned}
\tag{2.64}
$$

In the above equations, the superscripts N, S, E, W, T and B denote the neighboring mesh regions or interfaces in the north, south, east, west, top and bottom, respectively. A_x, A_y and A_z are the x-, y- and z-normal interface areas and V_i is the volume of the Coarse Mesh (CM) region i, and CM is the set of regions in the coarse mesh.

Equations (2.64) form a large system of equations, which are solved in the code using standard algebra techniques. MPACT uses the PETSc library [3, 4], which is a powerful, parallel linear algebra library. MOCC uses the Eigen linear algebra library, which is unfortunately serial. The solution of the CMFD eigenvalue problem is then projected back to the transport mesh. If the solution is not yet converged, another series of transport sweeps are performed, producing new cross sections and $\hat{D}$ terms. Due to the equivalent formulation of CMFD, at convergence the solution to the CMFD system will agree with that of the transport sweeper. Since the CMFD system is much less expensive to solve than the transport equation, it makes a very effective accelerator for the convergence of the transport solution without affecting accuracy.

CHAPTER 3

Corrected Diamond Difference

The goal of the 2-D/3-D method is to take advantage of the speed and parallel efficiency of orthogonal mesh S_N and the in-plane accuracy of MoC to obtain full-core, 3-D, pin-resolved solutions to the transport equation. A key requirement of the method is for the orthogonal-mesh S_N to maintain accuracy at practical mesh refinements, which proves challenging for standalone S_N, since orthogonal meshes are notoriously poor at representing the geometries found in virtually all reactor designs. In this chapter, the poor performance of standalone orthogonal mesh S_N in reactor applications is demonstrated experimentally. The CDD scheme, reminiscent of Weighted Diamond Difference (WDD), is then introduced, which incorporates correction data from an underlying MoC sweeper. Finally, it is demonstrated that in 2-D, CDD is capable of maintaining the same accuracy as the embedded MoC solution, and some observations of the correction factors are presented.

3.1 Standard Sn with Flux-Weighted Cross Sections

The simplest method considered for improving the accuracy of orthogonal mesh S_N is to provide flux-weighted cross sections to the coarse mesh using a FM flux distribution. This was the first method investigated for improving the accuracy of Denovo simulations, and was studied in detail in 2012 [35] at Oak Ridge National Laboratory. By default, the Denovo code operates by overlaying an orthogonal mesh on the physical pin geometry as depicted in Fig. 3.1, and calculating volume-weighted cross sections for each region of

the orthogonal grid. The number of sub-regions per pin is user-defined, with increasing numbers of regions tending to result in less modeling error.

Early in the application of Denovo to reactor simulations, it was found that a prohibitively large number of regions was necessary to achieve satisfactory accuracy in many cases. In particular, cases with small regions of strong neutron absorbers, such as those found in IFBA pins require upwards of 50-by-50 mesh elements per pin to yield acceptable global solutions. As stated above, this would result in a prohibitively large mesh in full-core situations. The error incurred by using an orthogonal mesh comes from several major sources:

- discretization of the solution itself,
- poor representation of the actual pin geometry and its curved surfaces, and
- the mixture of cross sections, since multiple real-world materials may inhabit the same orthogonal mesh regions.

Much of the error in the cross sections is due to the volume weighting that is used, which assumes a flat flux profile within each coarse region. This grants equal importance to all subregions, which in most cases is inappropriate. In realistic reactor scenarios, the flux shape tends to exhibit strong radial dependence within a pin cell due to heterogeneities and self-shielding effects. Therefore, flux-weighting was used to achieve lower error due to cross sections for the same mesh refinement.

Flux weighting of the group constants is carried out using the following relationships,

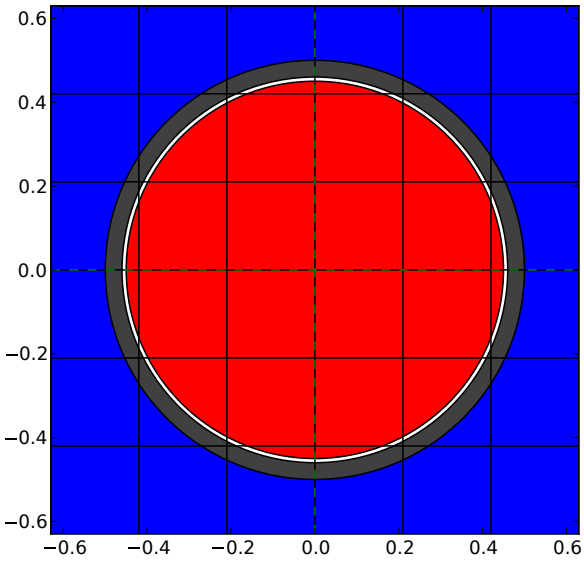


Figure 3.1: Orthogonal mesh overlay on a typical fuel pin.

which attempt to preserve total reaction rates within a homogenized region:

$$\Sigma_{t,V,g} = \frac{\sum\limits_{i \in V} \Sigma_{t,i,g} \phi_{i,g} V_i}{\sum\limits_{i \in V} \phi_{i,g} V_i}$$

$$\nu\Sigma_{f,V,g} = \frac{\sum\limits_{i \in V} \nu\Sigma_{f,i,g} \phi_{i,g} V_i}{\sum\limits_{i \in V} \phi_{i,g} V_i}$$

$$\Sigma_{s,V,g'g} = \frac{\sum\limits_{i \in V} \Sigma_{s,g'g} \phi_{g',i} V_i}{\sum\limits_{i \in V} \phi_{g',i} V_i}$$

$$\chi_{g,V} = \frac{\sum\limits_{i \in V} \chi_{i,g} V_i \sum\limits_{g' \in G} \phi_{i,g'} \nu\Sigma_{f,i,g'}}{\sum\limits_{i \in V} V_i \sum\limits_{g' \in G} \phi_{i,g'} \nu\Sigma_{f,i,g'}}$$

Following the generation of the flux-weighted cross sections, any S_N scheme may be employed to solve the resultant global problem, such as DD, Step Characteristics (SC) or Linear Discontinuous (LD).

In the 2012 study, we considered a 3-by-3 pin array with a central control rod guide tube and several pins containing IFBA coatings, depicted in Fig. 3.2. Several mesh refinements were used, ranging from 4-by-4 regions per pin to 24-by-24 regions per pin. In each case,

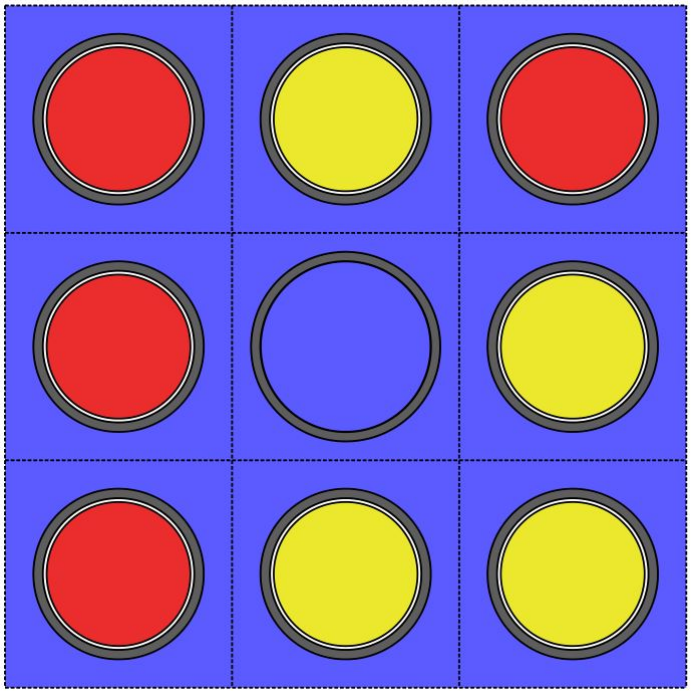


Figure 3.2: 3-by-3 pin array. Yellow pins contain IFBA coating, red pins are bare fuel.

an MoC calculation was performed on a sufficiently resolved conformal mesh (see Section 2.4.2), and the resulting scalar flux distribution was used to generate flux-weighted cross sections for each orthogonal mesh region. Using these cross sections, an S_N calculation was performed with the LD differencing scheme, and the system eigenvalue was compared against a very fine-mesh reference solution. The results are shown in Fig. 3.3.

Several important observations were made:

- As expected, the eigenvalue error with flux-weighted cross sections was lower for all cases than the volume-weighted cross sections.
- Even at 24-by-24 regions per pin the eigenvalue error was in the hundreds of per cent mille (pcm), which is far from acceptable.
- The eigenvalue error does not improve monotonically as the mesh is refined. Instead, some coarser S_N meshes produce better answers than finer ones. This is likely due to the interaction of the IFBA coating with the orthogonal mesh; depending on where

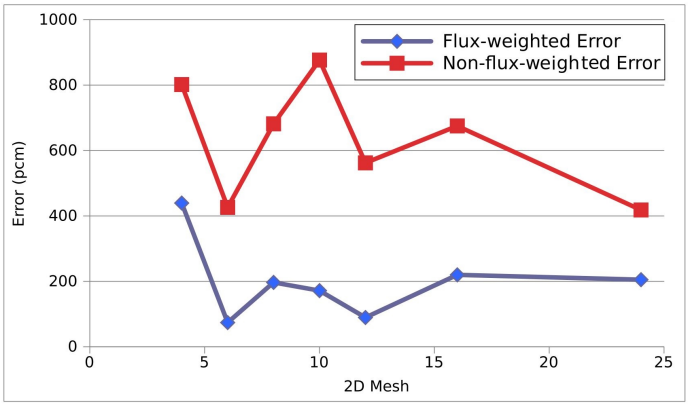


Figure 3.3: Mesh convergence behavior for 3x3 pin array with IFBA pins

the S_N mesh boundaries lie in relation to the thin IFBA layer, it is possible for some regions to contain only a small portion of IFBA, which contaminates the rest of the region's cross sections, resulting in poor representation of the pin geometry. The severity of this IFBA "corner clipping" effect depends somewhat chaotically upon mesh parameters, and is difficult to characterize. Operating on a pin-homogenized mesh (1-by-1 per pin), would alleviate this particular issue, but is so inaccurate due to the coarseness of the mesh that it is also not practical.

The results of the above analysis effectively rule out the use of traditional S_N methods for use in reactor calculations with reasonably coarse grids, even if flux-weighted cross sections are used. In the following sections, a more sophisticated technique is developed for informing a coarse mesh S_N solver.

3.2 Corrected Diamond Difference

In Section 3.1 the usefulness of fine-mesh MoC flux weighted cross sections was explored to improve the accuracy of coarse mesh S_N. While this demonstrated roughly increasing accuracy with refinement of the S_N mesh, operating on the order of a pin-level mesh was far too inaccurate. There is however much more information from the fine mesh sweep at our disposal, which can be used to formulate a more accurate S_N approach. In the

CDD method, a pair of correction factors are developed, which are calculated using flux data from a FM MoC sweeper, and correct the streaming and collision terms of the S_N equation. Both corrections apply sufficient modifications to the original DD equations to force them to assume the same transport behavior as the underlying MoC sweeper under certain circumstances.

3.2.1 Streaming Correction Factor

The first component of CDD involves adding a pair of correction terms, α_x and α_y, which are calculated using fine-mesh flux quantities from an MoC sweep.

Determining the streaming correction factor begins with the discrete ordinate form of the differential transport equation,

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y} + \Sigma_t(\vec{r})\right)\psi^n(\vec{r}) = Q(\vec{r}). \tag{3.1}$$

Integrating Eq. (3.1) over a pin cell and representing the angular flux derivatives as spatial differences produces

$$\frac{\eta^n}{\Delta x_i}\left(\psi_{i+1/2,j} - \psi_{i-1/2,j}\right) + \frac{\varepsilon^n}{\Delta y_j}\left(\psi_{i,j+1/2} - \psi_{i,j-1/2}\right) + \Sigma_t \overline{\psi}_{ij} = Q_{ij}. \tag{3.2}$$

Here the angle index has been omitted for clarity.

Starting with the DD scheme and applying the streaming correction factors, α_x and α_y gives

$$\begin{aligned} \overline{\psi}_{ij} &= \alpha_{x,ij}\left(\psi_{i+1/2,j} + \psi_{i-1/2,j}\right) \text{ and} \\ \overline{\psi}_{ij} &= \alpha_{y,ij}\left(\psi_{i,j+1/2} + \psi_{i,j-1/2}\right). \end{aligned} \tag{3.3}$$

The correction terms are calculated for each dimension of the mesh element for which fine mesh data is available. For example, in the case of using a 2-D MoC sweeper, these are

calculated for the x and y directions, as shown in Eqs. (3.3) and (3.4).

Given values from the fine mesh solution for $\bar{\psi}^F_{ij}$, $\psi^F_{i\pm1/2,j}$ and $\psi^F_{i,j\pm1/2}$, the terms can be calculated with the expressions

$$\alpha_{x,ij} = \frac{\bar{\psi}^F_{ij}}{\psi^F_{i+1/2,j} + \psi^F_{i-1/2,j}} \text{ and}$$
$$\alpha_{y,ij} = \frac{\bar{\psi}^F_{ij}}{\psi^F_{i,j+1/2} + \psi^F_{i,j-1/2}}. \tag{3.4}$$

3.2.2 Collision Correction Factor

To accurately represent the reaction rates within a coarse mesh cell, the fine mesh, angle-dependent collision rate, homogenized onto a coarse mesh region is considered,

$$R^n = \bar{\psi}^n \bar{\Sigma}_{\mathrm{t}}^{\,n}, \tag{3.5}$$

where R^n is the reaction rate of neutrons traveling in direction $\vec{\Omega}_n$ through a coarse mesh region for a single energy group. In this case the homogenized cross section Σ^n_t are angular flux-weighted, and defined as

$$\bar{\Sigma}_{\mathrm{t}}^{\,n} = \frac{\sum_{i\in \mathrm{FM}} \Sigma_{\mathrm{t},i} \bar{\psi}^n_i V_i}{\sum_{i\in \mathrm{FM}} \bar{\psi}^n_i V_i}, \tag{3.6}$$

where FM is the set of FM regions comprising a CM region. The corresponding cell-averaged angular flux is defined as

$$\bar{\psi}^n = \frac{\sum_{i\in \mathrm{FM}} \bar{\psi}^n_i V_i}{\sum_{i\in \mathrm{FM}} V_i}. \tag{3.7}$$

In a traditional S_N sweep, our collision term would employ a scalar flux-weighted total cross section, yielding a reaction rate of

$$\tilde{R}^n = \bar{\psi}^n \Sigma_t, \tag{3.8}$$

which under most circumstances would not reproduce the same reaction rate as the underlying FM sweeper.

An equivalent reaction rate could be recovered by calculating and storing an angular flux-weighed cross section, and using it in the S_N sweep. However, due to memory benefits discussed below, it is beneficial to instead seek a correction factor that may be applied similarly to the streaming correction factors. Introducing a correction factor, β^n, to the node-averaged flux, $\bar{\psi}^n$, the fine mesh reaction rate can be recovered,

$$R^n = \beta^n \bar{\psi}^n \Sigma_t. \tag{3.9}$$

Combining Eqs. (3.5) and (3.9), produces the definition

$$\beta^n = \frac{\bar{\Sigma}_t^n}{\Sigma_t}. \tag{3.10}$$

Incorporating the above correction factor into the CDD equation (Eq. (3.3)), gives

$$\begin{aligned} \bar{\psi}^n &= \beta^n \alpha_{x,ij}^n \left(\psi_{i+1/2,j}^n + \psi_{i-1/2,j}^n \right), \\ \bar{\psi}^n &= \beta^n \alpha_{y,ij}^n \left(\psi_{i,j+1/2}^n + \psi_{i,j-1/2}^n \right). \end{aligned} \tag{3.11}$$

At first, it appears that yet another another angularly-dependent quantity must be stored to employ this method. However, the entire quantity $\beta^n \alpha^n$ can easily be stored as a single value in memory.

3.2.3 CDD Sweep Algorithm

To incorporate the CDD equations into an S_N sweep algorithm, Eq. (3.3) is introduced into Eq. (3.2), yielding the expression

$$\bar{\psi}_{ij} = \frac{Q_{ij} + 2\left(\frac{\eta}{\Delta x_i}\psi_{i\mp 1/2,j} + \frac{\varepsilon}{\Delta y_j}\psi_{i,j\mp 1/2}\right)}{\frac{\eta}{\Delta x_i \beta \alpha_{x,ij}} + \frac{\varepsilon}{\Delta y_j \beta \alpha_{y,ij}} + \Sigma_{t,ij}}, \quad \eta \gtrless 0, \quad \varepsilon \gtrless 0, \tag{3.12}$$

which applies to solving purely 2-D problems.

In the normal way, the sweep starts at the upwind boundary of the spatial domain and uses the upwind boundary flux to calculate the node-average flux with Eq. 3.12. The node-average flux is then used to calculate the downwind surface flux using a rearranged form of Eqs. 3.3,

$$\begin{aligned} \psi_{i\pm 1/2,j} &= \frac{\bar{\psi}_{ij}}{\beta \alpha_{x,ij}} - \psi_{i\mp 1/2,j}, \quad \eta \gtrless 0 \text{ and} \\ \psi_{i,j\pm 1/2} &= \frac{\bar{\psi}_{ij}}{\beta \alpha_{y,ij}} - \psi_{i,j\mp 1/2}, \quad \varepsilon \gtrless 0 \end{aligned} \tag{3.13}$$

Generation of the correction factors for use in the CDD equations is a potentially costly operation, which relies on knowledge of the angular flux. For realistic problem sizes, the full angular flux is too large to store in memory. As explained in Section 2.4.2, most MoC sweeping algorithms avoid this by only storing the state of the angular flux for a single ray at a time as that ray traverses the problem geometry. Therefore, data needed to compute the CDD correction terms must be collected during the ray sweep itself. A slight modification is made to the MoC sweeper kernel such that every time a ray crosses a cell boundary of the CM, its angular flux contributes to a running average for the surface angular flux on that surface. This concept is illustrated in Fig. 3.4. Also, the ray segment-average flux is contributed to a running average of the cell-average angular flux. At the end of sweeping all rays for a given angle, the resultant angular flux quantities may be used to calculate correction factors for each coarse cell. The CDD equations require the storage of two

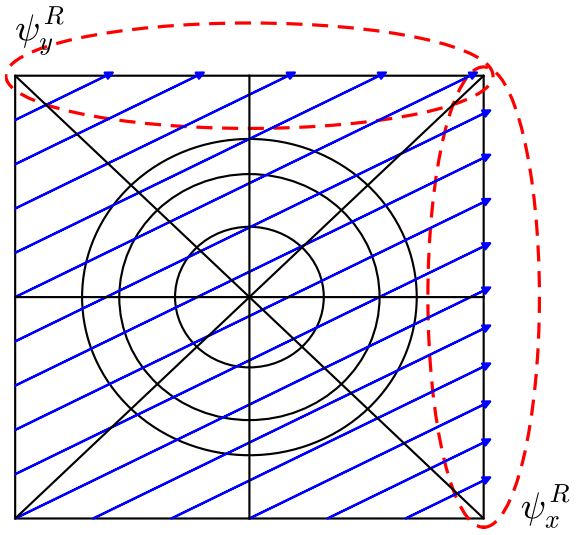


Figure 3.4: CM angular flux accumulation.

correction factors ($\beta\alpha_x$ and $\beta\alpha_y$) per pin cell per angle per energy group.

3.2.4 Coarse Mesh Equivalence

To demonstrate that the corrected diamond difference equations are capable of reproducing important quantities from a fine mesh sweep, a simple one-dimensional problem is examined. Figure 3.5 depicts our computational grid, which features coarse mesh regions that are divided into multiple fine mesh regions. While the figure shows all fine mesh regions being the same size, this is not a requirement. In the below analysis, it is shown that for a given incident flux on a coarse mesh region and fine mesh data in the form of the correction factors α^n and β^n, the exiting angular flux and bulk reaction rate from the fine mesh sweep can be preserved. This results in equivalent cell balance between the two methods and an "equivalent" global solution.

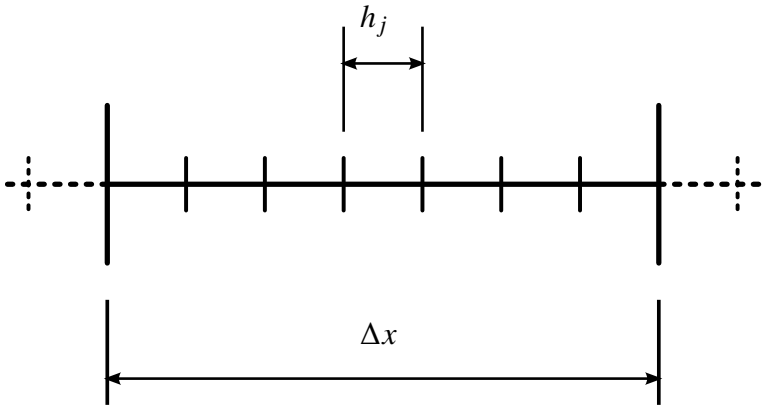


Figure 3.5: One-dimensional fine and coarse mesh.

Fine mesh projections

The quantities that are preserved are fine mesh values that have been projected onto the coarse mesh. The following definitions apply:

$$\bar{\Sigma}_t^n \equiv \frac{\sum\limits_{j\in\Delta x} \Sigma_{t,j}\bar{\psi}_j^n h_j}{\sum\limits_{j\in\Delta x} \bar{\psi}_j^n h_j}, \tag{3.14a}$$

$$Q_C = Q_F \equiv \frac{\sum\limits_{j\in\Delta x} Q_j h_j}{\Delta x}, \tag{3.14b}$$

and

$$\bar{\psi}_F^n \equiv \frac{\sum\limits_{j\in\Delta x} \bar{\psi}_j^n h_j}{\Delta x}. \tag{3.14c}$$

Fine mesh equation

For a single angle, $\hat{\mathbf{\Omega}}^n$, and energy, the fine mesh solution may be cast into a differential form similar to the S_N equation, even though the angular flux quantities may have come

from some other method, giving

$$\frac{\mu}{h_j}\left(\psi^n_{j+1/2} - \psi^n_{j-1/2}\right) + \bar{\psi}^n_j \Sigma_{t,j} = Q_j. \tag{3.15}$$

As with any other S_N-like sweeping method, the above is rearranged to obtain the outgoing angular flux,

$$\psi^n_{j+1/2} = \frac{Q_j + \frac{\mu}{h_j}\psi^n_{j-1/2} - \bar{\psi}^n_j \Sigma_{t,j}}{\frac{\mu}{h_j}}, \tag{3.16}$$

$$\psi^n_{j+1/2} = \psi^n_{j-1/2} + \frac{Q_j h_j}{\mu} - \frac{\bar{\psi}^n_j \Sigma_{t,j} h_j}{\mu}. \tag{3.17}$$

The above applies to flux propagation through a single fine-mesh region. The flux propagation through all fine-mesh regions in a coarse-mesh region can therefore be expressed as the following:

$$\psi^{n,R}_F = \psi^{n,L}_F + \sum_{j=1}^{J} \frac{Q_j h_j}{\mu} - \sum_{j=1}^{J} \frac{\bar{\psi}^n_j \Sigma_{t,j} h_j}{\mu}. \tag{3.18}$$

Referring back to the definitions in Eqs. (3.14), the above can be expressed as

$$\psi^{n,R}_F = \psi^{n,L}_F + \frac{\Delta x}{\mu}\left(Q - \bar{\Sigma}^n_t \bar{\psi}^n_F\right). \tag{3.19}$$

Coarse mesh equation

The discrete ordinates form of the transport equation for the coarse mesh region assumes a form similar to the fine mesh equations, Eq. (3.15):

$$\frac{\mu}{\Delta x}\left(\psi^{n,R}_C - \psi^{n,L}_C\right) + \Sigma_t \bar{\psi}^n = Q. \tag{3.20}$$

Instead of having known quantities for the angular flux, the CDD streaming and collision correction factors, Eq. (3.3), are inserted to get

$$\frac{\mu}{\Delta x}\left(\psi^{n,R}_C - \psi^{n,L}_C\right) + \Sigma_t \beta^n \alpha^n \left(\psi^{n,L}_C + \psi^{n,R}_C\right) = Q. \tag{3.21}$$

Applying the definition for β^n (3.10) and canceling out the isotropic Σ_t,

$$\frac{\mu}{\Delta x}\left(\psi_C^{n,R}-\psi_C^{n,L}\right)+\bar{\Sigma}_t^n\alpha^n\left(\psi_C^{n,L}+\psi_C^{n,R}\right)=Q. \tag{3.22}$$

Applying the definition of α^n,

$$\frac{\mu}{\Delta x}\left(\psi_C^{n,R}-\psi_C^{n,L}\right)+\bar{\Sigma}_t^n\left(\frac{\bar{\psi}_F^n}{\psi_F^{n,L}+\psi_F^{n,R}}\right)\left(\psi_C^{n,L}+\psi_C^{n,R}\right)=Q. \tag{3.23}$$

For a given state of the fine mesh flux, $\bar{\psi}$ and incoming angular flux, $\psi_F^{n,L}\equiv\psi_c^{n,L}$, it can be shown that the coarse mesh solution can reproduce the fine-mesh outgoing angular flux. Rearranging Eq. (3.19) for $\bar{\psi}_F^n$ and inserting into Eq. (3.23), gives

$$\frac{\mu}{\Delta x}\left(\psi_C^{n,R}-\psi_C^{n,L}\right)+\left(\frac{\bar{\Sigma}_t^n}{\psi_F^{n,L}+\psi_F^{n,R}}\right)\left(\psi_F^{n,L}-\psi_F^{n,R}+\frac{\Delta x}{\mu}Q\right)\frac{\mu}{\bar{\Sigma}_t^n\Delta x}\left(\psi_C^{n,L}+\psi_C^{n,R}\right)=Q. \tag{3.24}$$

After some manipulation, above reduces to our desired result,

$$\psi_C^{n,R}=\psi_F^{n,R}. \tag{3.25}$$

By the CDD equations, the resultant product of the cell-averaged $\bar{\psi}^C$ and the isotropic transport cross section, Σ_{tr}, will yield the same reaction rate and an equivalent coarse-mesh solution. This same analysis, while more complicated, may be extended to 2-D.

3.3 Numerical Results

The Corrected Diamond Difference S_N formulation by itself does not constitute a 2-D/3-D method. That said, it is still valuable to examine some of its behavior in 2-D scenarios before moving on to 3-D problems. In this section, a number of small, 2-D problems are used to calculate CDD correction factors, and their angular and energy dependence are presented. The 2-D C5G7 benchmark is then used to demonstrate experimentally that

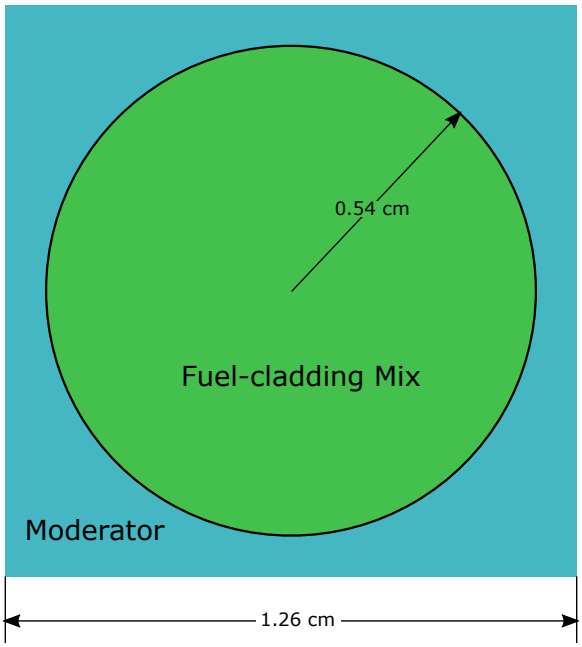


Figure 3.6: C5G7 pin geometry.

the CDD equations are indeed capable of reproducing the same result as the underlying FM solution that produced the CDD and cross section data, as though it were homogenized onto the coarse mesh. Lastly, the 2-D C5G7 benchmark is run with coupled MoC and CDD S_N to demonstrate that the CDD equations may be used to accelerate the convergence of the global eigenvalue problem.

3.3.1 Small Pin Array Test Cases

A number of small 2-D pin arrays were analyzed using the CDD method, mostly for illustrative purposes, and to study the correction terms in an accessible manner. Each of the cases uses pin cells derived from the C5G7 benchmark, which is described in detail and studied in more completeness in Sections 3.3.3 and 5.1. The pin geometry for the C5G7 benchmark is depicted in Fig. 3.6 and consists of single cylindrical regions with a radius of 0.54 cm, surrounded by moderator material, and arrayed in a square lattice with a pitch of 1.26 cm.

Two infinite lattice cases are presented, along with a case that features vacuum boundary

conditions. Plots of the variation of the correction factors, $\alpha_{x|y}$, with azimuthal angle are presented for each unit cell. The same product quadrature was used for all cases, with 12 azimuthal angles and 3 polar angles per octant. The values presented for α_x and α_y were extracted from all azimuthal angles corresponding to the middle polar angle. It was found that the qualitative behavior of the correction factors did not change significantly with polar angle. In all figures, the blue line indicates the x correction term, α_x, and the red line indicates the y correction term, α_y.

As noted in Section 3.2.1, values of the correction factors near $\frac{1}{2}$ constitute small corrections to the DD equations. Correction factors that deviate from $\frac{1}{2}$ indicate situations in which the flux variation within a coarse mesh region is more complicated.

One notable aspect shared by all cases is the large difference in behavior between energy groups; higher energy groups tend to exhibit far more angular variation of the correction factors than the lower energy groups. This is the result of smaller cross sections in the high-energy groups, leading to more streaming and a more anisotropic angular flux. In the lower energy groups, the correction factors take on a much smoother shape.

Another common behavior of the streaming correction factors is that they tend to deviate from their trivial value most in angles that are more normal to the streaming direction being corrected. For example, α_x tends to assume values furthest from 0.5 in angles that point mostly along the y axis. This is because angles normal to the direction of transmission produce apparent optical thicknesses that are much larger than otherwise. In extreme circumstances, this can be considered in context of the DD positivity condition,

$$\frac{\Delta x \Sigma_{\mathrm{t}}}{|\eta|} < 2, \tag{3.26}$$

which is discussed in more detail in Section 4.1.1. In short, as the direction cosine becomes small, the apparent optical thickness of the cell becomes large, and DD becomes more poorly-behaved. As a result, the corresponding correction factor steps in to compensate.

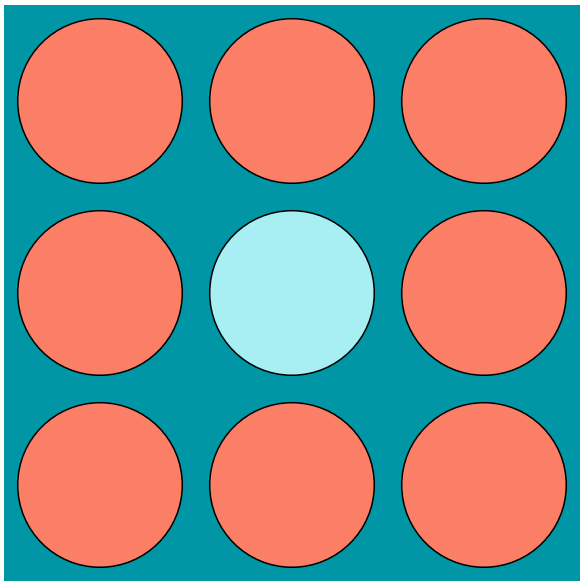


Figure 3.7: Unrodded 3×3 pin array, with reflective boundary conditions.

Unrodded 3-by-3 This case is an infinite lattice of 3-by-3 arrays of UO_2 fuel pins with a central water-filled control rod guide tube, as depicted in Fig. 3.7. Figures 3.8-3.10 depict the correction terms for select energy groups. In this case, the variation of the correction factors is relatively weak, though small peaks can be seen in the highest energy groups for the steepest angles.

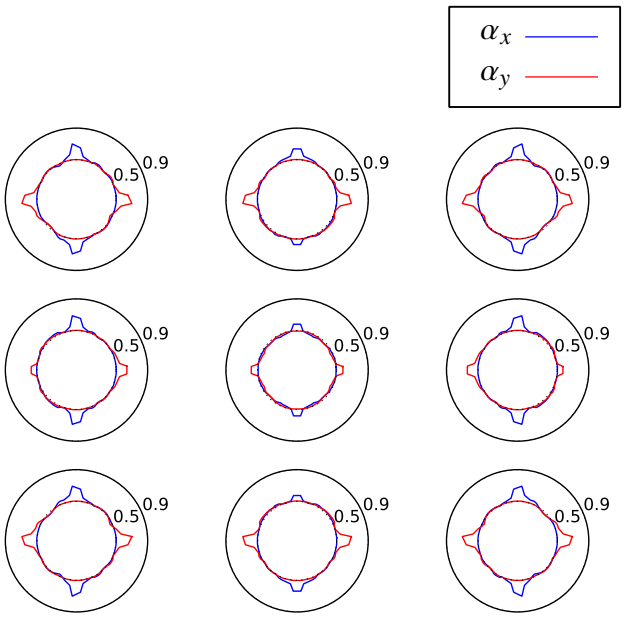


Figure 3.8: Group 1 correction terms for the unrodded 3×3 case.

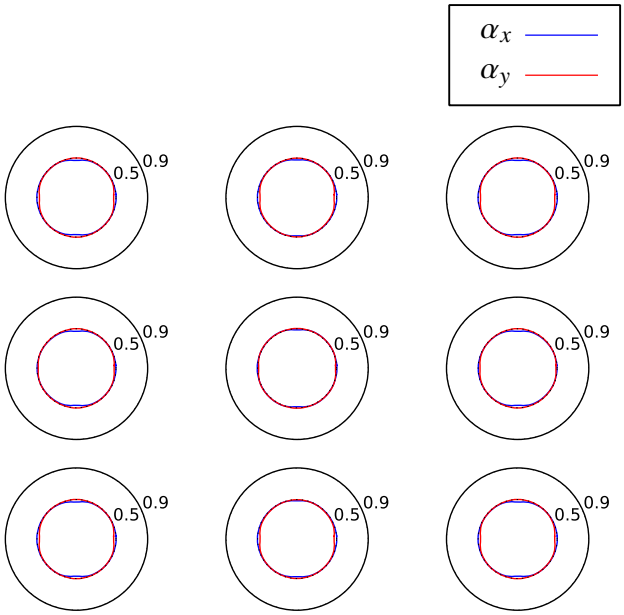


Figure 3.9: Group 4 correction terms for the unrodded 3×3 case.

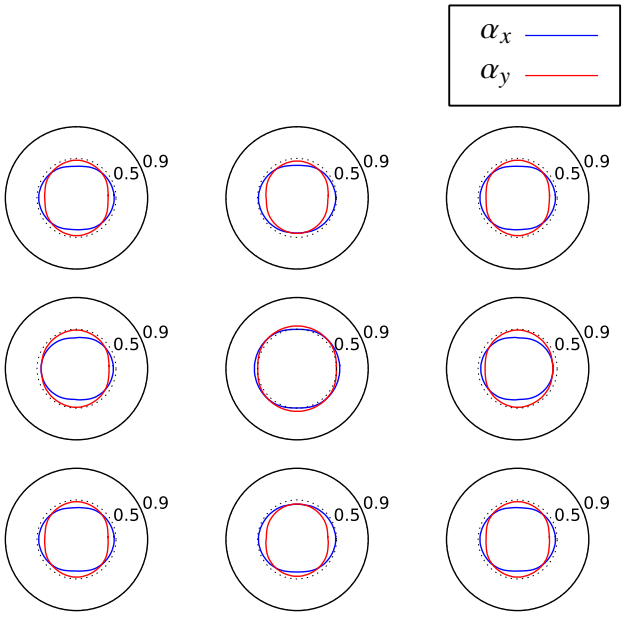


Figure 3.10: Group 7 correction terms for the unrodded 3×3 case.

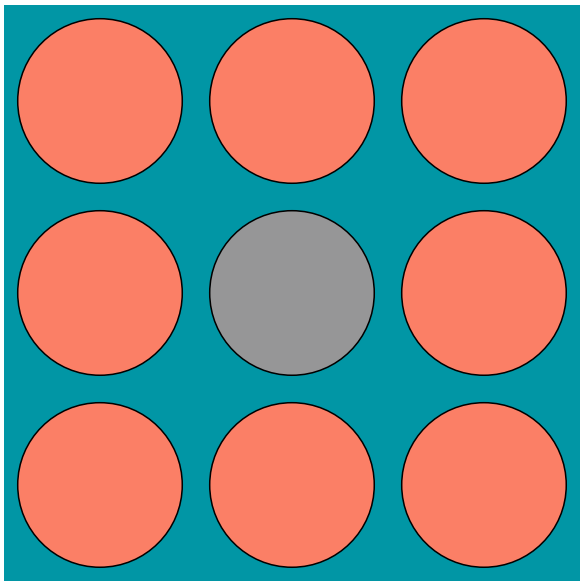


Figure 3.11: Rodded 3×3 pin array, with reflective boundary conditions.

Rodded 3-by-3 This case is the same as the lattice as above, with the guide tube replaced by an inserted control rod, as shown in Fig. 3.11. Figures 3.12-3.14 depict the correction terms for selected energy groups. At the highest energy group, the correction factors are much like those in the unrodded case. At lower-energies, the corrections assume a more interesting variation. Most notable is that the corrections in the control rod cell are uniformly less than 0.5; this is to be expected in a region with a strong neutron absorber in its center, as it results in a cell-average flux that is smaller than the cell edge flux.

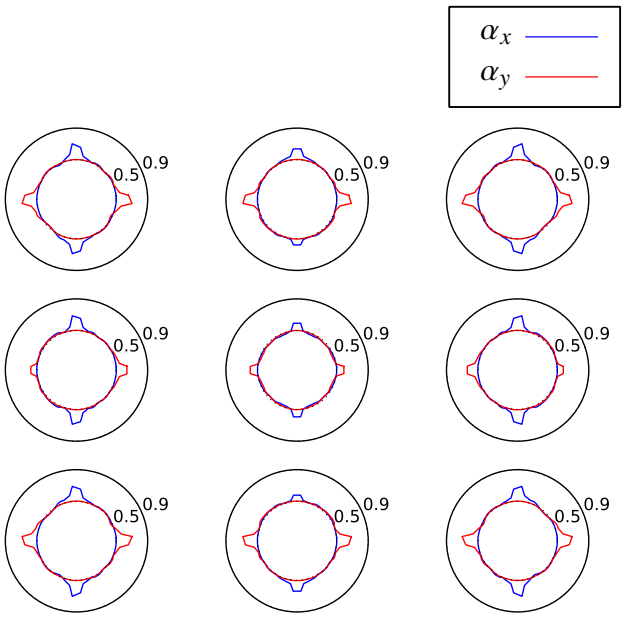


Figure 3.12: Group 1 correction terms for the rodded 3×3 case.

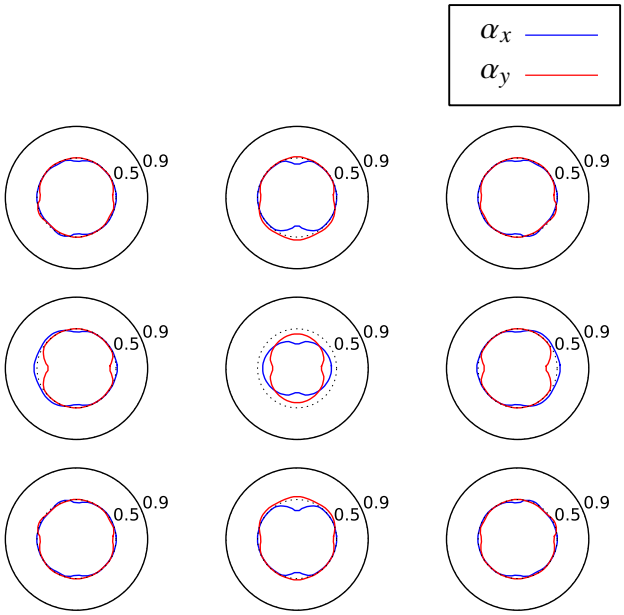


Figure 3.13: Group 4 correction terms for the rodded 3×3 case.

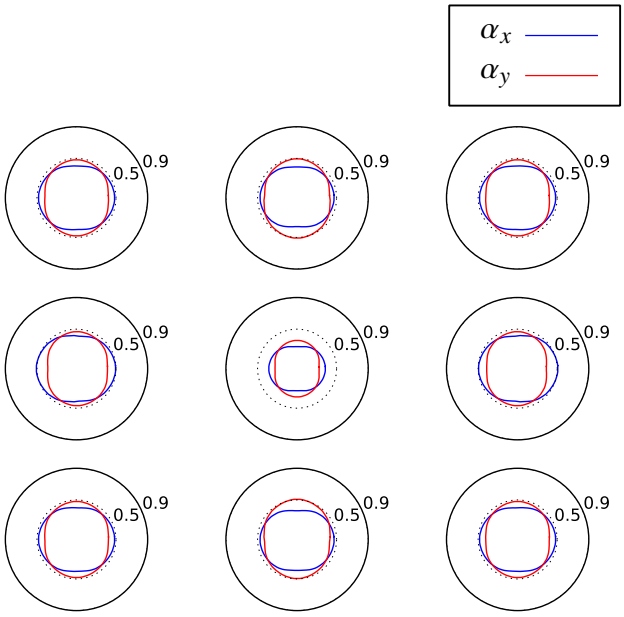


Figure 3.14: Group 7 correction terms for the rodded 3×3 case.

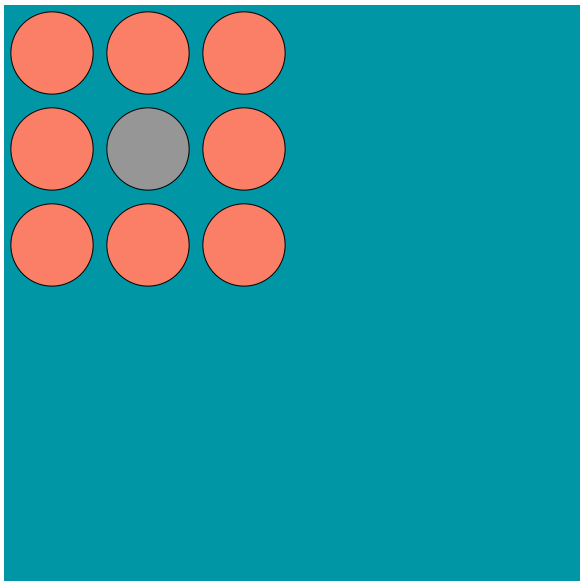


Figure 3.15: Rodded 3×3 pin array, with reflector and vacuum boundary conditions.

3-by-3 with reflector and vacuum boundary This case is similar to the rodded 3×3 case, with the addition of a three pin-width reflector and vacuum boundary conditions to the right and bottom, shown in Fig. 3.15. Figures 3.16-3.18 show only the x correction factors for clarity. Since the problem is symmetric, the y correction factors are a 90 degree reflection of the x correction factors. The addition of the vacuum boundary gives rise to much more dramatic corrections. The large angular spikes inside and immediately neighboring the fuel rods arise from streaming paths between the pins themselves. Another interesting feature is the angular discontinuity in the correction factors on the edge reflector regions, where the inward-facing angles see positive corrections to the cell-average flux. These occur due to the inability of the DD equations, on such a coarse mesh, to resolve the boundary layer effects near the vacuum boundary.

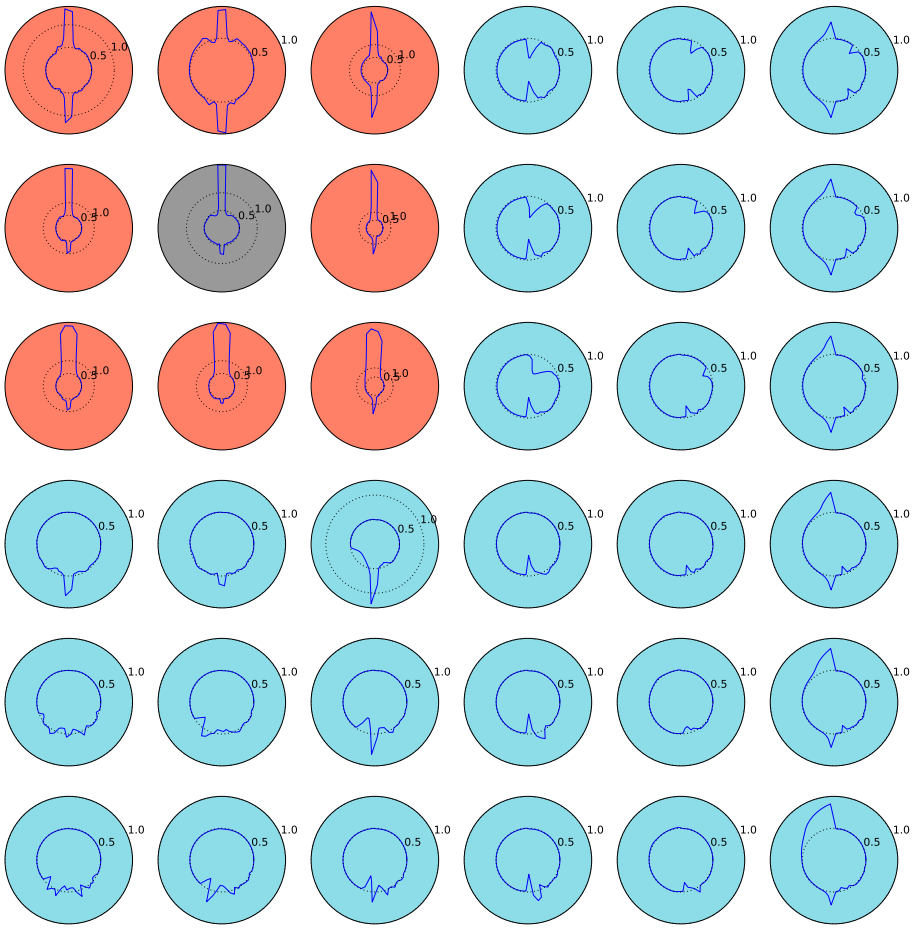


Figure 3.16: α_x for rodded 3×3 case with reflector and vacuum boundary, group 7.

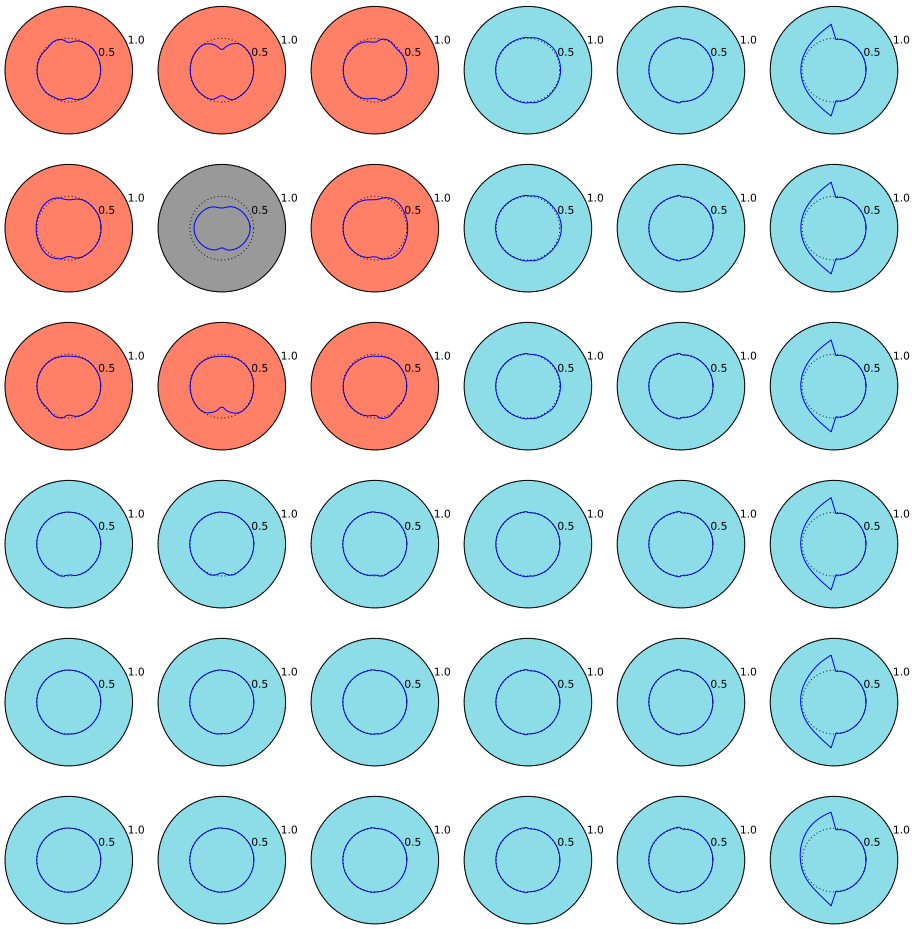


Figure 3.17: α_x for rodded 3×3 case with reflector and vacuum boundary, group 4.

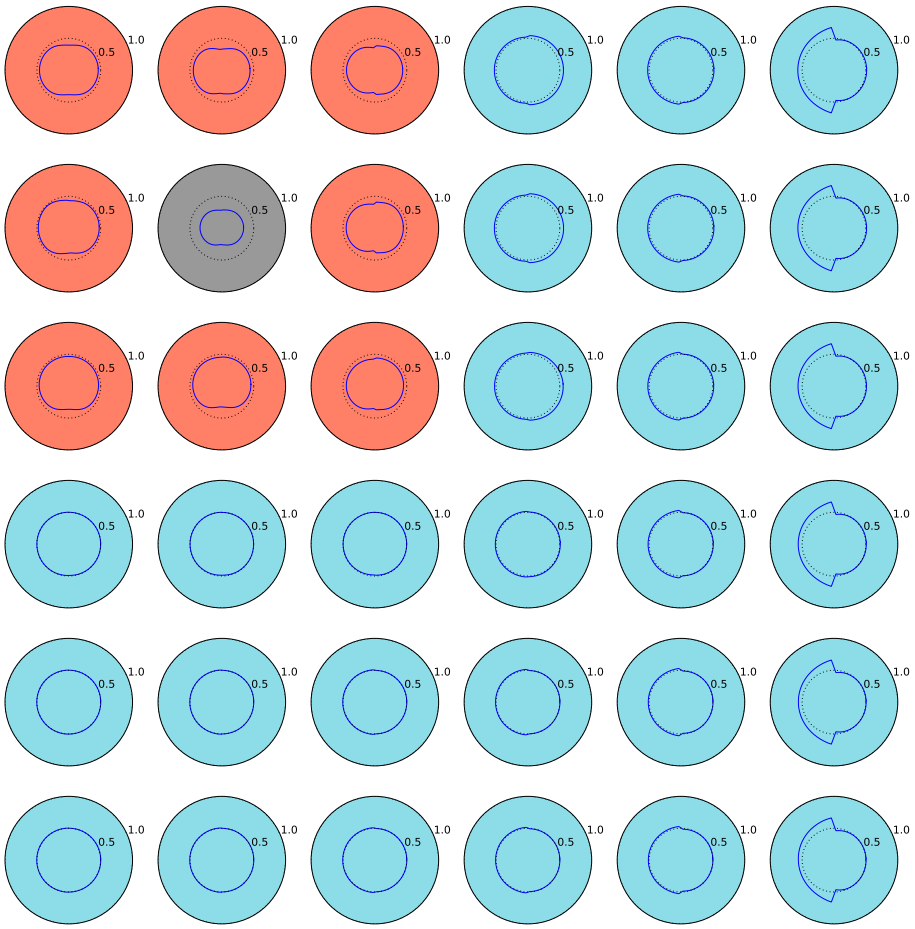


Figure 3.18: α_x for rodded 3×3 case with reflector and vacuum boundary, group 1.

3.3.2 2-D C5G7

A number of analyses were performed using the 2-D C5G7 benchmark [26]. First, it was used to verify experimentally that the CDD S_N formulation is equivalent to the underlying fine-mesh MoC solution, as well as to study the nature of the correction factors in a more realistic setting than the pin arrays discussed in the previous section. Lastly, the performance of a simple coupled MoC-S_N iteration scheme was investigated.

The 2-D C5G7 benchmark is a small, heterogeneous reactor core with simplified pin geometry, identical to that used in the pin arrays discussed above (see Fig 3.6). The benchmark provides seven-group macroscopic cross sections describing seven different materials: four fuel compositions, control rod guide tube, assembly-centered fission chamber, and moderator. As depicted in Fig. 3.19, the quarter-symmetric core is comprised of a 2×2 array of assemblies, surrounded by a solid blanket of moderator, which is referred to as the "radial reflector."

3.3.3 2-D Convergence

To experimentally demonstrate the 2-D equivalence between the fine-mesh MoC solution and the coarse-mesh S_N solution, a 2-D C5G7 benchmark calculation was performed using a two-step process.

An MoC calculation was performed, and the correction factors from Sections 3.2.1 and 3.2.2 were calculated using the converged solution, along with flux-weighted cross sections. These data were then used in a 2-D S_N calculation on the pin-homogenized mesh[1]. The MoC calculation yielded identical system eigenvalues of 1.18570. Furthermore, the scalar flux distribution from the S_N calculation was the same as the pin-homogenized MoC flux distribution.

Figure 3.20 shows the values of the weighting terms as calculated by Eq. (3.11) for a

[1] A pin-homogenized mesh was used for purely practical reasons. This method can accommodate finer S_N meshes, however collecting the correction data on such a mesh becomes onerous. Section 6.3 discusses the potential benefits of moving to a finer S_N mesh.

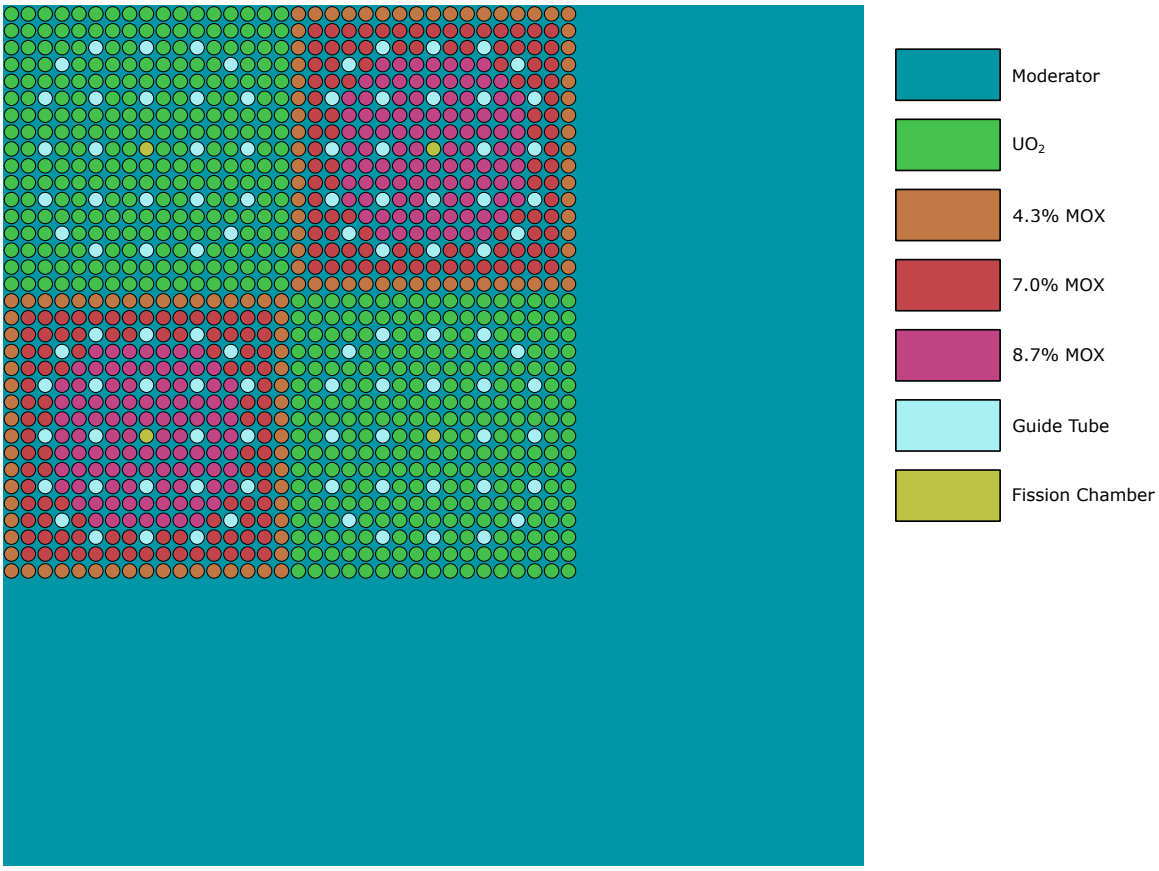


Figure 3.19: 2-D C5G7 benchmark geometry. Presented as modeled, with quarter symmetry; top and left boundaries are reflective, bottom and right boundaries vacuum.

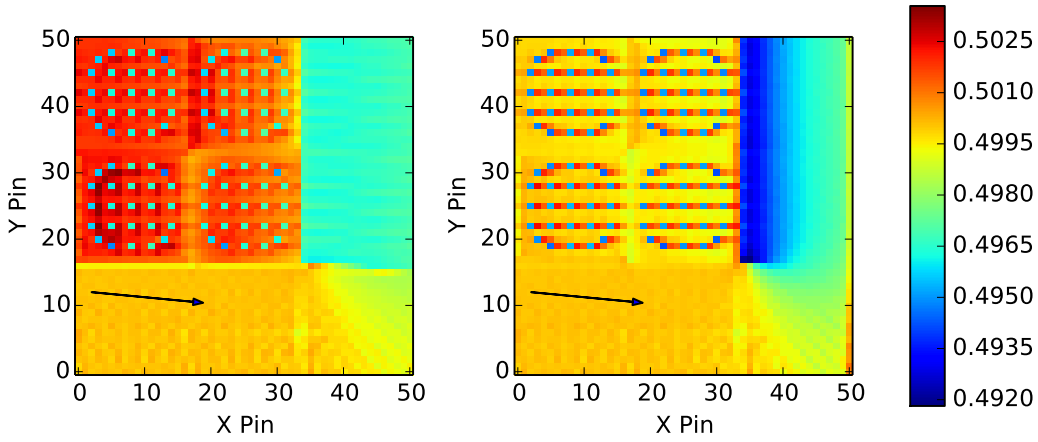


Figure 3.20: CDD correction factors, α_x (left) and α_y (right) for the angle $\hat{\mathbf{\Omega}} = (0.602\hat{i} - 0.06\hat{j} + 0.797\hat{k})$ and energy group one. The arrow indicates the direction of neutron travel.

single angle and energy. As one might expect, the largest corrections are present in regions where there are strong heterogeneities normal to the direction of neutron travel. The largest magnitude correction factors are observed in the portion of the radial reflector downwind of the active core region. The lower portion of the radial reflector experiences the smallest correction, since the streaming path of neutrons through that region only encounter the homogeneous reflector region.

CHAPTER 4

CDD for 2-D/3-D Applications

In Chapter 3 it was discussed how the CDD auxiliary equations, applied to 2-D problems using an embedded MoC sweeper, are capable of producing equivalent solutions to the underlying FM solution from which the correction terms and cross sections were derived. In the context of 2-D problems, this aspect of the CDD equations has only marginal value, since having already obtained a FM solution, performing extra computation just to reproduce a coarsened version of the same solution makes little sense[1]. However, CDD becomes useful when applied to 3-D problems, which can no longer by treated with 2-D MoC alone. In such cases it is possible calculate correction factors and cross sections for each axial subdomain of the 3-D mesh using 2-D MoC sweepers, then solve the global, 3-D problem on a pin-homogenized mesh with an S_N sweeper. Such approaches are referred as 2-D/3-D methods.

2-D/3-D comprises a class of methods which mostly vary in the nature of the coupling between the S_N and MoC sweepers, and the approach used to treat the axial dimension in the 3-D S_N sweeper. In this chapter, a number of options are introduced for each, and their effectiveness is investigated. Two of many possible axial treatments are described. A one-way and a transverse leakage-based two-way S_N-MoC interface are also developed.

[1]The equivalence property makes it possible to use the S_N sweeper to accelerate the convergence of 2-D MoC. Preliminary results have shown progress, and are discussed in Section 6.3

4.1 Axial Differencing

When interfaced with a 2-D MoC sweeper, the CDD equations only receive correction factors for the radial (x and y) directions. How these correction factors are calculated and provided to the S_N sweeper is discussed in more detail in Section 4.2. The axial dimension remains uncorrected and still requires an auxiliary equation. Many traditional S_N auxiliary relations may be used to treat the axial dimension alongside the CDD relations. In this section, sweeping strategies based upon the following diamond difference and step characteristic auxiliary relations are discussed.

4.1.1 Diamond Difference

One of the easiest approaches to handling the axial dimension in a 3-D CDD sweep is to use the conventional (uncorrected) DD equation. Performing similar steps used to produce Eq. (3.12), the following expression for the cell-average flux is obtained

$$\overline{\psi}_{ijk} = \frac{Q_{ijk} + 2\left(\frac{\eta}{\Delta x_i}\psi_{i\mp 1/2,j,k} + \frac{\varepsilon}{\Delta y_j}\psi_{i,j\mp 1/2,k} + \frac{\mu}{\Delta z_k}\psi_{i,j,k\mp 1/2}\right)}{\frac{\eta}{\Delta x_i \beta\alpha_{x,ijk}} + \frac{\varepsilon}{\Delta y_j \beta\alpha_{y,ijk}} + \frac{2\mu}{\Delta z_k} + \Sigma_{t,ijk}},$$

$$\eta \gtrless 0, \quad \varepsilon \gtrless 0, \quad \mu \gtrless 0. \tag{4.1}$$

After determining $\bar{\psi}$, the downwind fluxes may be calculated as

$$\psi_{i\pm 1/2,j,k} = \frac{\bar{\psi}}{\beta\alpha_{x,ij}} - \psi_{i\mp 1/2,j,k}, \tag{4.2a}$$

$$\psi_{i,j\pm 1/2,k} = \frac{\bar{\psi}}{\beta\alpha_{y,ij}} - \psi_{i,j\mp 1/2,k}, \text{ and} \tag{4.2b}$$

$$\psi_{i,j,k\pm 1/2} = 2\bar{\psi} - \psi_{i,j,k\mp 1/2}. \tag{4.2c}$$

The diamond difference equations are well known to exhibit second-order convergence [24] and are astonishingly easy to evaluate computationally. However, they do possess a

number of negative characteristics which are well known, and pose challenges to their use in a 2-D/3-D solver.

One issue with the DD equations is that they tend to exhibit oscillatory behavior in certain cases. These oscillations in the flux can not only be very inaccurate, but tend to cause issues when coupled with other solution techniques.

Second, and more importantly, they are not positive; it is possible for DD to produce negative angular fluxes in the presence of positive source and incoming angular flux. A relevant positivity condition can be determined by considering a 1-D DD scheme,

$$\frac{\mu}{\Delta x}\left(\psi_{i+1/2}-\psi_{i-1/2}\right)+\Sigma_{\mathrm{t}}\bar{\psi}_i = Q_i, \tag{4.3a}$$

$$\bar{\psi}_i = \frac{1}{2}\left(\psi_{i+1/2}+\psi_{i-1/2}\right). \tag{4.3b}$$

Using the DD equation to eliminate the cell-average flux, $\bar{\psi}$ yields an expression for the downwind flux:

$$\psi_{i\pm1/2} = \frac{Q+\left(\frac{|\mu|}{\Delta x}-\frac{\Sigma_{\mathrm{t}}}{2}\right)\psi_{i\mp1/2}}{\frac{|\mu|}{\Delta x}+\frac{\Sigma_{\mathrm{t}}}{2}}, \quad \mu \gtrless 0. \tag{4.4}$$

For the purposes of determining positivity, we need only concern ourselves with the numerator of Eq. (4.4), since the cross section is always positive, giving the condition

$$Q+\psi_{i\mp1/2}\left(\frac{|\mu|}{\Delta x}-\frac{\Sigma_{\mathrm{t}}}{2}\right) > 0, \quad \mu \gtrless 0. \tag{4.5}$$

Since the positivity condition should hold for zero incoming flux and zero inhomogeneous source, it simplifies to

$$\frac{\Delta x \Sigma_{\mathrm{t}}}{|\mu|} < 2. \tag{4.6}$$

This result has a natural physical interpretation: if the apparent optical thickness of a cell is too large, it becomes possible for the resultant flux to be negative, especially if the incoming flux or inhomogeneous source are small.

It should be noted in the case of 2-D/3-D that the correction factors applied to the radial

x and y dimensions enforce positivity upon the flux in those directions at convergence, and in the absence of any axial streaming, so long as the FM scheme used to produce the correction factors is also positive. In the case of MoC, this is true. This is because the CDD correction factors are calculated using fluxes that are unconditionally-positive, and enforce an equivalent relationship in the CDD solution. When the S_N solution is unconverged, it is possible for negative fluxes to arise, though this was never witnessed in any of the cases run as part of this work.

Including the uncorrected DD equation as our axial treatment makes the possibility of negative solutions far more likely, and care must be taken to ensure that the axial mesh is sufficiently fine. While the resultant scalar flux may still be positive due to contributions from other angles, these negative angular fluxes are clearly unphysical and troubling. In certain situations, even if the scalar flux is positive, the negative angular fluxes can wreak havoc with CMFD acceleration by producing pathologically large $\hat{D}$ terms. It was found that in some of the cases studied as part of this work, CMFD failed to converge due to these negative fluxes. This behavior is discussed more in Chapter 5.

It is common in situations where there is concern about negative fluxes to employ a negative flux "fix-up." This flux fix-up is performed by checking if ψ_z^R, calculated using Eqs. (4.1) and (4.2c) is negative. If so, it is set artificially to zero, and $\bar{\psi}$ is recalculated using the expression

$$\overline{\psi}_{i,j,k} = \frac{Q_{i,j,k} + 2\left(\frac{\eta}{\Delta x_i}\psi_{i\mp 1/2,j,k} + \frac{\varepsilon}{\Delta y_j}\psi_{i,j\mp 1/2,k}\right) + \frac{\mu}{\Delta z_k}\psi_{i,j,k\mp 1/2}}{\frac{\eta}{\Delta x_i \beta \alpha_{x,ijk}} + \frac{\varepsilon}{\Delta y_j \beta \alpha_{y,ijk}} + \Sigma_{t,ijk}}. \tag{4.7}$$

While this tends to result in a more robust CMFD acceleration, it can have noticeably adverse impacts on the accuracy of the solution. It was found that the negative flux fix-up was necessary to run some of the benchmarks discussed in Chapter 5.

4.1.2 Step Characteristic Axial Treatment

Another option for axial differencing is to use the Step Characteristics (SC) auxiliary equation. The Step Characteristics (SC) differencing scheme assumes an exponential shape to the angular flux within a cell, and can be derived using the same characteristic form of the transport equation, Eq. (2.35), used to derive MoC. It gives the auxiliary relation

$$\bar{\psi}_k^n = \rho_k^n \psi_{k\mp 1/2}^n + \left(1-\rho_k^n\right)\psi_{k\pm 1/2}^n, \quad \mu^n \gtrless 0, \tag{4.8}$$

where

$$\rho_k^n = \frac{1}{\tau_k^n} - \frac{1}{e^{\tau_k^n}-1} \tag{4.9}$$

and τ_k^n is the apparent optical thickness of cell k for angle n,

$$\tau_k^n = \frac{\Sigma_{t,k}\Delta z_k}{|\mu^n|}. \tag{4.10}$$

It is easy to see that unlike the DD scheme, SC is positive whenever the source and incoming flux are positive. While in 1-D, SC exhibits second-order convergence, as shown in Chapter 5, it tends to be noticeably less accurate than DD in many cases.

The SC auxiliary equation is applied alongside the CDD equations to yield a 2-D/3-D method:

$$\begin{aligned}
&\frac{\eta}{\Delta x}\left(\psi_{i+1/2,j,k} - \psi_{i-1/2,j,k}\right) \\
&+ \frac{\varepsilon}{\Delta y}\left(\psi_{i,j+1/2,k} - \psi_{i,j-1/2,k}\right) \\
&+ \frac{\mu}{\Delta z}\left(\psi_{i,j,k+1/2} - \psi_{i,j,k-1/2}\right) + \Sigma_{\mathrm{tr},ijk}\bar{\psi}_{ijk} = Q_{ijk}
\end{aligned} \tag{4.11a}$$

$$\bar{\psi}_{ijk} = \beta\alpha_{x,ijk}\left(\psi_{i+1/2,j,k} + \psi_{i-1/2,j,k}\right) \tag{4.11b}$$

$$\bar{\psi}_{ijk} = \beta\alpha_{y,ijk}\left(\psi_{i,j+1/2,k} + \psi_{i,j-1/2,k}\right) \tag{4.11c}$$

$$\bar{\psi}_{ijk} = \rho\psi_{i,j,k\mp1/2} + (1-\rho)\psi_{i,j,k\pm1/2}, \quad \mu \gtrless 0 \tag{4.11d}$$

Eliminating instances of downwind flux from Eq. (4.11a) using Eqs. (4.11b)-(4.11d) gives

$$\begin{aligned}
&\frac{\eta}{\Delta x}\left(\frac{\bar{\psi}_{ijk}}{\beta\alpha_{x,ijk}} - 2\psi_{i\mp1/2,j,k}\right) \\
&+ \frac{\varepsilon}{\Delta y}\left(\frac{\bar{\psi}_{ijk}}{\beta\alpha_{y,ijk}} - 2\psi_{i,j\mp1/2,k}\right) \\
&+ \frac{\mu}{\Delta z}\left(\frac{\bar{\psi}_{ijk} - \rho\psi_{i,j,k\mp1/2}}{1-\rho} - \psi_{i,j,k\pm1/2}\right) + \Sigma_{tr}\bar{\psi} = Q_{ijk}, \\
&\eta \gtrless 0, \quad \varepsilon \gtrless 0, \quad \mu \gtrless 0.
\end{aligned} \tag{4.12}$$

Collecting like terms and rearranging for $\bar{\psi}_{ijk}$ yields

$$\bar{\psi}_{ijk} = \frac{Q_{ijk} + \frac{2\eta}{\Delta x_i}\psi_{i\mp1/2,j,k} + \frac{2\varepsilon}{\Delta y_j}\psi_{i,j\mp1/2,k} + \frac{\mu}{\Delta z_k(1-\rho)}\psi_{i,j,k\mp1/2}}{\frac{\eta}{\Delta x_i\beta\alpha_{x,ijk}} + \frac{\varepsilon}{\Delta y_j\beta\alpha_{y,ijk}} + \frac{\mu}{\Delta z_k(1-\rho)} + \Sigma_{tr,ijk}}, \tag{4.13}$$

with downwind fluxes calculated using the auxiliary relations in Eqs. (4.11b)-(4.11d).

4.2 2-D/3-D Iteration Schemes

In the previous sections, it was discussed how to calculate correction factors and flux-weighted cross sections from an MoC solver, as well as how such data might be used in a 3-D S_N sweeper. The final step is to integrate these two components into a 2-D/3-D method for solving actual 3-D problems.

4.2.1 One-Way Coupling

The simplest approach to generating correction factors and cross sections from a series of 2-D MoC solvers is to do just that:

1. Break the problem into axial slabs,
2. model each slab as a 2-D problem (implicitly-reflective axial boundary conditions) using MoC,
3. calculate flux-weighted cross sections and correction factors in the last sweep, and
4. compose the data from the slabs into a separate 3-D S_N calculation.

This was the first approach that was tried, due to the ease with which it can be implemented in existing transport codes. While somewhat contrived, it provided good preliminary results which motivated further development of the method described in Section 4.2.2. It also has a number of advantages and disadvantages worthy of note.

First, the treatment of each axial slab as completely independent of all others permits a great deal of re-use of the correction factors and cross sections. The usefulness of this feature is highly dependent upon the type of problem to be solved, but in the benchmark problems such as the 3-D C5G7 rodded configurations discussed in Chapter 5, this re-usability feature finds much utility. On the other hand, the treatment of each plane individually causes a number of issues. Most obviously, in the absence of any axial streaming the resultant cross sections and correction factors are unlikely to be representative of what they should be if axial behavior were taken into account. Less apparent, but arguably more important, is the difficulty it presents in producing *any* useful data in axial regions devoid of multiplying media; in such cases without a source of neutrons, one is forced to rely upon volume-weighted cross sections and no correction factors at all.

While this behavior is discussed in more detail in Chapter 5, in general it was found that the one-way coupling technique managed to perform relatively well considering its

crudeness. As applied to the 3-D C5G7 benchmark cases, its primary defects were found in the region of the core closest to the axial reflector. This was due to the lack of valid data for the axial reflector itself and the strong axial streaming effects in the vicinity of the active core/axial reflector interface. In the next section, a more sophisticated method is developed, which addresses both of these deficiencies.

4.2.2 Transverse Leakage Coupling

In the previous section, a crude method was discussed for generating CDD correction factors and flux-weighted cross sections to be used in a 3-D S_N solver. While it performed well considering its simplicity, the lack of reasonable cross sections and correction factors for the axial reflectors, and neglecting the axial streaming effects resulted in poor performance in certain situations. This section describes a method similar to 2-D/1-D to address both of these issues.

Rather than perform the 2-D MoC calculations for each plane separately, they are coupled using TL sources calculated iteratively from the 3-D S_N sweeper, which in turn uses correction factors and cross sections derived from the 2-D MoC sweepers. The coupled sweepers are then accelerated using CMFD. The 3-D currents used to calculate the CMFD $\hat{D}$ factors are extracted from the S_N sweeper. The transverse leakage source is updated following each CMFD solve, and a new iteration begins. This process is shown in Fig. 4.1, and is referred to as the "two-way" or "transverse leakage-based" coupling.

As with 2-D/1-D methods, the 3-D S_N sweeper is operating on a pin-homogenized mesh, and therefore the flat TL approximation is used in this work, though the use of shape functions may be considered in future work. Furthermore, while access to the full angular flux variable is possible during the S_N sweep, an isotropic approximation is made to the axial TL provided to the MoC sweepers:

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y}\right) \psi_g^n(x,y) + \Sigma_{\mathrm{t},g} \psi_g^n(x,y) + L_n^g = Q_g^n(x,y), \tag{4.14a}$$

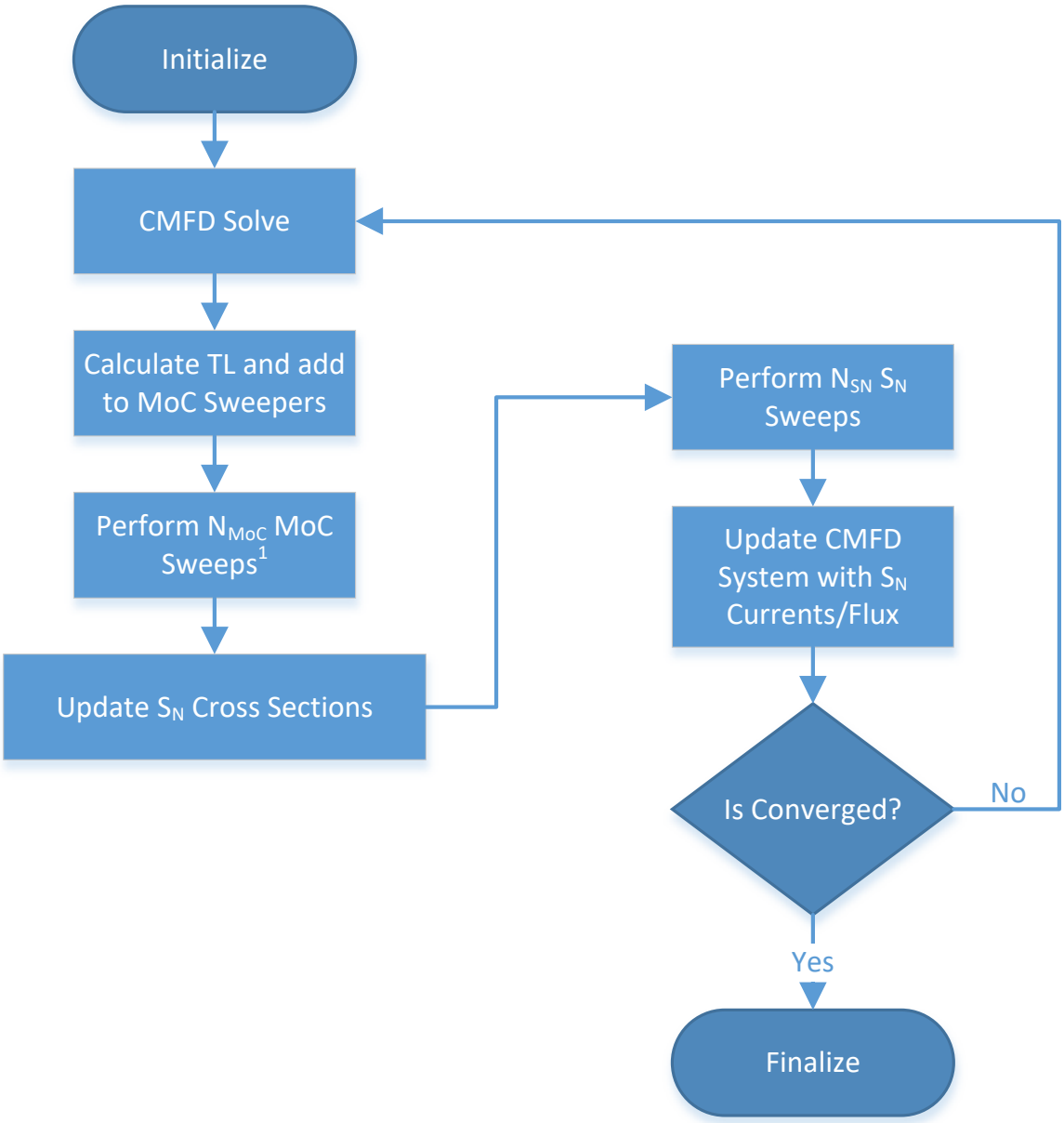


Figure 4.1: Transverse leakage iteration scheme.

$$
\begin{aligned}
L_g^n &= \frac{\mu^n}{h_z}\left(\psi_{g,T}^n(x,y) - \psi_{g,B}^n(x,y)\right) \\
&\approx \frac{\mu^n}{h_z}\left(\psi_{g,T}^n - \psi_{g,B}^n\right) \\
&\approx \frac{J_g^T - J_g^B}{h_z},
\end{aligned} \tag{4.14b}
$$

where J^T and J^B are the neutron currents across the top and bottom surfaces of the MoC pin, respectively, and defined as

$$
\begin{aligned}
J_g^T &= \sum_{n\in N} w^n \mu^n \psi_{g,T}^n \text{ and} \\
J_g^B &= \sum_{n\in N} w^n \mu^n \psi_{g,B}^n.
\end{aligned} \tag{4.15}
$$

These currents are calculated either using a CMFD solution, if used, or throughout the course of a S_N sweep if not.

As with step characteristics, MoC has the property that given a non-negative source and incoming angular flux, it will always produce positive fluxes. With the introduction of TL sources, however, it becomes possible for the total source to be negative. Indeed, in most problems with axial vacuum boundary conditions, it is expected that the overall TL source will be negative due to net axial leakage. In regions with large scattering and fission sources, or with large incident fluxes, this tends not to result in negative fluxes. In regions where the flux is small, absent of fission and with little in-scattering (e.g. high energy groups in reflector regions), it is possible for TL to drive the total source negative enough to produce negative fluxes. In many cases, the MoC can have difficulty recovering from this situation, leading to divergence.

To avoid this, transverse leakage "splitting" is used. Splitting involves dividing the TL source by the angular flux and incorporating it as an augmentation to the transport cross section:

$$
\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y}\right)\psi_g^n(x,y) + \left(\Sigma_{t,g} + \frac{L_n^g}{\psi_g^n(x,y)}\right)\psi_g^n(x,y) = Q_g^n(x,y). \tag{4.16}
$$

Since this addition to the cross section must occur before the sweep takes place, and because angular flux variable is not stored explicitly, the following approximation is used:

$$\left(\eta^n \frac{\partial}{\partial x} + \varepsilon^n \frac{\partial}{\partial y}\right)\psi_g^n(x,y) + \left(\Sigma_{\mathrm{t},g} + \frac{4\pi L_n^g}{\phi_g(x,y)}\right)\psi_g^n(x,y) = Q_g^n(x,y), \tag{4.17}$$

where the scalar flux, ϕ, is drawn from the previous iteration. Due to its approximate nature, transverse leakage splitting is only applied in regions where the total source, before adding self-scattering, would be negative.

It can be shown that the coarse mesh equivalence of the CDD equations holds in the presence of explicit, angle-dependent transverse leakage, and when the MoC and S_N axial meshes are the same. However, imposing such transverse leakages would be very onerous for a number of reasons. First, 2-D MoC sweepers typically assume axial symmetry, allowing them to only have to treat half of the angular space; applying angle-dependent transverse leakage would require treating both the positive and negative half-spaces, effectively doubling the amount work in an MoC sweep. Second, storage of the transverse leakages would effectively require the storage of another correction factor, increasing the burden of 2-D/3-D by 50 percent. Therefore, this work constrains itself to isotropic transverse leakage treatment, which has important ramifications.

Furthermore, being responsible for resolving the axial shape of the neutron flux, it is necessary for the S_N mesh to be appreciably finer in the axial direction than the MoC planes. In the analyses contained in this work, meshes similar to that shown in Fig. 4.2 are used. The S_N sweeper operates on a radially-coarse, axially-fine orthogonal mesh, while the MoC sweepers operate on an axially-coarse, radially-fine mesh. Correction factors and flux-weighted cross sections are stored on the MoC axial mesh, giving considerable savings in memory.

Due to the above, unlike 2-D/1-D, where the axial and radial sweepers maintain equivalence, the 3-D S_N sweeper in 2-D/3-D has an extra degree of freedom, and the requirement

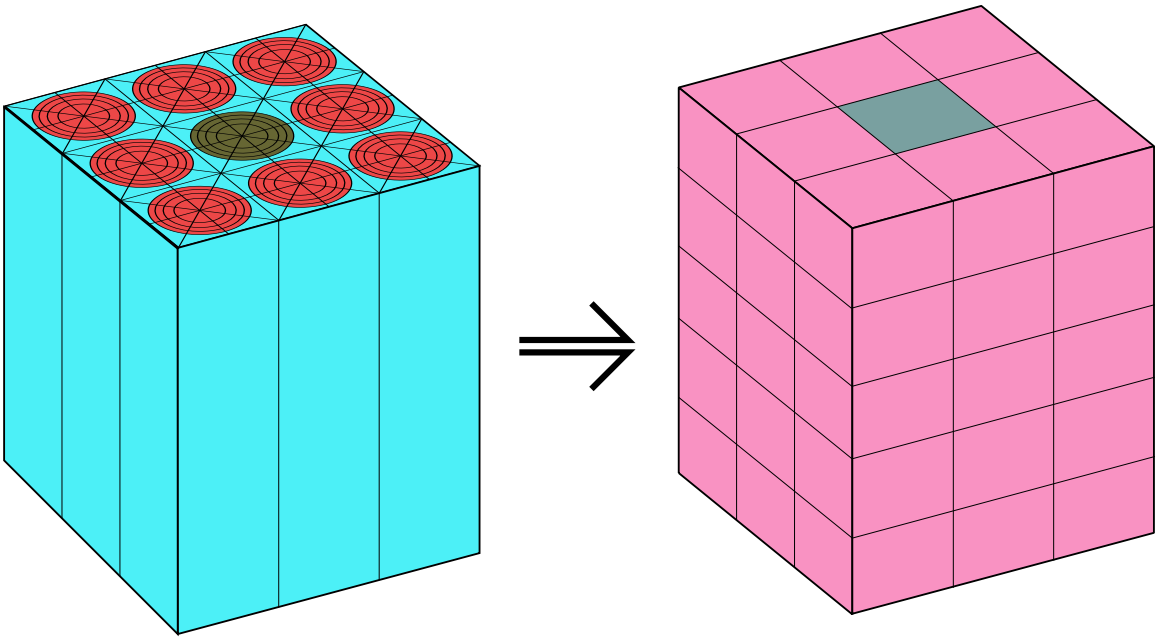


Figure 4.2: MoC and S_N meshes.

for equivalence is relaxed. Instead of the MoC and S_N sweepers converging to the same solution (as it would be homogenized onto a coarse axial, coarse radial mesh), they converge to some residual, which may not necessarily be zero. This behavior is investigated in more detail in Chapter 5, and this residual tends to be rather small. To allow for this discrepancy between the MoC and S_N sweepers, yet still converge the global eigenvalue problem, the state of the S_N sweeper is ultimately used to calculate updated eigenvalues and fission sources for each iteration of the eigenvalue solver. The state of the MoC sweeper is also used to ensure that the FM solution is also sufficiently converged. In this way, it is easiest to think of the 2-D/3-D method as 3-D, orthogonal-mesh S_N with embedded MoC.

CHAPTER 5

Results and Analysis

5.1 C5G7 Benchmark

The 2-D/3-D method was applied using both one-way and two-way, transverse leakage-based coupling, to the rodded C5G7 benchmarks [29] the results of which are presented in this section. The C5G7 benchmark is a small quarter-symmetric reactor, which supplies 7-group macroscopic cross sections to describe each of the materials. The active core region is comprised of a two-by-two array of assemblies, each a square lattice of 17-by-17 fuel pins. Two of the assemblies are composed primarily of Uranium Dioxide (UO_2) pins, while the other two consist mostly of mixed oxide (MOx) fuel pins of various Plutonium concentrations. Rather than resolve the full pin geometry, including cladding and gap, the fuel regions are a mixture of the fuel and cladding materials and modeled as a single cylinder of radius 0.54 cm as depicted in Fig 3.6. Each assembly contains a central fission chamber and a bundle of control rod guide tubes, all of which employ the same geometry as the fuel pins. A detailed top view of the active core region is presented in Fig. 3.19. The active core region is 42.84 cm high with a 21.42 cm-tall axial reflector above. The top surface of the axial reflector is a vacuum boundary condition, while the bottom of the active core region is reflective. This produces a rather strange reactor, in which control rods are inserted from the top and bottom of the core. The benchmark includes three different

3-D cases featuring control rods in different configurations, each of which are depicted in Figs. 5.1-5.4.

5.1.1 Methods and Discretization

Results were obtained for all three configurations using 2-D/1-D with NEM and SP_3 axial treatment, as well as a collection of 2-D/3-D methods, and compared to Monte Carlo reference solutions. The following shorthand is used to refer to various methods used:

- 2D1D NEM: 2-D/1-D with nodal expansion method axial treatment
- 2D1D SP3: 2-D/1-D with simplified P_3 axial treatment
- TW CDD-DDFF: 2-D/3-D with two-way coupling, corrected diamond difference correction factors and diamond difference axial treatment with negative flux fix-up
- TW CDD-DD: 2-D/3-D with two-way coupling, corrected diamond difference correction factors and diamond difference axial treatment with no negative flux fix-up
- TW CDD-SC: 2-D/3-D with two-way coupling, corrected diamond difference correction factors and step characteristic axial treatment
- OW DD: 3-D S_N with diamond difference using flux-weighted cross sections obtained from standalone 2-D MoC solutions
- OW CDD-DD: 3-D S_N with corrected diamond difference and uncorrected diamond difference axial treatment using data from standalone 2-D MoC solutions

The one-way 2-D/3-D schemes required three separate 2-D MoC calculations to produce cross sections and correction factors for all three of the rodded configurations. Each of the slabs from the rodded B configuration were needed to form S_N meshes for all of the configurations by composing them in different combinations. The axial reflector region, absent

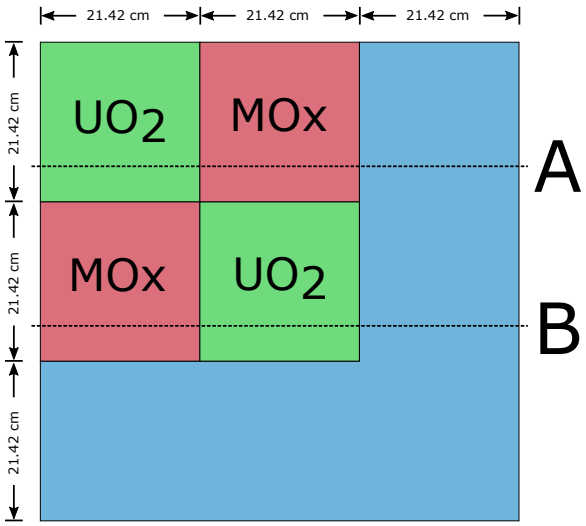


Figure 5.1: C5G7 top assembly view.

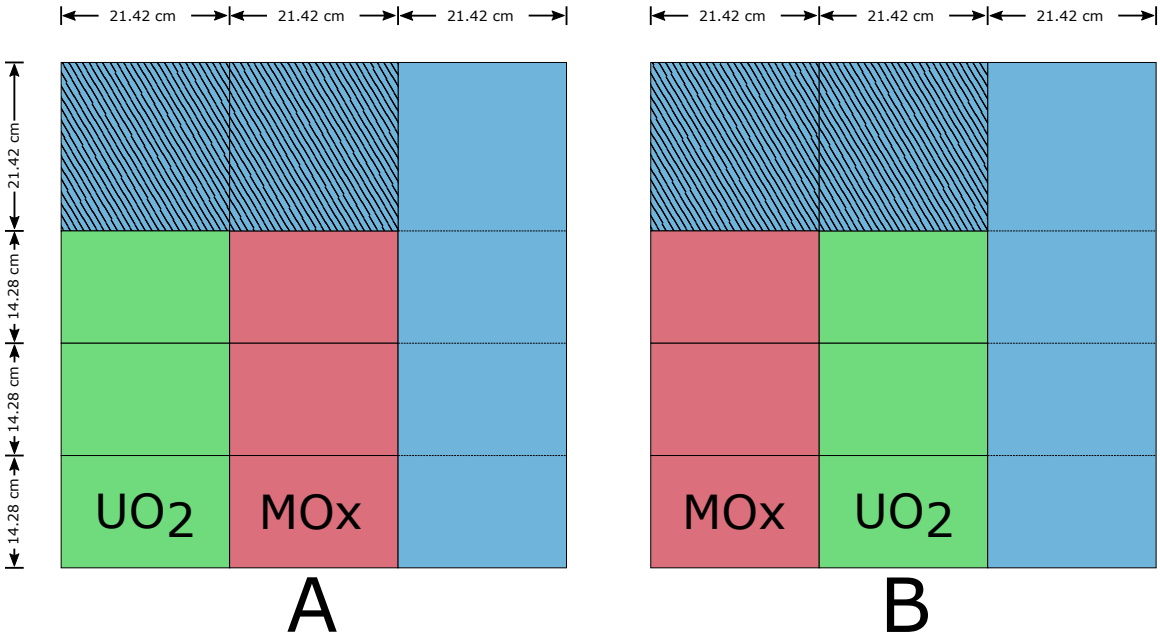


Figure 5.2: C5G7 unrodded case, side view. Hashed regions indicate presence of control rods.

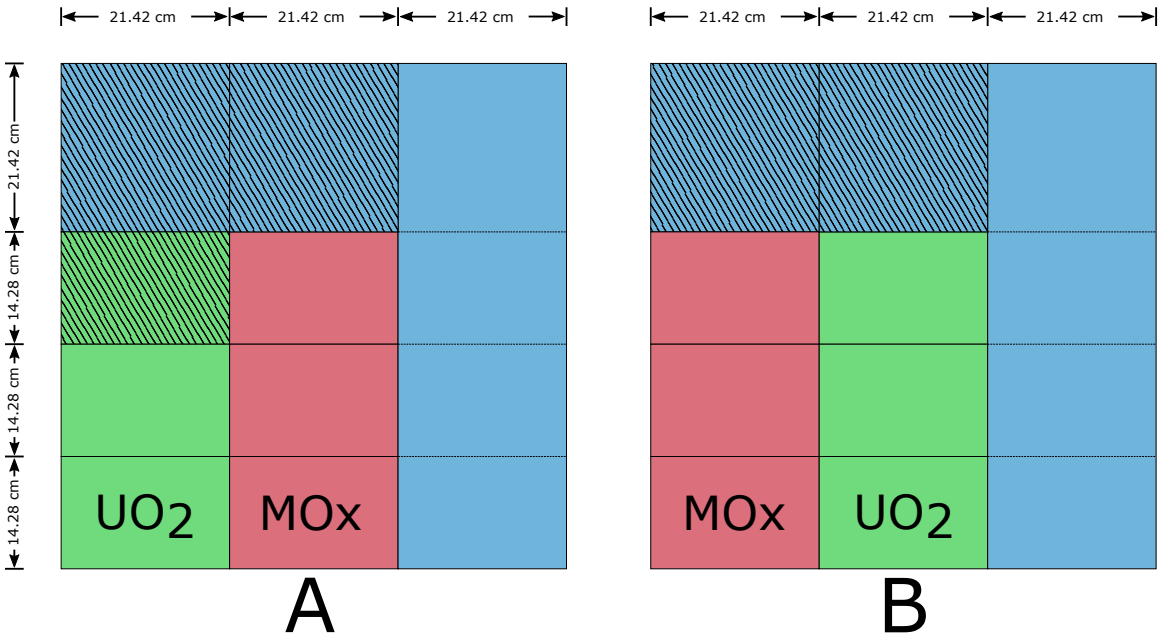


Figure 5.3: C5G7 rodded case A, side view. Hashed regions indicate presence of control rods.

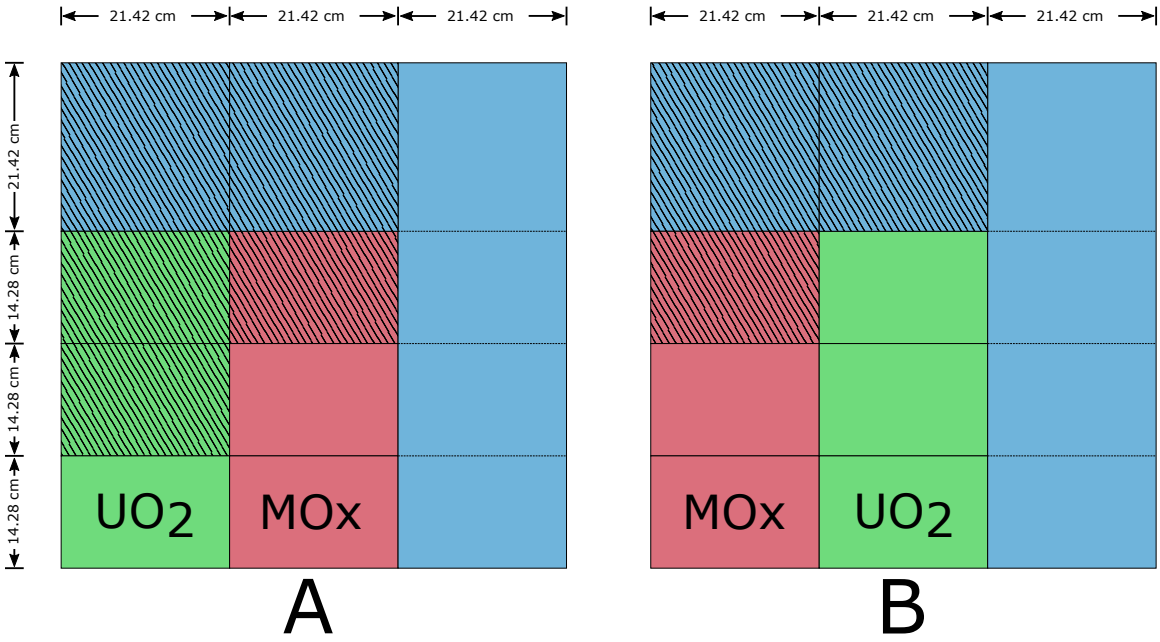


Figure 5.4: C5G7 rodded case B, side view. Hashed regions indicate presence of control rods.

fissile material used volume-weighted cross sections and trivial values for the correction factors.

The MPACT code was used to obtain the 2-D/1-D solutions, while MOCC was used to obtain all of the 2-D/3-D solutions. All calculations used the same spatial mesh for the MoC sweepers, in which the cylindrical pin cells (fuel, guide tube, fission chamber, and control rod) used 5 radial rings in the non-moderator region, two extra radial rings in the surrounding moderator and 8 azimuthal subdivisions. This is the same mesh that is shown in Fig. 2.10. The homogeneous moderator regions in the radial reflector are represented using a rectangular mesh with 5 equally-spaced subdivisions in the x and y directions. The angular quadrature was relatively fine, using a Chebychev quadrature 16 azimuthal angles and a Gauss quadrature with 4 polar angles per octant, for a total of 64 angles per octant. Rays were traced with a nominal spacing of 0.05 cm. The two codes differ slightly in their ray modularization approach, leading to very slight differences in ray placement, but all other aspects of the mesh and angular quadrature were identical. The two methods also employed the same MoC plane height of 3.57 cm. The S_N mesh used for all 2-D/3-D methods applies an axial sub-mesh of 5 regions per MoC plane, giving a uniform axial mesh height of 0.714 cm.

5.1.2 Results

The tables in this section contain estimates of key quantities of interest in reactor analysis: system eigenvalue, 2-D and 3-D assembly powers, and a number of aggregate pin power error metrics were used. Assembly powers are presented integrated over the entire axial dimension of the core, as well as over each slab individually. In all cases, “slab 1” refers to the bottom-most slab, and “slab 3” to top-most. Three aggregate pin power error metrics are also presented: average pin power error, AVG, mean relative error, MRE, and root mean

squared (RMS) error. Each are calculated using the following formulas:

$$\text{AVG} = \frac{\sum_N |e_n|}{N} \tag{5.1}$$

$$\text{RMS} = \sqrt{\frac{\sum_N e_n^2}{N}} \tag{5.2}$$

$$\text{MRE} = \frac{\sum_N |e_n| \cdot p_n}{N \cdot p_{avg}} \tag{5.3}$$

All three configurations required transverse leakage splitting to converge using the two-way 2-D/3-D methods. Rodded configuration B and the unrodded configuration also required the negative flux fix-up in order to converge when using DD axial treatment and CMFD acceleration. Eschewing CMFD acceleration and using unaccelerated power iteration converges without the fix-up, but leads to unacceptably long run times. Nonetheless, these runs were performed at great expense for comparison purposes. The negative flux fix-up was required in these cases, since they exhibit stronger axial discontinuities, which are more difficult to resolve than does the rodded A configuration.

Table 5.1: C5G7 Eigenvalues

	Unrodded		Rodded A		Rodded B	
	Eigenvalue	Error	Eigenvalue	Error	Eigenvalue	Error
Reference	1.14308	±6.0	1.12806	±6.0	1.07777	±6.0
2D1D NEM	1.14220	-88.16	1.12727	-78.90	1.07635	-141.87
2D1D SP3	1.14281	-27.49	1.12791	-15.25	1.07745	-31.68
TW CDD-DDFF	1.14328	20.29	1.12849	43.44	1.07714	-63.34
TW CDD-DD	1.14328	20.13	1.12849	43.46	1.07819	42.34
TW CDD-SC	1.14276	-31.51	1.12764	-41.72	1.07691	-85.97
OW DD	1.14435	126.75	1.12876	70.44	1.07560	-217.07
OW CDD-DD	1.14286	-21.67	1.12831	24.99	1.07814	37.46

Table 5.1 shows the eigenvalue results for all configurations using all methods. As will be a common theme, and is to be expected, OW DD performs poorly at predicting k_{eff} in all configurations, except in rodded configuration A, in which it manages to surpass

2-D/1-D NEM. Somewhat surprisingly, the OW CDD-DD method managed to make good eigenvalue estimates, consistently performing better even than the two-way coupled step characteristic method, though it performed much worse in most other metrics. All two-way coupled 2-D/3-D methods outperformed NEM, but were close to the SP_3. Comparing eigenvalues between the TW CDD-DD and TW CDD-DDFF methods shows the effect of the negative flux fix-up upon the accuracy of the method. In the unrodded and rodded B cases, larger differences can be seen between the eigenvalues than in the rodded A configuration, especially in rodded B. Also, in both cases where the DD/DDFF discrepancy was largest, the DDFF method produced inferior results.

Table 5.2: C5G7 Unrodded 2-D pin power metrics.

	AVG	MRE	RMS	Max Error
2D1D NEM	0.57%	2.79%	0.61%	2.94%
2D1D SP3	0.57%	2.78%	0.61%	2.97%
TW CDD-DDFF	0.38%	1.79%	0.44%	2.44%
TW CDD-DD	0.38%	1.79%	0.44%	2.44%
TW CDD-SC	0.40%	1.75%	0.45%	2.24%
OW DD	1.59%	9.56%	2.08%	6.85%
OW CDD-DD	0.38%	1.80%	0.44%	2.56%

Table 5.3: C5G7 Unrodded slab-wise pin power metrics.

	Slab 1			Slab 2			Slab 3		
	AVG	MRE	RMS	AVG	MRE	RMS	AVG	MRE	RMS
2D1D NEM	0.17%	0.14%	0.20%	0.12%	0.15%	0.16%	0.16%	0.23%	0.21%
2D1D SP3	0.15%	0.15%	0.18%	0.12%	0.13%	0.16%	0.10%	0.14%	0.14%
TW CDD-DDFF	0.12%	0.11%	0.15%	0.10%	0.10%	0.13%	0.12%	0.18%	0.17%
TW CDD-DD	0.12%	0.11%	0.15%	0.10%	0.10%	0.13%	0.12%	0.18%	0.17%
TW CDD-SC	0.17%	0.25%	0.24%	0.12%	0.16%	0.17%	0.15%	0.15%	0.17%
OW DD	0.73%	0.89%	0.97%	0.56%	0.66%	0.72%	0.37%	0.41%	0.47%
OW CDD-DD	0.24%	0.25%	0.26%	0.10%	0.09%	0.13%	0.31%	0.45%	0.39%

Tables 5.2-5.7 contain the 2-D and slab-wise pin power error metrics for each configuration. For most configurations and situations the two-way 2-D/3-D methods outperform the 2-D/1-D methods, but not all. All of the methods that use CDD produced better 2-D pin

Table 5.4: C5G7 Rodded A 2-D pin power metrics.

	AVG	MRE	RMS	Max Error
2D1D NEM	0.57%	3.55%	0.62%	2.66%
2D1D SP3	0.57%	3.63%	0.62%	2.68%
TW CDD-DDFF	0.38%	2.40%	0.45%	2.16%
TW CDD-DD	0.38%	2.40%	0.45%	2.16%
TW CDD-SC	0.41%	2.20%	0.48%	2.15%
OW DD	1.49%	10.77%	1.84%	7.64%
OW CDD-DD	0.38%	2.72%	0.45%	2.09%

Table 5.5: C5G7 Rodded A slab-wise pin power metrics.

	Slab 1			Slab 2			Slab 3		
	AVG	MRE	RMS	AVG	MRE	RMS	AVG	MRE	RMS
2D1D NEM	0.19%	0.21%	0.24%	0.13%	0.15%	0.17%	0.05%	0.05%	0.07%
2D1D SP3	0.18%	0.18%	0.22%	0.11%	0.10%	0.14%	0.12%	0.14%	0.14%
TW CDD-DDFF	0.14%	0.13%	0.17%	0.09%	0.09%	0.12%	0.14%	0.17%	0.18%
TW CDD-DD	0.14%	0.13%	0.17%	0.09%	0.09%	0.12%	0.14%	0.17%	0.18%
TW CDD-SC	0.19%	0.27%	0.26%	0.14%	0.17%	0.18%	0.14%	0.14%	0.15%
OW DD	0.89%	1.13%	1.19%	0.56%	0.63%	0.70%	0.62%	0.72%	0.82%
OW CDD-DD	0.18%	0.19%	0.22%	0.09%	0.09%	0.12%	0.24%	0.28%	0.27%

Table 5.6: C5G7 Rodded B 2-D pin power metrics.

	AVG	MRE	RMS	Max Error
2D1D NEM	0.61%	4.09%	0.72%	3.12%
2D1D SP3	0.57%	4.17%	0.64%	2.76%
TW CDD-DDFF	0.43%	3.10%	0.51%	2.43%
TW CDD-DD	0.38%	2.79%	0.47%	2.29%
TW CDD-SC	0.46%	3.19%	0.56%	2.52%
OW DD	1.64%	13.88%	2.04%	9.52%
OW CDD-DD	0.38%	3.04%	0.47%	2.17%

Table 5.7: C5G7 Rodded B slab-wise pin power metrics.

	Slab 1			Slab 2			Slab 3		
	AVG	MRE	RMS	AVG	MRE	RMS	AVG	MRE	RMS
2D1D NEM	0.39%	0.52%	0.53%	0.12%	0.10%	0.16%	0.08%	0.07%	0.09%
2D1D SP3	0.20%	0.18%	0.26%	0.15%	0.16%	0.18%	0.07%	0.08%	0.09%
TW CDD-DDFF	0.51%	0.57%	0.55%	0.24%	0.27%	0.33%	0.37%	0.42%	0.44%
TW CDD-DD	0.19%	0.19%	0.23%	0.15%	0.17%	0.19%	0.13%	0.15%	0.16%
TW CDD-SC	0.29%	0.40%	0.40%	0.12%	0.12%	0.15%	0.16%	0.15%	0.17%
OW DD	1.16%	1.32%	1.48%	0.93%	1.01%	1.23%	0.66%	0.73%	0.86%
OW CDD-DD	0.23%	0.25%	0.27%	0.12%	0.12%	0.15%	0.21%	0.24%	0.24%

power error metrics than both 2-D/1-D methods for all three configurations. As expected, the uncorrected DD method with flux weighted cross sections performed quite poorly, with some 2-D metrics exceeding 10%. Examining the slab-wise metrics uncovers more mixed results.

For the unrodded configuration, the CDD-DD and CDD-DDFF methods produce slightly better answers in the bottom two slabs, but slightly worse in the top slab compared to SP_3. That being said, they are close enough to be more or less indistinguishable, especially considering that they are all falling within the statistical uncertainties of the reference Monte Carlo solution. The TW CDD-SC method appears to perform adequately in this case, and the OW CDD-DD is clearly suffering, especially in the top slab near the axial reflector. Even though it was clear from the eigenvalue results that the negative flux fix-up was affecting the solutions, these effects seem minor in the context of the pin power errors for the unrodded configuration.

Similarly to the unrodded configuration, the two-way 2-D/3-D methods perform better than the 2-D/1-D methods, except in the upper slab, where both 2-D/1-D methods performs better. The step characteristic-based TW CDD-SC method performs comparatively worse in the rodded A configuration than in the unrodded configuration, with errors in the bottom and middle slabs twice that of the DD-based 2-D/3-D methods. The OW CDD-DD method even manages to outperform the SC-based method in all but the top-most slab. Again, the effect of the negative flux fix-up is not noticeable in this case.

Examining the rodded B configuration on the other hand exhibits large differences in the pin power error between the TW CDD-DD and TW CDD-DDFF methods, with the negative flux fix-up producing much larger pin power errors than without. The rodded configuration B appears to be the only case where the negative flux fix-up has a significantly detrimental effect, which is also present in the assembly power estimates discussed below. The rest of the methods show similar behavior to the previous configurations, with the 2-D/3-D methods doing better than 2-D/1-D towards the bottom of the core but worse at the top.

Table 5.8: C5G7 Unrodded 2-D assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	491.20	±0.29%	212.70	±0.21%	139.40	±0.15%
2D1D NEM	489.74	-0.30%	213.13	0.20%	140.00	0.44%
2D1D SP3	489.75	-0.30%	213.12	0.20%	140.01	0.45%
TW CDD-DDFF	490.06	-0.23%	213.05	0.16%	139.84	0.33%
TW CDD-DD	490.06	-0.23%	213.05	0.16%	139.84	0.33%
TW CDD-SC	489.78	-0.29%	213.21	0.24%	139.80	0.29%
OW DD	493.53	0.47%	211.60	-0.52%	139.27	-0.09%
OW CDD-DD	490.09	-0.23%	213.03	0.15%	139.86	0.34%

Table 5.9: C5G7 Unrodded Slab 1 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	219.00	±0.19%	94.50	±0.14%	62.10	±0.10%
2D1D NEM	219.05	0.01%	95.04	0.54%	62.64	0.84%
2D1D SP3	218.63	-0.19%	94.85	0.34%	62.52	0.64%
TW CDD-DDFF	218.95	-0.04%	94.85	0.34%	62.46	0.55%
TW CDD-DD	218.94	-0.04%	94.85	0.34%	62.46	0.55%
TW CDD-SC	217.99	-0.48%	94.47	-0.06%	62.19	0.11%
OW DD	221.22	1.00%	94.51	-0.03%	62.39	0.45%
OW CDD-DD	219.71	0.31%	95.17	0.68%	62.67	0.90%

Tables 5.8-5.19 contain assembly power estimates for all three C5G7 configurations, in 2-D and for each axial slab individually. Overall, much of the same behavior from the aggregate pin power error metrics is found in the assembly power predictions; OW DD

Table 5.10: C5G7 Unrodded Slab 2 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	174.20	±0.17%	75.20	±0.13%	49.50	±0.09%
2D1D NEM	173.72	-0.30%	75.40	0.20%	49.69	0.47%
2D1D SP3	173.79	-0.26%	75.43	0.24%	49.71	0.52%
TW CDD-DDFF	173.93	-0.18%	75.40	0.20%	49.65	0.40%
TW CDD-DD	173.93	-0.18%	75.40	0.20%	49.65	0.40%
TW CDD-SC	173.58	-0.37%	75.29	0.06%	49.56	0.22%
OW DD	175.42	0.68%	74.99	-0.34%	49.51	0.12%
OW CDD-DD	174.20	-0.02%	75.51	0.35%	49.72	0.55%

Table 5.11: C5G7 Unrodded Slab 3 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	97.90	±0.13%	42.90	±0.10%	27.80	±0.07%
2D1D NEM	96.97	-0.98%	42.69	-0.54%	27.68	-0.51%
2D1D SP3	97.33	-0.61%	42.83	-0.20%	27.79	-0.12%
TW CDD-DDFF	97.18	-0.77%	42.80	-0.28%	27.74	-0.29%
TW CDD-DD	97.18	-0.76%	42.80	-0.28%	27.74	-0.29%
TW CDD-SC	98.21	0.28%	43.44	1.22%	28.05	0.83%
OW DD	96.90	-1.06%	42.10	-1.91%	27.36	-1.64%
OW CDD-DD	96.18	-1.79%	42.35	-1.34%	27.46	-1.29%

Table 5.12: C5G7 Rodded A 2-D assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	461.20	±0.28%	221.70	±0.22%	151.40	±0.16%
2D1D NEM	459.77	-0.31%	222.13	0.19%	151.96	0.37%
2D1D SP3	459.87	-0.28%	222.08	0.17%	151.96	0.37%
TW CDD-DDFF	460.23	-0.21%	222.01	0.13%	151.76	0.24%
TW CDD-DD	460.23	-0.21%	222.01	0.13%	151.76	0.24%
TW CDD-SC	459.54	-0.36%	222.28	0.26%	151.90	0.33%
OW DD	461.96	0.17%	221.09	-0.28%	151.85	0.30%
OW CDD-DD	460.66	-0.11%	221.87	0.07%	151.60	0.13%

Table 5.13: C5G7 Rodded A Slab 1 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	237.40	±0.20%	104.50	±0.15%	69.80	±0.11%
2D1D NEM	236.67	-0.31%	104.83	0.34%	70.23	0.62%
2D1D SP3	236.93	-0.20%	104.90	0.40%	70.25	0.64%
TW CDD-DDFF	237.43	0.01%	104.85	0.36%	70.12	0.46%
TW CDD-DD	237.43	0.01%	104.85	0.36%	70.12	0.46%
TW CDD-SC	236.25	-0.49%	104.47	-0.01%	69.88	0.11%
OW DD	240.86	1.45%	105.00	0.49%	70.47	0.96%
OW CDD-DD	237.83	0.18%	105.00	0.49%	70.21	0.59%

Table 5.14: C5G7 Rodded A Slab 2 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	167.50	±0.17%	78.00	±0.13%	53.40	±0.09%
2D1D NEM	166.93	-0.35%	78.16	0.20%	53.62	0.43%
2D1D SP3	167.25	-0.16%	78.19	0.24%	53.63	0.45%
TW CDD-DDFF	167.35	-0.10%	78.14	0.17%	53.56	0.32%
TW CDD-DD	167.34	-0.10%	78.14	0.17%	53.56	0.32%
TW CDD-SC	166.82	-0.42%	78.10	0.12%	53.55	0.30%
OW DD	168.39	0.52%	77.86	-0.19%	53.64	0.47%
OW CDD-DD	167.61	0.06%	78.20	0.24%	53.57	0.35%

Table 5.15: C5G7 Rodded A Slab 3 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	56.30	±0.09%	39.20	±0.09%	28.20	±0.07%
2D1D NEM	56.17	-0.15%	39.14	-0.22%	28.11	-0.35%
2D1D SP3	55.69	-1.00%	38.99	-0.60%	28.08	-0.45%
TW CDD-DDFF	55.45	-1.43%	39.01	-0.54%	28.08	-0.44%
TW CDD-DD	55.45	-1.43%	39.01	-0.54%	28.08	-0.44%
TW CDD-SC	56.47	0.38%	39.71	1.24%	28.47	0.95%
OW DD	52.71	-6.31%	38.24	-2.52%	27.75	-1.63%
OW CDD-DD	55.21	-1.85%	38.68	-1.40%	27.81	-1.40%

Table 5.16: C5G7 Rodded B 2-D assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	395.40	±0.26%	236.60	±0.23%	187.30	±0.18%
2D1D NEM	393.17	-0.57%	237.22	0.25%	188.40	0.56%
2D1D SP3	394.08	-0.34%	236.99	0.16%	187.94	0.32%
TW CDD-DDFF	394.01	-0.36%	236.98	0.15%	188.04	0.37%
TW CDD-DD	394.53	-0.23%	236.88	0.11%	187.71	0.20%
TW CDD-SC	393.54	-0.48%	237.16	0.23%	188.15	0.43%
OW DD	390.43	-1.26%	237.43	0.34%	190.71	1.80%
OW CDD-DD	394.86	-0.14%	236.83	0.09%	187.48	0.07%

Table 5.17: C5G7 Rodded B Slab 1 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	247.70	±0.21%	125.80	±0.17%	91.60	±0.13%
2D1D NEM	245.39	-0.95%	92.14	0.54%	125.82	0.02%
2D1D SP3	247.38	-0.15%	92.23	0.64%	126.27	0.39%
TW CDD-DDFF	249.63	0.76%	92.66	1.12%	127.02	0.98%
TW CDD-DD	248.29	0.22%	92.04	0.44%	126.24	0.36%
TW CDD-SC	245.96	-0.72%	91.76	0.13%	125.66	-0.10%
OW DD	251.62	1.56%	93.98	2.55%	127.57	1.42%
OW CDD-DD	248.57	0.33%	92.08	0.48%	126.30	0.41%

Table 5.18: C5G7 Rodded B Slab 2 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	106.60	±0.13%	81.40	±0.14%	65.00	±0.11%
2D1D NEM	106.60	0.04%	81.69	0.35%	65.43	0.63%
2D1D SP3	105.90	-0.62%	81.39	-0.02%	65.18	0.24%
TW CDD-DDFF	105.04	-1.42%	81.35	-0.07%	65.22	0.30%
TW CDD-DD	105.77	-0.74%	81.45	0.05%	65.16	0.22%
TW CDD-SC	106.18	-0.35%	81.57	0.20%	65.31	0.45%
OW DD	101.55	-4.70%	81.53	0.16%	66.20	1.81%
OW CDD-DD	106.12	-0.42%	81.52	0.13%	65.16	0.22%

Table 5.19: C5G7 Rodded B Slab 3 assembly powers.

	Inner UO_2	Error	MOX	Error	Outer UO_2	Error
Reference	41.10	±0.08%	29.40	±0.08%	30.70	±0.07%
2D1D NEM	41.18	0.14%	29.71	0.97%	30.83	0.49%
2D1D SP3	40.80	-0.79%	29.33	-0.33%	30.53	-0.48%
TW CDD-DDFF	39.33	-4.35%	28.60	-2.79%	30.16	-1.70%
TW CDD-DD	40.47	-1.59%	29.20	-0.76%	30.50	-0.58%
TW CDD-SC	41.39	0.66%	29.93	1.71%	31.08	1.30%
OW DD	37.26	-9.41%	28.33	-3.71%	30.54	-0.46%
OW CDD-DD	40.18	-2.30%	29.01	-1.41%	30.24	-1.43%

performs poorly throughout, OW CDD-DD performs admirably for the axially integrated powers and slab 1 and slab 2, but fails miserably in slab 3. The negative flux fix-up shows no strong effects on the assembly powers for the unrodded and rodded A configurations, but in the rodded B configuration the difference in error is dramatic. In the unrodded and rodded A configurations, the assembly power results from the TW CDD-DD and TW CDD-DDFF methods outperform or are on par with the 2-D/1-D methods. Even in the rodded B configuration, the 2-D/3-D methods perform comparatively well, except in the top-most slab.

5.1.3 Convergence behavior

Convergence behavior was compared between the 2-D/3-D and 2-D/1-D methods. Figures 5.5-5.7 present the values of the fission source convergence criterion at each iteration for each configuration. Only the TW CDD-DDFF and TW CDD-SC 2-D/3-D methods are shown, as the TW CDD-DD method does not reliably converge with CMFD acceleration enabled, requiring around 100 iterations to converge. Since MOCC and MPACT calculate their convergence criteria on a different mesh basis, the MOCC results were scaled to match MPACT for a fair comparison.

In general, the TW CDD-DDFF converges at a similar overall rate as both 2-D/1-D methods. For much of the convergence, the TW CDD-DDFF method performed worse

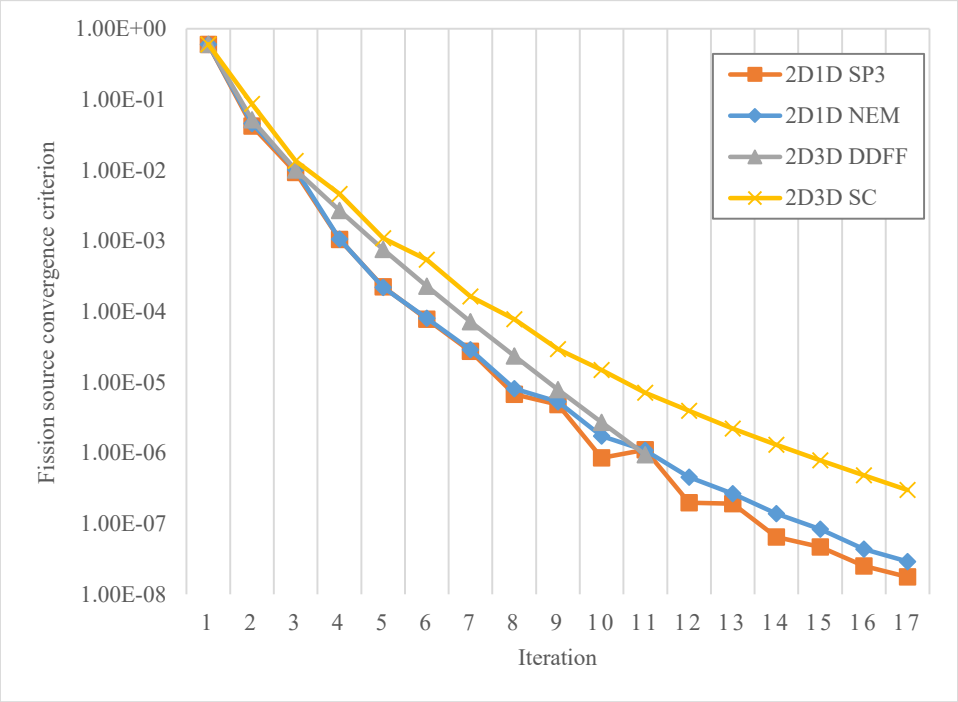


Figure 5.5: Fission source convergence for the C5G7 unrodded configuration.

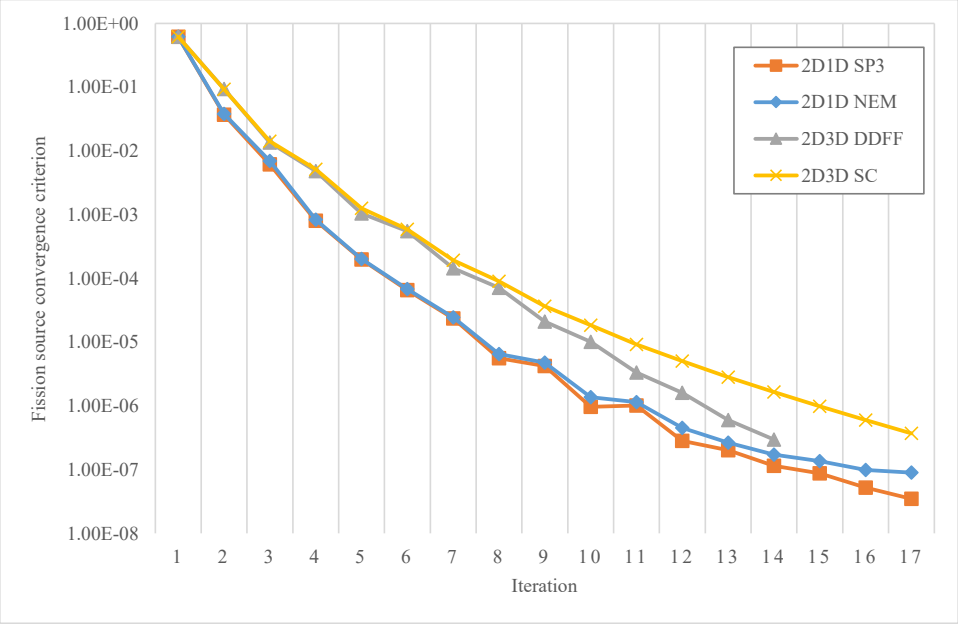


Figure 5.6: Fission source convergence for the C5G7 rodded A configuration.

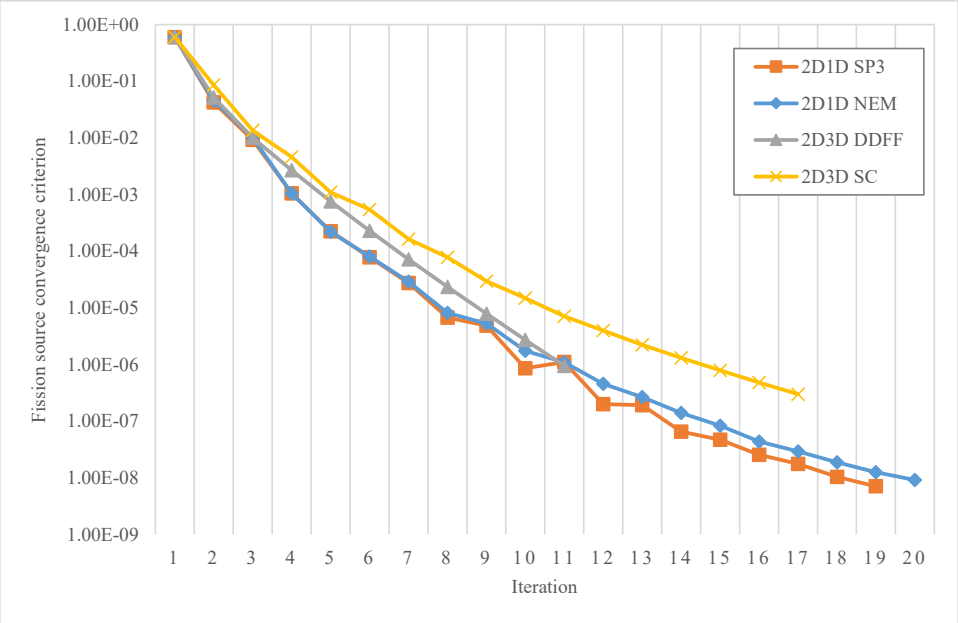


Figure 5.7: Fission source convergence for the C5G7 rodded B configuration.

than the 2-D/1-D methods, though this may be the result of insufficient S_N inner iterations. These are discussed more in Section 5.2. The TW CDD-SC method converged noticeably slower in all cases.

5.1.4 Effect of Transverse Leakage on Correction Factors

The purpose of implementing the two-way transverse leakage-based 2-D/3-D method is twofold:

- to provide better correction factors and flux-weighted cross sections to the S_N sweeper, and
- to improve the validity of the fine-mesh flux as it would be reconstructed using the S_N flux.

The second aspect is difficult to demonstrate without a fine-mesh reference solution. However, effects of the transverse leakage on the correction factors can be observed. In this section, correction factors are compared between those obtained using a two-way method

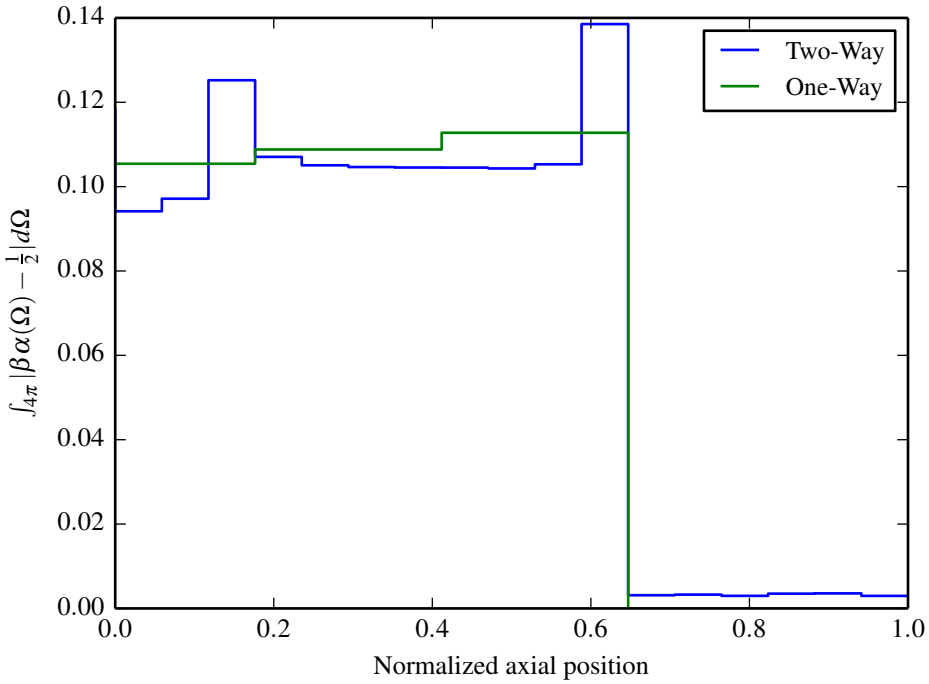


Figure 5.8: Axial dependence of $\beta\alpha_x$ for pin at location (1,1).

to their one-way counterparts. The rodded B configuration was used for all of these comparisons because it exhibits the most interesting axial behavior.

Because the correction factors exhibit variations in space, energy, and angle, they can be difficult to visualize. To aid in their comparison, an angular integral of the correction factor deviations from their trivial value,

$$\overline{\beta\alpha} = \int_{4\pi} \left| \beta\alpha_x - \frac{1}{2} \right| d\Omega, \tag{5.4}$$

is used. Due to the symmetry of the problem, it does not matter which correction factor ($\beta\alpha_x$ or $\beta\alpha_y$) is used to obtain the metric defined in Eq. (5.4). Several axial line-outs were obtained from the converged state of the rodded B configuration using the TW CDD-DDFF method and compared to the one-way correction factors.

Figures 5.8-5.10 contain these axial line-outs for specific pin cells for the highest energy group. The first is taken from the center-most UO_2 pin cell, which is representative of

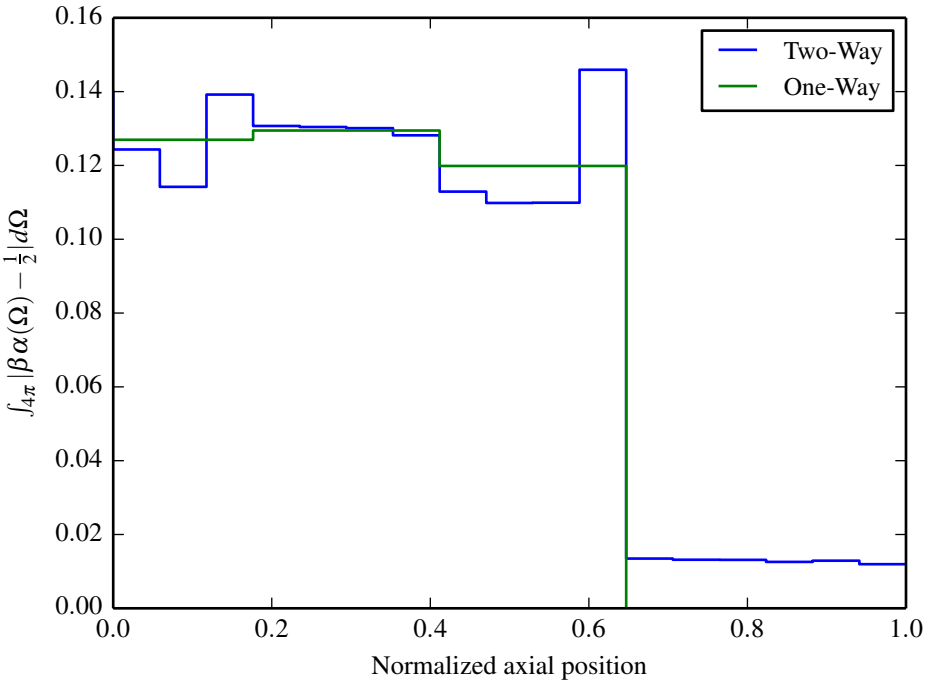


Figure 5.9: Axial dependence of $\beta\alpha_x$ for pin at location (27,9), in one of the MOx assemblies.

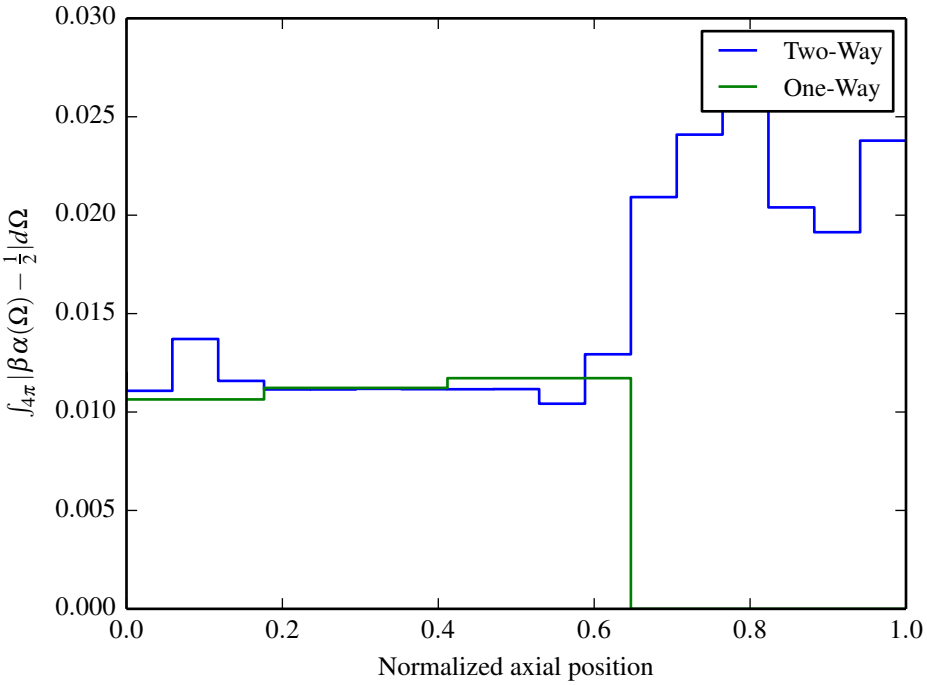


Figure 5.10: Axial dependence of $\beta\alpha_x$ for pin at location (39,35), in the radial reflector.

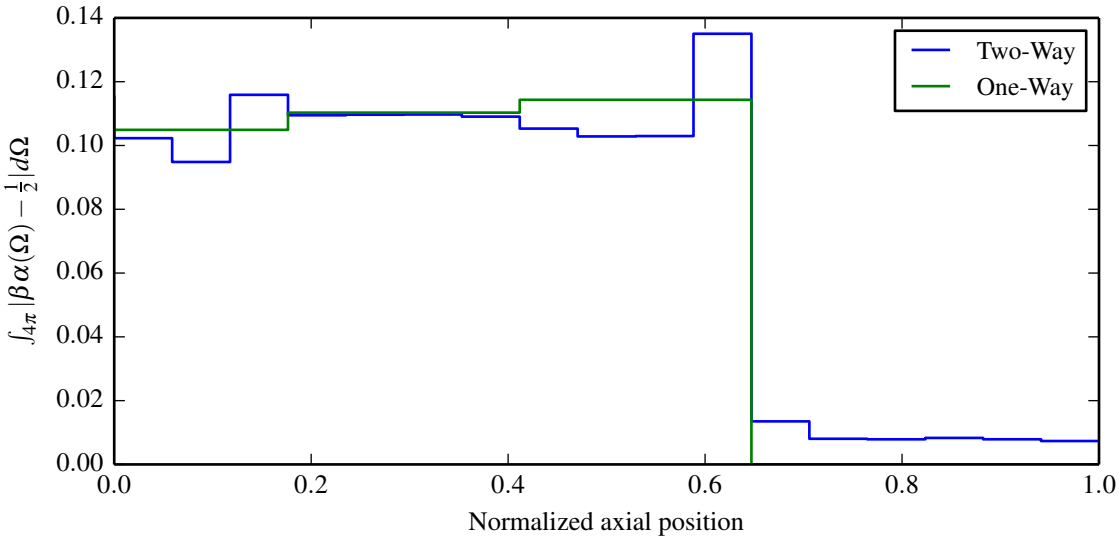


Figure 5.11: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 1.

most of the pins in the central assembly. Large differences between the one- and two-way corrections are clear, especially in the planes immediately below the tips of the control rods and the axial reflector interface. This behavior is expected, as transverse leakage is greatest in magnitude in these regions. The second (Fig. 5.9) is a fuel pin located in one of the MOx assemblies. Even in the absence of control rod tips at the interface between the bottom and middle slabs, variation is found in the corrections, likely due to the presence of control rods in the neighboring UO_2 assembly. Located in the radial reflector near the active core region, the third cell (Fig. 5.10) highlights the significance of the correction factors in the axial reflector region, which the one-way method lacks entirely. This discrepancy is responsible for much of the error found in the OW CDD-DD method in the top slab.

Figures 5.11-5.17 show comparisons for the same pin cell for all energy groups. The pin cell shown is the fission chamber in the center of one of the MOx assemblies. Due to significant streaming behavior and more anisotropic flux at high energies, the two-way correction factors exhibit the most axial variation in these groups. At lower energies, less

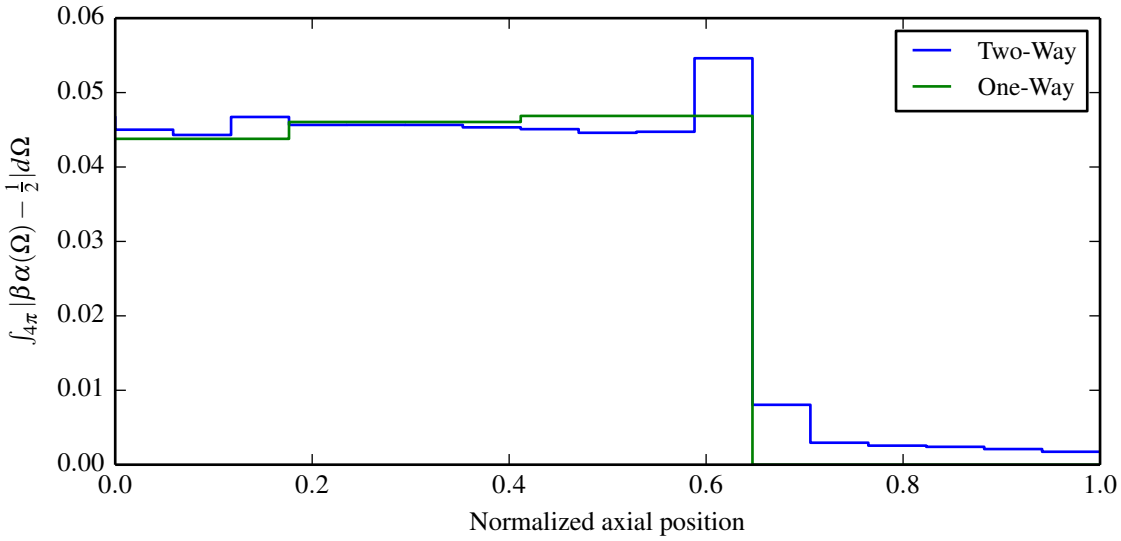


Figure 5.12: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 2.

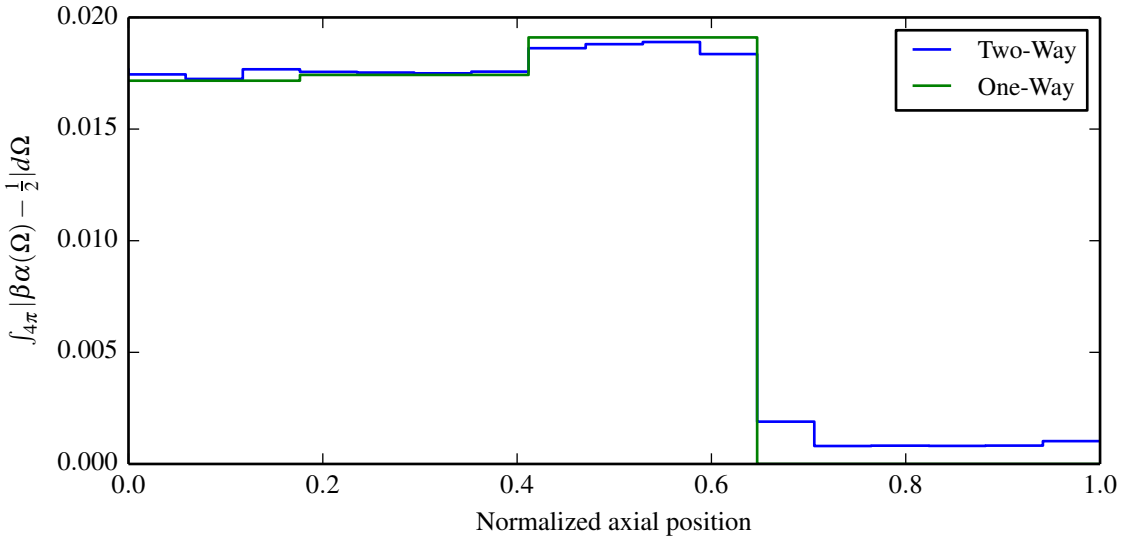


Figure 5.13: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 3.

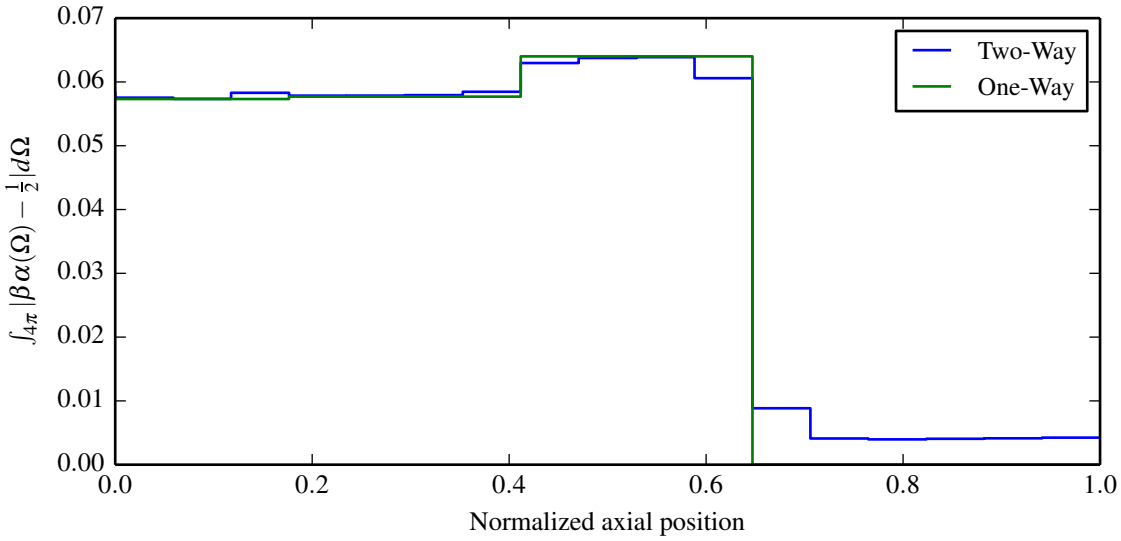


Figure 5.14: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 4.

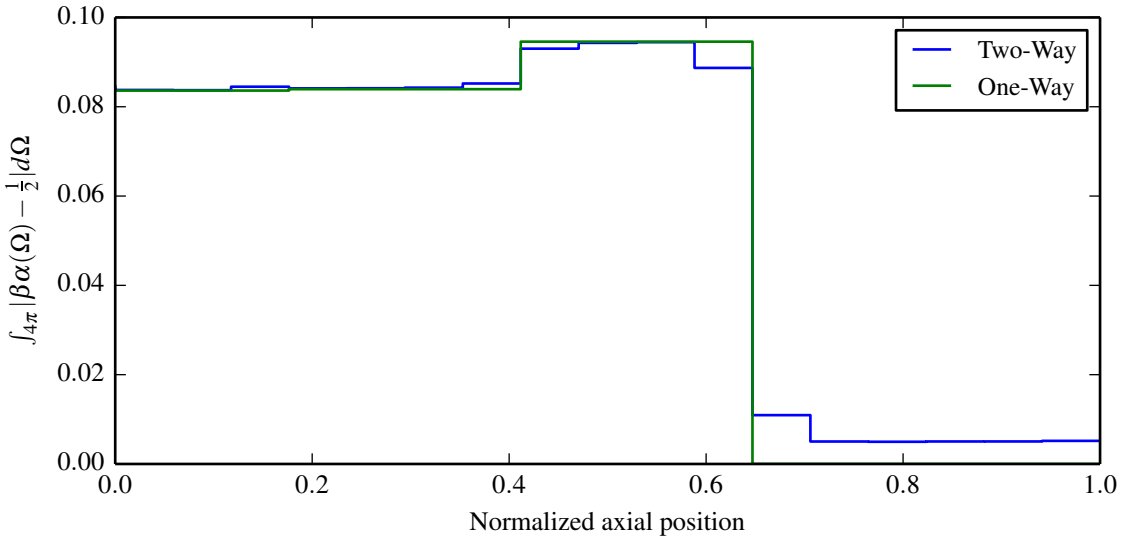


Figure 5.15: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 5.

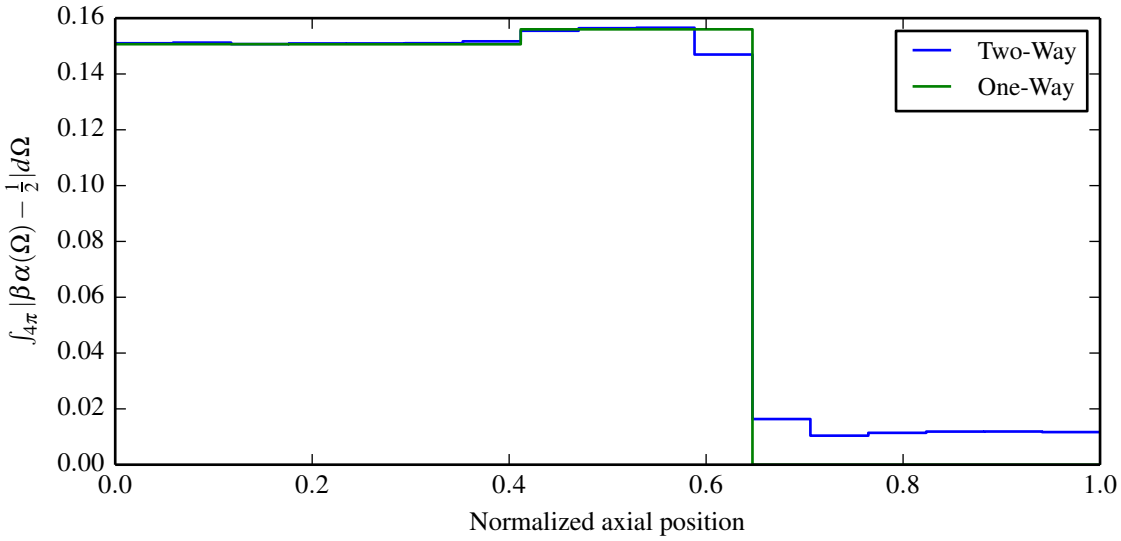


Figure 5.16: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 6.

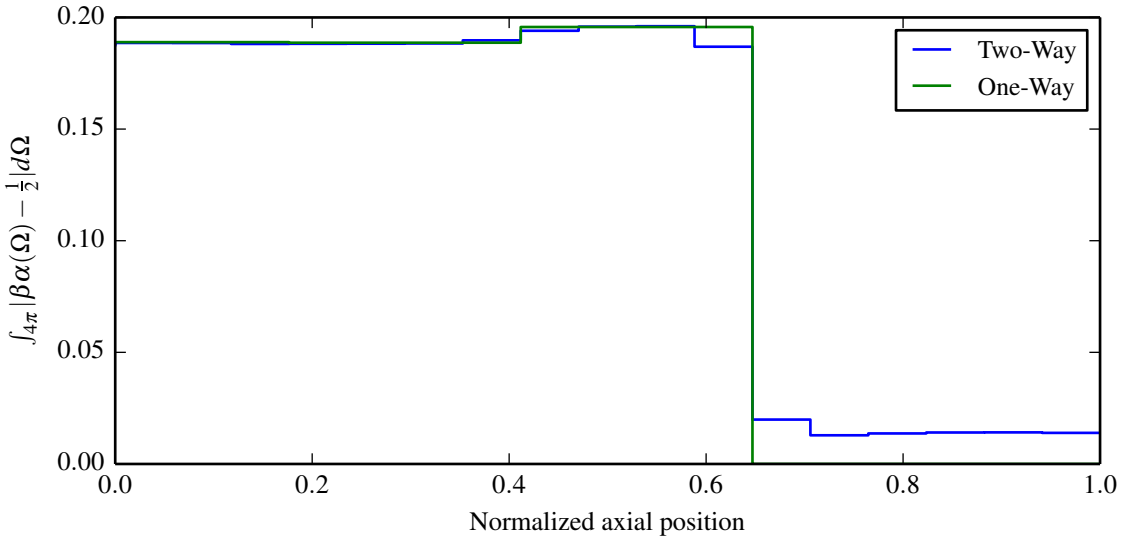


Figure 5.17: Axial dependence of $\beta\alpha_x$ for pin at location (8,25), the MOx fission chamber, for group 7.

axial variation is found. While small, there are still corrections applied in the axial reflector region which are missed in the one-way methods.

Cross section differences, while present, tended to be very slight between the one-way and two-way methods.

5.1.5 2-D/3-D Memory Footprint and Timing

In the previous section, it was demonstrated that the 2-D/3-D method was capable of producing good predictions of system eigenvalue and power distribution for all of the C5G7 rodded configurations. A large source of concern, however, is the amount of memory required to use the 2-D/3-D method with CDD; two correction factors must be stored for each pin, for every angle and energy group. At first blush this sounds prohibitive, but for the cases shown they tended to be quite manageable due to the coarseness of the mesh.

For the one-way approach, the memory requirements depended upon the number of 2-D MoC planes that are needed to capture the nature of the problem. The unrodded configuration needed just one (all assemblies unrodded), while the rodded A and B configurations required two and three, respectively, to handle the slabs with the rodded central UO_2 assembly and rodded MOx assemblies. Each plane required about 71 megabytes of memory to store all of the correction factors, as demonstrated in Table 5.20. Furthermore, the S_N sweeper required unique cross sections for each cell. Since the cross sections lack angular dependence, they only require several megabytes per plane. Since all C5G7 configurations are symmetric about the 45° axis, all of these requirements could be cut roughly in half. This optimization does not extend to general cases and is therefore not used. The total memory requirement for storing the correction factors and cross sections for the one-way method ended up being around 214 megabytes.

The two-way coupling approach required the storage of considerably more correction factors, since more MoC planes were used. Rather than the three planes used in the one-way method, the two-way cases used 18 planes, resulting in a memory requirement of

Table 5.20: C5G7 plane memory storage requirements for CDD correction factors.

Number of angles	256	
Number of pin cells	2,601	
Number of groups	7	
Number of CDD factors	2	
Bytes per factor	8	
Memory per plane	74,575,872	Bytes
	71.12	MB

1,280 megabytes. This constitutes a much larger memory burden, though it is by no means prohibitive. Extrapolating to a real quarter-core, 3-D reactor simulation such as AMA Problem 5 [12] with realistic cross sections would require about 86 gigabytes of memory. Table 5.21 shows the values used to arrive at this estimate. Under most reasonable spatial decompositions for such a problem, on the order of a thousand processors [23], the resultant memory per node becomes quite tractable at tens of megabytes per processor.

Table 5.21: Quarter-core memory storage requirements for CDD correction factors.

Number of assemblies	49	
Number of pin cells	14,200	
Number of angles	144	
Number of planes	60	
Number of groups	47	
Number of CDD factors	2	
Bytes per factor	8	
	85.9	GB

Comparing the time requirements between the 2-D/3-D and 2-D/1-D methods using MOCC and MPACT is difficult. MPACT is a mature, well-profiled code, upon which many person-hours have been devoted to optimizing performance. MOCC, on the other hand is considerably newer, and much less effort has been spent on this type of optimization. This is not to imply that comparable performance could not be achieved with sufficient effort. Furthermore, the primary approach to parallelism in MPACT is MPI-based distributed memory, while MOCC relies on a shared-memory OpenMP threading approach. While

MPACT does implement a thread-based parallel model, it is poorly-supported at the time of this writing. As a result, direct comparisons of run times between the two codes, while fair, would not constitute an effective comparison of the underlying methods themselves. The MoC sweep algorithm is the same for both methods, and calculation of the CDD correction factors imposes no appreciable overhead beyond what is otherwise required to calculate currents for CMFD or radial transverse leakage for a 2-D/1-D method.

A more effective comparison lies in the proportional time required to perform the axial solve (in the case of 2-D/1-D) or the S_N sweeps (in the case of 2-D/3-D) to the MoC sweep time. With the settings used in the above cases, it was found that the S_N sweeps in the 2-D/3-D methods required about 25-27% of the time spent in the MoC sweeps. In contrast, the SP_3 2-D/1-D method required about 12% of the MoC to perform the nodal solution. NEM required a meager 2% of MoC to perform the axial solve; being a diffusion-based solver, its relative cheapness is to be expected. The time required to perform the S_N sweeps is highly dependent upon the particular S_N implementation used, as well as the number of inner iterations performed. In the above cases, 10 S_N iterations were used per outer iteration, which was selected because it appeared to perform well. The optimal number of S_N inner iterations is problem-dependent, affected by physical characteristics such as the scattering ratio. Future work will investigate more sophisticated methods of determining the optimal number of S_N sweeps to perform, or whether more advanced iterative techniques could reduce the number of S_N iterations needed to achieve good performance. Either way, many possible improvements are possible to the S_N sweeper, and it is likely within easy striking distance of SP_3, performance-wise.

5.2 Inner iteration convergence behavior

To examine the convergence behavior of the 2-D/3-D method in more detail, a smaller test case was used. This case was very similar to the C5G7 rodded A configuration, with a

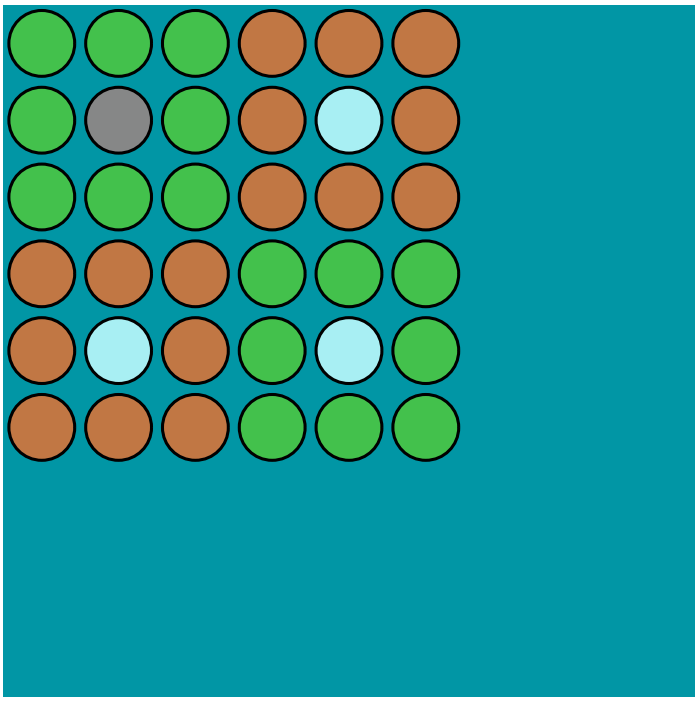


Figure 5.18: Small C5G7 core layout. Axial mesh is identical to C5G7 rodded configuration cd A.

much smaller radial geometry as depicted in Fig. 5.18. The middle 3-by-3 rod assembly contains a control rod inserted one third of the way into the active core regions, as in rodded A, and the peripheral 3-by-3 assemblies contain control rod guide tubes with their rods extracted into the axial reflector region. This case was run using the TW CDD-DD method for various combinations of S_N and MoC inner iterations, and the resultant convergence behavior is plotted against iteration index in Figs. 5.19 and 5.21 and against outer iteration index in Figs. 5.20 and 5.22.

Varying the number of MoC iterations uncovers some very interesting behavior. It was found that increasing from one inner MoC iteration per outer iteration to three resulted in improved convergence rate, but increasing further had no appreciable benefit. This is best illustrated by Fig. 5.20. As shown in Fig. 5.19, the improved iteration-for-iteration convergence rate of three MoC translated to an improved wall time convergence, though increasing from there ended up costing more time.

Similar behavior was found for the S_N iterations. Exceeding 20 inner iterations per

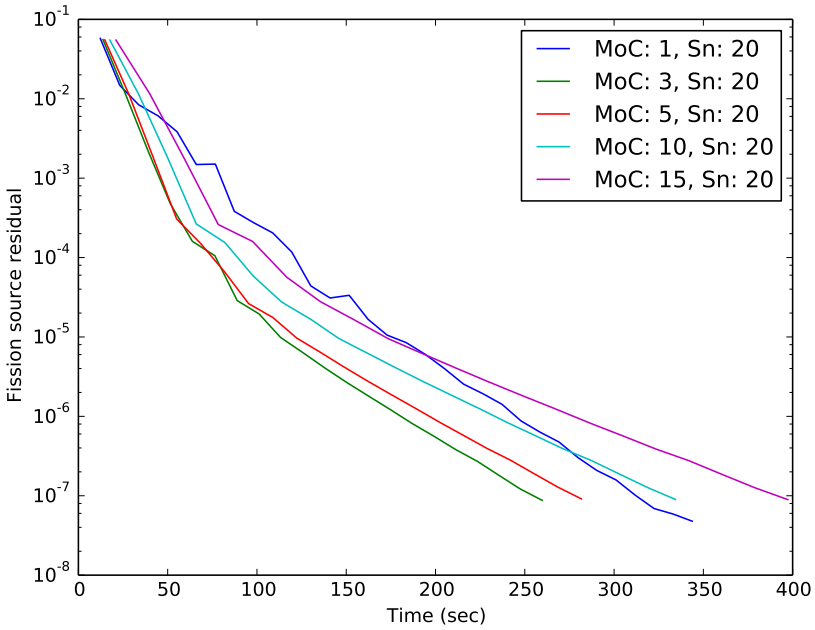


Figure 5.19: Convergence behavior with varying MoC inner iterations for small 3-D case, by time.

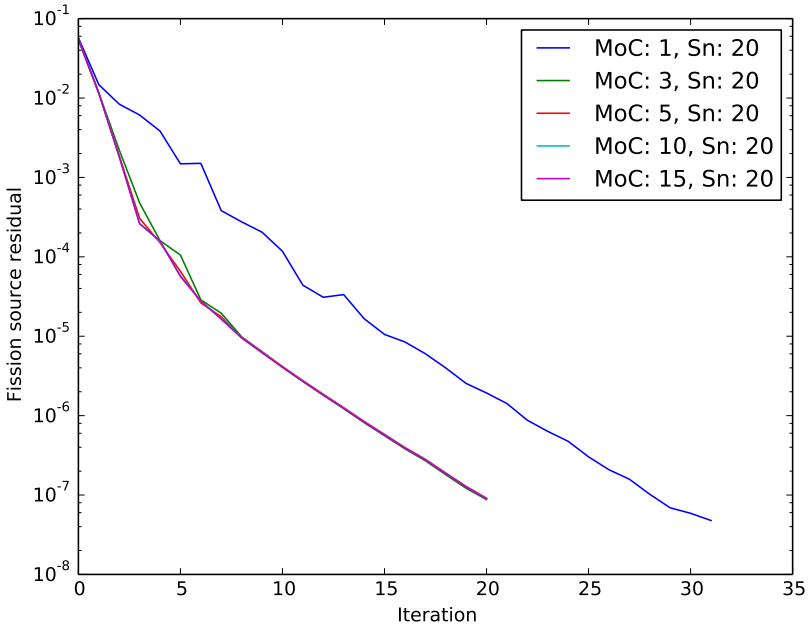


Figure 5.20: Convergence behavior with varying MoC inner iterations for small 3-D case, by iteration.

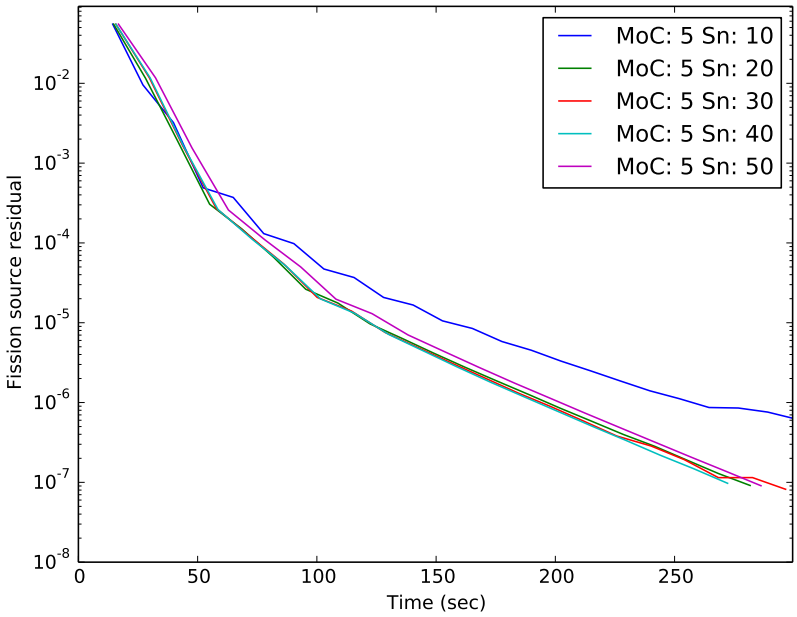


Figure 5.21: Convergence behavior with varying S_N inner iterations for small 3-D case, by time.

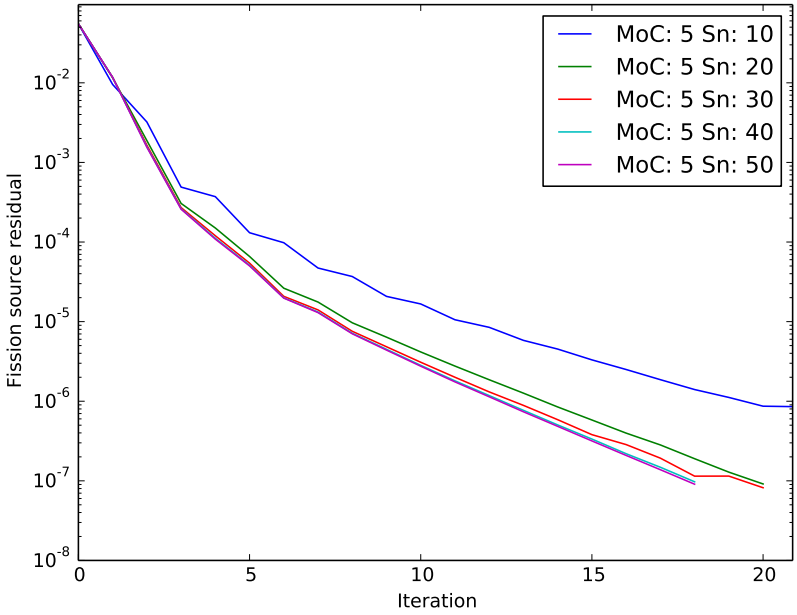


Figure 5.22: Convergence behavior with varying S_N inner iterations for small 3-D case, by iteration.

outer iteration posed little benefit, though more than the MoC inner iterations. Comparing Figs. 5.20 and 5.22 implies that convergence is limited sooner by the S_N sweeper, providing further incentive to emphasize S_N performance.

CHAPTER 6

Conclusions and Future Work

6.1 Brief Summary

This work developed a class of techniques for solving 3-D reactor neutronics problems, which joins a coarse-mesh, 3-D S_N sweeper with a collection of fine-mesh, 2-D Method of Characteristics (MoC) sweepers. We called these 2-D/3-D methods, after the 2-D/1-D methods that inspired and preceded them. A major motivation for the development of such an approach stemmed from the impressive computational performance of orthogonal-mesh S_N, as well as its suitability for modern high-performance computing architectures. Another motivation was that using a fully 3-D transport solver to obtain transverse leakages for the MoC sweepers would prove more accurate than the 1-D nodal diffusion methods typically used with 2-D/1-D methods.

The development of the 2-D/3-D scheme began by introducing the neutron transport equation along with various discretization and solution methods of import, namely the S_N and MoC methods. The concept of S_N auxiliary equations was discussed, and it was demonstrated that existing S_N schemes are too impractical to be used for pin-resolved transport on an orthogonal mesh. A new differencing scheme to the S_N equations, Corrected Diamond Difference (CDD), was proposed which maintains accuracy on a very coarse grid by using information from an underlying fine-mesh solver. This was performed by extracting correction factors from the fine-mesh solver which preserve streaming and col-

lision behavior. It was then demonstrated that the CDD equations are capable of yielding equivalent coarse-mesh solutions to the fine-mesh solution that generated the corrections. The CDD scheme was then extended to 3-D using the Diamond Difference (DD) and Step Characteristics (SC) differencing schemes to treat the axial dimension. A one-way and a transverse leakage-based two-way coupling scheme were described for applying CDD to 3-D problems.

These methods were implemented in a new MoC-based transport code, MOCC, and applied to the C5G7 benchmark problems, as well as some other test cases to study their efficacy. It was found that the 2-D/3-D methods (especially the two-way methods) were capable of making very good predictions of system eigenvalue and pin power distributions, especially compared to the nodal diffusion-based 2-D/1-D Nodal Expansion Method (NEM) method. In most cases 2-D/3-D outperformed the transport-based 2-D/1-D SP_3 method as well from an accuracy standpoint.

6.2 Conclusions and Proposed Future Work

The 2-D/3-D method was capable of producing very accurate results for the C5G7 benchmarks on all measures of interest. While 2-D/3-D did not outperform 2-D/1-D on all measures, overall it produced results closer to the reference solution. The most consistent area where the 2-D/1-D SP_3 method performed better was in the top region of the core, where the active core region interfaces with the axial reflector region. The one-way 2-D/3-D methods performed rather horribly in there area, while the two-way methods performed better but not as well as the SP_3 method. In most other regions of the core 2-D/3-D outperformed SP_3. Specifically, among the 2-D/3-D methods studied, the DD axial treatment was generally capable of producing the best results. Unfortunately, due to its difficult behavior and propensity to generate negative fluxes, in many cases it prevented the use of Coarse Mesh Finite Difference (CMFD) acceleration, which is a virtual necessity to obtain solu-

tions in reasonable time. A negative flux fix-up was applied to the DD equations which rendered the CMFD iteration robust. This fix-up performed admirably for the unrodded and rodded A cases, but it introduced large errors in the rodded B case. The SC differencing scheme, while positive and well-behaved, tended to produce poor results owing to its first-order error convergence properties. The difficulties presented by the DD equations and the lack of accuracy from SC motivates the development of a 2-D/3-D method that employs a higher-order or more sophisticated axial treatment.

As discussed in Chapter 5, the memory burden of the CDD correction factors appears large at first blush, but under most circumstances ought not to pose a challenge. Even though storage of these factors requires nearly one hundred gigabytes of memory for a typical 3-D quarter-core problem, such large-scale calculations are performed on thousands of CPUs, requiring only tens of megabytes per core. That being said, compression of these correction factors to save memory would prove valuable.

The computational cost of the 3-D S_N solver was found to be larger than that of the SP_3 method to which it was compared. Configured as it was, the S_N sweeps constituted about a quarter of the time spent performing MoC sweeps, whereas the SP_3 solver required 10-15% of the MoC time. That being said the S_N algorithm that was used was fairly rudimentary, with little effort having been spent on optimization. Implementation of more advanced parallel algorithms, such as Koch-Baker-Alcouffe (KBA), and a more efficient implementation overall could see a large reduction in the S_N overhead. Furthermore, orthogonal-mesh S_N is particularly well-suited for being offloaded to massively-parallel co-processors, such as general-purpose GPUs, Intel MICs, or similar. Doing so effectively would make the S_N iterations virtually free.

6.3 Future Work

Based on the conclusions discussed above, the following make for promising topics of future work:

- Memory consumption is a negative aspect of of the CDD equations. For this work, the correction factors were stored explicitly in angle, requiring two double-precision floating-point values per angle per region per energy group. While not prohibitive, memory savings would leave more space for other uses, such a multi-physics coupling, depletion analysis, or other potentially memory-intensive tasks. At the expense of accuracy, the corrections factors could be stored as moments of a functional expansion. A Fourier series expansion could be used for each polar angle of a product quadrature, or a spherical harmonics expansion could be used in the more general, level-symmetric quadrature case. Examining the angular behavior of the correction factors (Section 3.3.1), it appears that capturing the angular dependence of the high-energy groups might be difficult, while the lower energy groups might be well approximated with few moments. Furthermore, advantage may be taken of common features of the angular variation of the correction factors, such as the tendency of the corrections to assume their largest magnitude in angles most orthogonal to the direction of travel being corrected.

- The other challenge to 2-D/3-D is the speed performance of the S_N solver used. Implementing a better S_N sweeper in MOCC/MPACT would go a long way towards making 2-D/3-D more attractive.

- 2-D/3-D could be used to assist in control rod "decusping". 2-D/1-D methods have difficulty treating situations in which a control rod is inserted partially into an MoC plane. The use of a finer axial mesh for the S_N sweeper could be used to resolve the axial flux shape through an MoC plane, potentially providing better plane-homogenized cross sections.

- Implementing the S_N sweep on a co-processor such as a GPU would expose the possibility of developing adaptive algorithms in which the S_N and MoC sweepers operate in conjunction, further reducing the overhead of the 3-D S_N solution.

- While the 2-D/3-D method managed to produce more accurate results than 2-D/1-D for the C5G7 benchmark problems, the error associated with the transverse leakage source becomes the limiting factor in many planar synthesis methods, 2-D/1-D included. This work used the isotropic-in-angle, flat-in-space transverse leakage approximation, which is very crude. Applying angular and/or spatial shape functions to the transverse leakage is likely to further increase the accuracy of the method. Refining the S_N mesh in the radial directions could provide an intra-pin transverse leakage shape to the MoC sweeper, though the interaction of the coarse and fine meshes at the sub-pin level may be complicated.

BIBLIOGRAPHY

[1] R. E. Alcouffe. Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations. *Nuclear Science and Engineering*, 64(1):344–355, 1977.

[2] C. Baker, G. Davidson, T. M. Evans, S. Hamilton, J. Jarrell, and W. Joubert. High performance radiation transport simulations: Preparing for TITAN. *SC 2012 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2012.

[3] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

[4] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2016.

[5] William Boyd, Andrew Seigel, Shuo He, Benoit Forget, and Kord Smith. Parallel performance results for the OpenMOC neutron transport code on multicore platforms. *The International Journal of High Performace Computing Applications*, 2016.

[6] William R. Boyd, Kord Smith, and Benoit Forget. A massively parallel method of characteristic neutral particle transport code for GPUs. *ANS MC2013*, 2013.

[7] Jin Young Cho and Han Gyu Joo. Solution of the C5G7MOX benchmark three-dimensional extension problems by the DeCART direct whole core calculation code. *Progress in Nuclear Energy*, 48:456–466, 2006.

[8] Benjamin Collins, Shane Stimpson, Blake Kelley, Mitchell T. H. Young, Brendan Kochunas, Edward W. Larsen, Thomas Downar, and Andrew Godfrey. Stability and accuracy of three-dimensional neuton transport simulations using the 2D/1D method in MPACT. *Journal of Computational Physics*, 326(1):612–628, 2016.

[9] Mark D. DeHart. Advancements in general-geometry discrete ordinates transport for lattice physics calculations. *Proc. of PHYSOR-2006*, pages 10–14, 2006.

[10] Thomas M. Evans, Alissa S. Stafford, Rachel N. Slaybaugh, and Kevin T. Clarno. Denovo: A new three-dimensional parallel discrete ordinates code in SCALE. *Nuclear Technology*, 171(2):171–200, August 2010.

[11] Tom Evans, Greg Davidson, Josh Jarrell, Steven Hamilton, Seth Johnson, and Tara Pandya. Exnihilo transport methods manual. February 2015.

[12] Andrew T. Godfrey. VERA core physics benchmark progression problem specifications. 2013. AMA advanced modeling and applications.

[13] M. J. Halsall. CACTUS, a characteristics solution to the neutron transport equations in complicated geometries. Technical report, United Kingdom Atomic Energy Authority, 1980.

[14] Mathieu Hursin, Benjamin Collins, Yunlin Xu, and Thomas Downar. The development and implementation of a one-dimensional Sn method in the 2D-1D integral transport equation. *Nuclear Science and Engineering*, 176:186–200, 2014.

[15] Joshua J. Jarrell, Thomas M. Evans, Gregory G. Davidson, and Andrew T. Godfrey. Full core reactor analysis: Running Denovo on Jaguar. *Nuclear Science and Engineering*, 175:283–291, 2013.

[16] Han Gyu Joo, Jin Young Cho, Kang Seog Kim, Chung Chan Lee, and Sung Quun Zee. Methods and performance of a three-dimensional whole-core transport code DeCART. *PHYSOR - The Physics of Fuel Cycles and Advanced Nuclear Systems, Lagrange Park, IL*, 2004.

[17] Blake W. Kelley, Benjamin Collins, and Edward W. Larsen. 2D/1D approximations to the 3D neutron transport equation II: numerical comparisons. *Proc. American Nuclear Society, Mathematics and Computation, Sun Valley, ID*, pages 1358–1368, 2013.

[18] Blake W. Kelley and Edward W. Larsen. 2D/1D approximations to the 3D neutron transport equation. I: theory. *Proc. American Nuclear Society, Mathematics and Computation, Sun Valley, ID*, pages 1346–1357, 2013.

[19] Dave Knott and Akio Yamamoto. *Handbook of Nuclear Engineering*, chapter 9, pages 913–1239. Springer, 2010.

[20] K. R. Koch, R. S. Baker, and Raymond E. Alcouffe. A parallel algorithm for 3D S_n transport sweeps. Technical report, LA-CP-92-406, Los Alamos National Laboratory, 1992.

[21] Brendan Kochunas. *A Hybrid Parallel Algorithm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Compute Clusters*. PhD thesis, University of Michigan, 2013.

[22] Brendan Kochunas, Benjamin Collins, Dan Jabaay, Thomas J. Downar, and William R. Martin. Overview of development and design of MPACT michigan parallel characteristics transport code. *Proc. American Nuclear Society, Mathematics and Computation, Sun Valley, ID*, 1:42–53, 2013.

[23] Brendan Kochunas, Dan Jabaay, Benjamin Collins, and Thomas Downar. Demonstration of neutronics coupled to thermal-hydraulics for a full-core problem using COBRA-TF/MPACT. Technical report, University of Michigan - Consortium for the Advanced Simulation of Light Water reactors, 2014.

[24] K.D. Lathrop. Spatial differencing of the transport equation: Positivity vs. accuracy. *Journal of Computational Physics*, 4:475–498, 1969. step characteristics, Wendroff scheme.

[25] Gil Soo Lee and Nam Zin Cho. 2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX. *Progress in Nuclear Energy*, 48:410–423, 2006.

[26] E.E. Lewis. Benchmark on deterministic transport calculations without spatial homogenization: a 2-D/3-D MOX fuel assembly benchmark. Technical Report NEA/NSC/DOC(2003)16, Nuclear Energy Agency Organisation for Economic Co-Operation and Development, 2003.

[27] Karl O. Ott and Robert J. Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, 1985.

[28] K. S. Smith. Assembly homogenization techniques for light water reactor analysis. *Progress in Nuclear Energy*, 17(3):303–335, 1986.

[29] M.A. Smith and E.E. Lewis. Benchmark on deterministic transport calculations without spatial homogenisation: MOX fuel assembly 3-D extension case. Technical Report NEA/NSC/DOC(2005)16, Nuclear Energy Agency Organisation for Economic Co-Operation and Development, 2005.

[30] Shane Stimpson, Benjamin Collins, and Thomas Downar. An azimuthal, Fourier moment-based transverse leakage approsimation for the MPACT 2D/1D method. In *ANS MC2015*, 2015.

[31] Shane G. Stimpson. *An Azimuthal, Fourier Moment-Based Axial Sn Solver for the 2D/1D Scheme*. PhD thesis, University of Michigan, 2015.

[32] The MPACT Team. *MPACT User's Manual*. University of Michigan, Ann Arbor, MI, November 2013.

[33] Akio Yamamoto, Masato Tabuchi, Naoki Sugimura, Tadashi Ushio, and Masaaki Mori. Derivation of optimum polar angle quadrature set for the method of characteristics based on approximation error for the bickley function. *Journal of Nuclear Science and Technology*, 44(2):129–136, 2007.

[34] Mitchell T. H. Young. MOCC: A method of characteristics-based nuclear reactor physics simulator. https://github.com/youngmit/mocc, 2016.

[35] Mitchell T. H. Young, Joshua J. Jarrell, and Tom Evans. Method of characteristics assisted cross section homogenization for denovo. Technical report, Oak Ridge National Laboratory, 2012.