Homogenization Methods for Full Core Solution of the Pn Transport Equations with 3-D Cross Sections

by

Andrew Charles Hall

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2015

Doctoral Committee:

Professor Thomas J. Downar, Chair
Assistant Professor Brian C. Kiedrowski
Professor Edward W. Larsen
Professor John C. Schotland
Adjunct Professor Yunlin Xu

2015

For Suzanne, Steve and Stephanie

Acknowledgements

I would like to first thank my advisor Dr. Thomas Downar for his guidance, support and patience during my time at the University of Michigan. I would also like to express my gratitude to Dr. Yunlin Xu for providing his insight and expertise. Without their contributions, this would not have been possible.

My committee provided numerous ideas that led to the development of this research. I would like to thank Dr. Edward Larsen, Dr. Brian Kiedrowski and Dr. John Schotland. Their knowledge, advice and criticism were invaluable to improving this thesis.

I also would like to acknowledge all of the people who worked on the RBWR project with me and HITACHI who developed the reactor. Thank you to Dr. Andrew Ward, Dr. Koroush Shirvan, Daniel Jabaay, Mike Jarrett and the many others who provided assistance. I would also like to thank the DOE NEUP program for sponsoring this work under grant NEUP-09-812.

Special thanks to Dr. Jaakko Leppänen and the Serpent development team for providing such an outstanding code. I would also like to thank the Serpent community for their assistance and contributions. Also, I owe a great deal of gratitude to Dr. Volkan Seker who performed numerous simulations for this work.

Thank you to all my friends and colleagues, especially those of 144 Hill Street. Without your constant support, this process would have been far more grueling. I also owe an immense amount of gratitude to Kim for being there every step of the way and lifting my spirits on the bad days. Lastly, I would like to thank my family (Suzanne, Steve and Stephanie) for their unending love and instilling in me the importance of education and never giving up.

Table of Contents

List of Figures

List of Tables

List of Appendices

Abstract

The design of advanced light water reactors having reduced moderation and axially varying fuel has exposed the limitations of diffusion theory methods based on two-dimensional homogenized group constants. Instead, an alternative approach is required to account for the axial neutron streaming within the core. The development of three-dimensional Monte Carlo cross sections and axial discontinuity factors has improved upon the lower-order diffusion solution, but numerical instabilities can still arise from large discontinuity factors. Therefore the use of higher-order transport corrections is necessary for the solution of full core problems.

In this thesis the spherical harmonics (Pn) and Quasidiffusion equations are derived for one-dimensional applications to improve upon the angular approximation implicit in diffusion. Discontinuity factors for each method were determined based on finite difference, as well as the generation of Eddington factors from Monte Carlo results. A subplane method based on refining the Pn and Quasidiffusion solution was introduced to reduce the spatial discretization error. An alternative definition of the discontinuity factor based on an additive relation was investigated to improve the numerical stability.

Numerical results based on an axially heterogeneous assembly demonstrate that P3 and Quasidiffusion improved the accuracy of the spatial flux distribution the most. Discontinuity factors allowed each of the lower-order methods to reproduce the reference Monte Carlo eigenvalue and flux. Minor improvement was seen when bounding the discontinuity factors compared to diffusion. The addition of the subplane method reduced the spatial discretization error and improved the range of discontinuity factors seen for all four methods at the cost of increased computational run time. Additive discontinuity factors for each method eliminated the possibility of large discontinuity factors and were able to reproduce the reference solution. The combination of the Quasidiffusion and subplane methods provided the most accurate axial solution.

Chapter 1
Introduction

1.1 Motivation

The modeling of nuclear reactors has evolved over the years to account for new and innovative designs, as well as for the considerable advances in computational performance. The current generation of neutron transport methods for solving the whole-core reactor problem generally involves a two-step approach where a high-fidelity transport calculation is performed on a smaller fuel assembly lattice size problem to generate homogenized spatial and energy group constants (cross sections) for a lower-order nodal method to solve the whole-core problem [1]. For the assembly lattice problem, several transport methods have been used including collision probability [2], method of characteristics [3] [4] and discrete ordinates (Sn) [5]. Traditionally, all of these methods are applied to a two-dimensional radial problem with an assumed axial boundary condition. However, several new reactor designs such as the Resource Renewable Boiling Water Reactor (RBWR) have introduced axial heterogeneity into their fuel models [6] and the two-dimensional transport calculation has not been sufficient to capture the spectral effects introduced from this axial heterogeneity. Instead, three-dimensional Monte Carlo methods have been used to properly model the neutron spectrum within each region and generate 3-D homogenized cross sections for the core calculation. Similarly, the lower-order diffusion-based nodal methods for solving the core calculation has had a difficult time accounting for the severe flux gradients that occur at material interfaces. By implementing a higher-order transport solution, such as Pn spherical harmonics instead of diffusion theory, improved flux and power can be predicted for these complex designs. The objective of the research here is to investigate methods for generating three-dimensional Monte Carlo cross

sections for use in the Pn transport equations, which can be applied to complex reactor designs such as the RBWR.

1.2 Reactor Physics Equations

1.2.1 Time-Independent Boltzmann Transport Equation

To illustrate the methods discussed in this work, some background is provided on neutron-nuclear interactions in a reactor. The behavior of neutrons within a given system is governed by the time-dependent Boltzmann transport equation:

$$\begin{aligned}\frac{1}{v}\frac{\partial\psi}{\partial t}(\vec{r},E,\vec{\Omega},t)+\ \vec{\Omega}\cdot\nabla\psi(\vec{r},E,\vec{\Omega},t)+\Sigma_t(\vec{r},E)\psi(\vec{r},E,\vec{\Omega},t)\\ =\int_{4\pi}d\vec{\Omega}'\int_0^\infty dE'\,\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\cdot\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}',t)\\ +\frac{\chi(E)}{4\pi}\int_{4\pi}d\vec{\Omega}'\int_0^\infty dE'\,\nu\Sigma_f(\vec{r},E')\psi(\vec{r},E',\vec{\Omega}',t)+q_{ex}(\vec{r},E,\vec{\Omega},t).\end{aligned} \tag{1-1}$$

In this equation, ψ is the angular neutron flux as a function of space ($\vec{r}$), energy (E), angle ($\vec{\Omega}$) and time (t). The quantity "ν " represents the neutron speed and Σ_u is the macroscopic cross-section of type "u" which include total (t), scatter (s) and fission (f). $\chi(E)$ represents the probability that a fission neutron is created in *dE* about energy *E* and "ν" is the average number of neutrons generated per fission. The term "q_{ex}" is an external source. Both E' and $\vec{\Omega}'$ represent the incoming neutron energy and angle respectively.

The first term on the left-hand side represents the time-rate of change in the angular neutron density. The second term describes neutron streaming in the direction $\vec{\Omega}$ that enter and leave the differential volume d$\vec{r}$. The third term represents interactions that remove neutrons from the flux through absorption or scattering away from the differential angular and energy space. On the right-hand side, the first term represents neutrons with energy E' and direction $\vec{\Omega}'$ that scatter into the interval *dE* about energy *E* and $d\vec{\Omega}$ about angle $\vec{\Omega}$ at position $\vec{r}$. The second term on the right-hand side represents neutrons born from fission into the interval

dE about energy *E.* And the final term on the right-hand side represents the addition of neutrons into the system from an external source.

For most applications the final q_{ex} term is ignored, since there is no external source applied to the problem. The time derivative is set to zero for steady-state applications which removes the time dependence. In this equation it is assumed that the fission neutrons are emitted isotropically and all fission neutrons are emitted promptly. This solution of this equation is nontrivial due to the complicated energy and space-dependency of the cross sections, the angular dependency of the scattering cross section and complexity due to the streaming term. Each method discussed in this research is either based on or derived from Eq. 1-1.

The scattering cross section can be expanded using Legendre polynomials for the scattering cosine $\mu_0 = \vec{\Omega}' \cdot \vec{\Omega}$ without azimuthal integration as:

$$\Sigma_s(\vec{r}, E' \to E, \mu_0) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \Sigma_{sn}(\vec{r}, E' \to E) P_n(\mu_0), \tag{1-2}$$

where, for $0 \le n \le \infty$ the nth Legendre moment of the scattering cross section is,

$$\Sigma_{sn}(\vec{r}, E' \to E) = 2\pi \int_{-1}^{1} P_n(\mu_0') \Sigma_s(\vec{r}, \mu_0') d\mu_0', \tag{1-3}$$

and the first two Legendre polynomials and recurrence relation are,

$$P_0(\mu_0) = 1, \tag{1-4a}$$

$$P_1(\mu_0) = \mu_0, \tag{1-4b}$$

$$(n+1)P_{n+1}(\mu_0) = (2n+1)\mu_0 P_n(\mu_0) - nP_{n-1}(\mu_0). \tag{1-4c}$$

The goal in reactor analysis is to find a configuration in which the system is in steady-state. This is referred to as a critical configuration. It exists when both sides of the steady-state transport equation are equal, or when the production of neutrons balances the removal of neutrons from the system. Determining a time-dependent solution for this situation is

computationally difficult, so an approximate method is devised that is reasonably valid near steady-state. In this expression, the time derivative is asserted to be zero and the balance of the transport equation is ensured by artificially increasing or decreasing the fission source by a multiplicative factor $\lambda = 1/k$. This form of the transport equation is an eigenvalue problem, and is much simpler to solve. The k-eigenvalue equation is only valid at $k = 1$, due to a bias introduced in the energy spectrum by this approximation. However, the errors introduced are typically small, allowing this form of the transport equation to be applied to non-critical systems.

If a given system has a fissile region in which the fission cross section is non-zero, then the criticality eigenvalue k always exists, and the corresponding angular flux, or eigenfunction, is unique and positive. The goal is to find the largest value of k such that the solution of the angular flux exists and is nonzero. When the criticality eigenvalue is exactly one ($k = 1$), then the production of neutrons due to fission exactly balances the loss of neutrons due to capture and leakage. However, if the criticality eigenvalue is less than one ($k < 1$), then the system is subcritical and leakage and capture dominate fission. If the criticality eigenvalue is greater than one ($k > 1$), then the system is supercritical and fission dominates leakage and capture. The time-independent Boltzmann transport equation for eigenvalue problems is the following:

$$\begin{aligned}\vec{\Omega}\cdot\nabla\psi\left(\vec{r},E,\vec{\Omega}\right)+\Sigma_t(\vec{r},E)\psi\left(\vec{r},E,\vec{\Omega}\right)&\\ =\int_{4\pi}d\vec{\Omega}'\int_0^{\infty}dE'\sum_{n=0}^{\infty}\frac{2n+1}{4\pi}\Sigma_{sn}(\vec{r},E'\rightarrow E)P_n(\mu_0)\,\psi\left(\vec{r},E',\vec{\Omega}'\right)&\\ +\frac{\lambda\chi(E)}{4\pi}\int_{4\pi}d\vec{\Omega}'\int_0^{\infty}dE'\,\nu\Sigma_f(\vec{r},E')\psi\left(\vec{r},E',\vec{\Omega}'\right).&\end{aligned} \tag{1-5}$$

1.2.2 Neutron Diffusion Equation

The most common method for solving the Boltzmann transport equation for practical Light Water Reactor (LWR) applications has been to assume that the angular flux can be approximated as a linear function of angle [7]. The scalar neutron flux (Eq. 1-6) and the neutron current (Eq. 1-7) are defined as:

$$\phi(\vec{r},E)=\int_{4\pi}\psi\left(\vec{r},E,\vec{\Omega}\right)d\vec{\Omega}, \tag{1-6}$$

$$\vec{J}(\vec{r},E)=\int_{4\pi}\vec{\Omega}\psi\left(\vec{r},E,\vec{\Omega}\right)d\vec{\Omega}. \tag{1-7}$$

If the first two terms and last term of the transport equation are integrated over angle and Eq. 1-6 and Eq. 1-7 are used:

$$\int_{4\pi}d\vec{\Omega}\vec{\Omega}\cdot\nabla\psi\left(\vec{r},E,\vec{\Omega}\right)=\int_{4\pi}d\vec{\Omega}\nabla\vec{\Omega}\psi\left(\vec{r},E,\vec{\Omega}\right)=\nabla\vec{J}(\vec{r},E), \tag{1-8a}$$

$$\int_{4\pi}d\vec{\Omega}\Sigma_t(\vec{r},E)\psi\left(\vec{r},E,\vec{\Omega}\right)=\Sigma_t(\vec{r},E)\phi(\vec{r},E), \tag{1-8b}$$

$$\begin{aligned}\int_{4\pi}d\vec{\Omega}\frac{\lambda\chi(E)}{4\pi}\int_{4\pi}d\vec{\Omega}'\int_0^\infty dE'\,\nu\Sigma_f(\vec{r},E')\psi\left(\vec{r},E',\vec{\Omega}'\right)\\ =\lambda\chi(E)\int_0^\infty dE'\,\nu\Sigma_f(\vec{r},E')\phi(\vec{r},E'),\end{aligned} \tag{1-8c}$$

this provides expressions for three out of the four terms of the transport equation without angular dependence. The integration of the scattering term is less straight forward due to the angular dependence of the cross section. However, it is important to note that the angular dependence of the scattering cross section relies only on the scattering cosine $\vec{\Omega}'\cdot\vec{\Omega}$, and therefore the order of integrations over $\vec{\Omega}$ and $\vec{\Omega}'$ can be rewritten. The integration of the scattering term then becomes:

$$\begin{aligned}&\int_{4\pi}d\vec{\Omega}\int_{4\pi}d\vec{\Omega}'\int_0^\infty dE'\,\Sigma_s\left(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega}\right)\psi\left(\vec{r},E',\vec{\Omega}'\right)\\ &=\int_{4\pi}d\vec{\Omega}'\int_{4\pi}d\vec{\Omega}\int_0^\infty dE'\,\Sigma_s\left(\vec{r},E'\to E,\vec{\Omega}\to\vec{\Omega}'\right)\psi\left(\vec{r},E',\vec{\Omega}'\right)\\ &=\int_{4\pi}d\vec{\Omega}'\int_0^\infty dE'\,\Sigma_s(\vec{r},E'\to E)\psi\left(\vec{r},E',\vec{\Omega}'\right)\\ &=\int_0^\infty dE'\,\Sigma_s(\vec{r},E'\to E)\phi(\vec{r},E').\end{aligned} \tag{1-8d}$$

The combination of these four terms produces the neutron continuity equation:

$$\begin{aligned}\nabla \cdot \vec{J}(\vec{r},E) + \Sigma_t(\vec{r},E)\phi(\vec{r},E) \\ = \int_0^\infty dE'\,\Sigma_s(\vec{r},E'\to E)\phi(\vec{r},E') \\ + \lambda\chi(E)\int_0^\infty dE'\,\nu\Sigma_f(\vec{r},E')\phi(\vec{r},E').\end{aligned} \tag{1-9}$$

If the transport equation is multiplied by $\overrightarrow{\Omega}$ and integrated over angle using a similar process for the derivation of the neutron continuity equation, this leads to the following expression:

$$\nabla \cdot \int_{4\pi} d\overrightarrow{\Omega}\,\overrightarrow{\Omega}\overrightarrow{\Omega}\psi\left(\vec{r},E,\overrightarrow{\Omega}\right) + \Sigma_t(\vec{r},E)\vec{J}(\vec{r},E) = \int_0^\infty dE'\,\Sigma_{s1}(\vec{r},E'\to E)\vec{J}(\vec{r},E'), \tag{1-10}$$

where the first-moment of the scattering cross section is found from,

$$\int_{4\pi} d\,\overrightarrow{\Omega}\int_{4\pi} d\,\overrightarrow{\Omega}'\int_0^\infty d\,E'\overrightarrow{\Omega}\Sigma_s\left(\vec{\mathrm{r}},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega},\mathrm{E}'\to\mathrm{E}\right)\psi\left(\vec{r},\overrightarrow{\Omega}',\mathrm{E}'\right) = \int_0^\infty d\,E'\Sigma_{s1}(\vec{r},\mathrm{E}'\to\mathrm{E})\vec{J}(\vec{r},\mathrm{E}').$$

The fission term disappears in Eq. 1-10 due to the isotropic nature of the fission source. The first term in Eq. 1-10 is not easily defined, so first the expression is modified by placing the omegas outside of the gradient using the following relation:

$$\nabla\cdot\left(\Omega_u\overrightarrow{\Omega}\psi\right) = \Omega_u\nabla\cdot\left(\overrightarrow{\Omega}\psi\right) + \left(\overrightarrow{\Omega}\psi\right)\cdot\nabla\Omega_u = \Omega_u\nabla\cdot\left(\overrightarrow{\Omega}\psi\right) = \Omega_u\overrightarrow{\Omega}\cdot\nabla\psi. \tag{1-11}$$

Next, the gradient of the angular flux is approximated as a linear function of angle:

$$\nabla\psi\left(\vec{r},E,\overrightarrow{\Omega}\right) \approx \nabla\left(\phi_0(\vec{r},E) + J_x(\vec{r},E)\Omega_x + J_y(\vec{r},E)\Omega_y + J_z(\vec{r},E)\Omega_z\right), \tag{1-12}$$

where ϕ_0 is the scalar flux and J_x, J_y, and J_z are the angular components of the current.

When Eq. 1-12 is substituted into the first term of Eq. 1-10, the integral simplifies to the following definition:

$$\nabla \cdot \int_{4\pi} d\vec{\Omega}\, \vec{\Omega}\vec{\Omega}\psi\left(\vec{r}, E, \vec{\Omega}\right) = \frac{1}{3}\nabla\phi(\vec{r}, E). \tag{1-13}$$

The combination of Eqs. 1-9 and 1-10 are considered the P1 equations. Next, the anisotropic contribution to energy transfer in a scattering collision is neglected. This is done using the Dirac δ-function so the scattering term becomes only a function of E:

$$\begin{aligned}\int_0^\infty d\,E'\Sigma_{s1}(\vec{r}, \mathrm{E}' \to \mathrm{E})\vec{J}(\vec{r}, \mathrm{E}') &= \int_0^\infty d\,E'\Sigma_{s1}(\vec{\mathrm{r}}, \mathrm{E}')\delta(E' - E)\vec{J}(\vec{r}, \mathrm{E}') \\ &= \Sigma_{s1}(\vec{\mathrm{r}}, E)\vec{J}(\vec{\mathrm{r}}, E).\end{aligned} \tag{1-14}$$

Since the neutron current no longer depends on E' in the scattering term, this simplifies Eq. 1-10 to the following form:

$$\frac{1}{3}\nabla\phi(\vec{r}, E) + \Sigma_{tr}(\vec{r}, E)\vec{J}(\vec{r}, E) = 0, \tag{1-15}$$

where the transport cross section (Σ_{tr}) is defined as:

$$\Sigma_{tr}(\vec{r}, E) = \Sigma_t(\vec{r}, E) - \Sigma_{s1}(\vec{r}, E).$$

If the terms in Eq. 1-15 are rearranged to solve for the neutron current, we obtain the following expression:

$$\vec{J}(\vec{r}, E) = -\frac{1}{3\Sigma_{tr}(\vec{r}, E)}\nabla\phi(\vec{r}, E), \tag{1-16}$$

where the neutron diffusion coefficient D is defined as:

$$D(\vec{r}, E) = \frac{1}{3\Sigma_{tr}(\vec{r}, E)}.$$

Equation 1-16 is what is often referred to as Fick's law. This expression relies on three approximations:

1) The angular flux is at most linearly dependent on the angular variable.

2) The time derivative of the neutron current density is small relative to the flux gradient.

3) There is no anisotropic energy-transfer in group-to-group scattering.

If the expression found for the neutron current is substituted into Eq. 1-9, we obtain the neutron diffusion equation:

$$\begin{aligned} -\nabla \cdot D(\vec{r},E)\nabla\phi(\vec{r},E) + \Sigma_t(\vec{r},E)\phi(\vec{r},E) \\ = \int_0^\infty dE'\,\Sigma_{s0}(\vec{r},E'\rightarrow E)\phi(\vec{r},E') \\ + \lambda\chi(E)\int_0^\infty dE'\,\nu\Sigma_f(\vec{r},E')\phi(\vec{r},E'). \end{aligned} \tag{1-17}$$

Computer codes to solve the neutron diffusion equation for large scale LWR problems have generally relied on coarse-mesh nodal methods, based on homogenizing fuel assembly size regions of the reactor. Nodal diffusion methods have generally provided sufficient accuracy and efficiency to solve the range of problems encountered in practical reactor design and analysis applications. However, it is important to note that the diffusion approximation has difficulty with the following problems:

1) Regions near vacuum boundaries and low-density material regions.

2) Regions with high-absorbing materials, such as control rods or burnable absorbers.

3) Regions with highly anisotropic scattering properties, such as moderator channels.

It is difficult to avoid these kinds of regions, since they typically arise in most reactor applications. However, if the behavior of the neutron flux does not have strong spatial dependence and has a small flux gradient, then diffusion provides an accurate estimate. There is one additional assumption typically made when solving reactor problems and that is the multigroup approximation.

1.2.3 Multigroup Diffusion Approximation

The diffusion equation defined in the previous section is continuous in energy. Except for extremely simple cases, it is not feasible to solve exact solutions of this form. Instead, the energy variable is discretized over a range into G intervals. This range spans from 0 eV to a large

enough energy such that all neutrons are represented. If the continuous-energy neutron diffusion equation is integrated over an energy group g, this produces the multigroup neutron diffusion equation [7]:

$$-\nabla \cdot D_g(\vec{r})\nabla\phi_g(\vec{r}) + \Sigma_{r,g}(\vec{r})\phi_g(\vec{r}) = \sum_{g'=1,g'\neq g}^{G} \Sigma_{s,g'\to g}(\vec{r})\phi_{g'}(\vec{r}) + \lambda\chi_g \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\vec{r})\phi_{g'}\,(\vec{r}), \tag{1-18a}$$

where the integral over an energy group g is:

$$\int_{\mathrm{g}} dE f(E) = \int_{\mathrm{E_g}}^{\mathrm{E_{g-1}}} dE f(E), \tag{1-18b}$$

and the terms in Eq. 1-18a are defined as:

$$\phi_g(\vec{r}) = \int_g dE\, \phi(\vec{r},E), \tag{1-18c}$$

$$\chi_g(\vec{r}) = \int_g dE\, \chi(\vec{r},E), \tag{1-18d}$$

$$\Sigma_{t,g}(\vec{r}) = \frac{\int_g dE\, \Sigma_t(\vec{r},E)\phi(\vec{r},E)}{\int_g dE\, \phi(\vec{r},E)}, \tag{1-18e}$$

$$\Sigma_{s,g'\to g}(\vec{r}) = \frac{\int_g dE \int_{g'} dE'\, \Sigma_{s0}(\vec{r},E'\to E)\phi(\vec{r},E')}{\int_{g'} dE'\, \phi(\vec{r},E')}, \tag{1-18f}$$

$$\nu\Sigma_{f,g'}(\vec{r}) = \frac{\int_{g'} dE' \nu\, \Sigma_f(\vec{r},E')\phi(\vec{r},E')}{\int_{g'} dE'\, \phi(\vec{r},E')}, \tag{1-18g}$$

$$\Sigma_{r,g}(\vec{r}) = \Sigma_{t,g}(\vec{r}) - \Sigma_{s,g'\to g}(\vec{r}), \tag{1-18h}$$

$$D_g(\vec{r}) = \frac{\int_g dE D(\vec{r},E)\nabla\phi(\vec{r},E)}{\int_g dE \nabla\phi(\vec{r},E)}. \tag{1-18i}$$

The majority of the terms are found using a flux-weighted average. However, it is important to note that the multigroup diffusion coefficient is found by taking the weighted average based on

the gradient of the flux. Equations 1-18 are exact when the scalar flux is known for the global problem. In practice though, the global solution is unknown and the terms in Eq. 1-18c through Eq. 1-18i are approximated based on a smaller lattice calculation with zero current boundary conditions. These group constants generated from the lattice level calculation are then used to solve the global homogenized multigroup neutron diffusion equation. The simulation could be improved by taking the assembly boundary conditions generated from the global solution and using them to produce a new set of cross sections. However, this would involve iterating between the lattice and global solution for every assembly until the two solutions converge. To simplify the procedure, the group constants for the global problem are approximated by a lattice calculation with zero current boundary conditions. This process of homogenization is described in the following section (Section 1.3).

Since there is no external source, then the multigroup diffusion equations are an eigenvalue problem, where λ is the inverse of the criticality eigenvalue k. This is the typical form of the diffusion equation that is solved for criticality problems.

1.3 Homogenization Methods

1.3.1 Cross Section Homogenization

The probability that a neutron will have an interaction is governed by its cross section. Thousands of experimental data points have been tabulated for various interactions at different energies in different materials. These microscopic cross sections are used for lattice-level calculations. Deterministic lattice codes such as CASMO [8] and HELIOS [2] have pre-built libraries that group these microscopic cross sections into a few hundred to thousand energy groups. Monte Carlo codes use a "continuous-energy" approach, which takes advantage of the whole set of data points. It is often useful to determine the macroscopic cross section, which is the sum of the product of the microscopic cross section and number density for each isotope *i*:

$$\Sigma(\vec{r},E) = \sum_{i=1}^{I} N_i(\vec{r},E)\sigma_i(\vec{r},E). \tag{1-19}$$

The inverse of the total macroscopic cross section provides the mean free path, or average distance a neutron travels, before a neutron interacts within a given material at a given energy. However, considerable information would have to be stored and tabulated for a full-core calculation. A more efficient method is to collapse this information into a smaller number of energy group cross sections that allow the nodal deterministic solution to preserve reaction rates of the node and multiplication factor of the core. This allows for simulations based on large coarse meshes (nodes) in which the properties are constant.

For the first step, a detailed lattice calculation is performed in space and energy over a small region of the core. In traditional calculations, this involves modeling a 2-D slice of an assembly, while the work presented here investigates modeling a full 3-D assembly. Figure 1.1 provides a visual of the homogenization process. For a region of interest, it is important to preserve the group reaction rates, group surface currents and reactor eigenvalue. This is done by collapsing the cross sections into a multigroup form. The homogenized parameters are found from:

$$\Sigma_{\alpha,g}^{hom,i} = \frac{\int_{V_i} \Sigma_{\alpha,g}^{het} \phi_g^{het}(\vec{r}) d\vec{r}}{\int_{V_i} \phi_g^{hom}(\vec{r}) d\vec{r}}, \tag{1-20}$$

$$D_g^{hom,i} = -\frac{\int_S \vec{J}_g^{het}(\vec{r}) \cdot d\vec{S}}{\int_S \nabla \phi_g^{hom}(\vec{r}) \cdot d\vec{S}}, \tag{1-21}$$

where α represents the various potential reactions that can take place within the system (total, absorption, scatter, etc.). Equations 1-20 and 1-21 are the volume integrated form of the group constants from Eq. 1-18 for region i. It is important to pick a set of energy groups that preserves the physics observed within the lattice code. This is usually dependent upon the spectrum observed within the region of interest. For a reactor designed to burn and breed, it is important to pick energy groups that capture the most important resonances of the fissionable isotopes.

This process of homogenization is carried out over various state conditions (fuel temperature, coolant temperature, coolant density, moderator density, control rod position,

fuel burnup, etc.). The idea is to determine group constants for a range of conditions that are possible within the core calculation. Thus, the homogenization calculation must be performed multiple times for each state calculation. These various simulations are referred to as branch calculations. Further detail on branch calculations are discussed in the second chapter.

The homogenization technique provides a group of constants that can be used to solve the full-core problem with much less effort. Though computational efficiency and power have increased over the years, running full-core transport calculations are still not computationally practical for analyzing the wide range of reactor conditions required in safety analysis. Therefore homogenization techniques and deterministic calculations will continue to be important for the foreseeable future. However, the use of Monte Carlo to generate multi-dimensional homogenized group constants has increased recently due to the development of large high performance computing machines. Appendix A provides information on the fundamentals of the Monte Carlo method, while later in this chapter discusses how Monte Carlo is applied for the purposes of homogenization for lower-order full-core calculations.

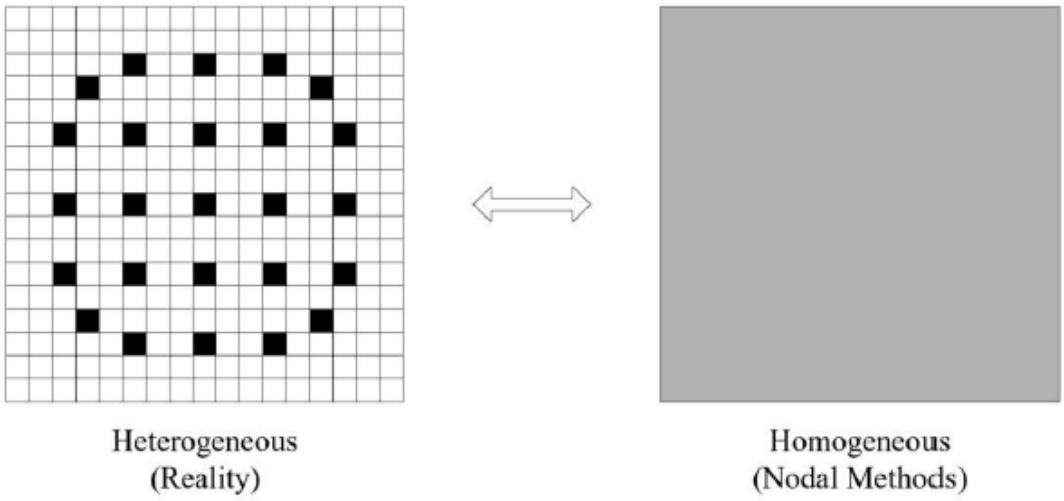

Figure 1.1: Example illustration of homogenization for an assembly [9]

1.3.2 Equivalence Theory

When using exact flux-weighted cross sections to preserve the reaction rates within the homogenized diffusion equation, there often still exist large errors. This is a direct result of the approximation used for the neutron leakage term and the definition of the diffusion coefficient.

Various techniques have investigated methods to formulate a more rigorous diffusion coefficient based on components of the heterogeneous response matrix elements [10] [11]. These methods are sound mathematically but are not widely used, due to the computational burden of constructing the response matrix.

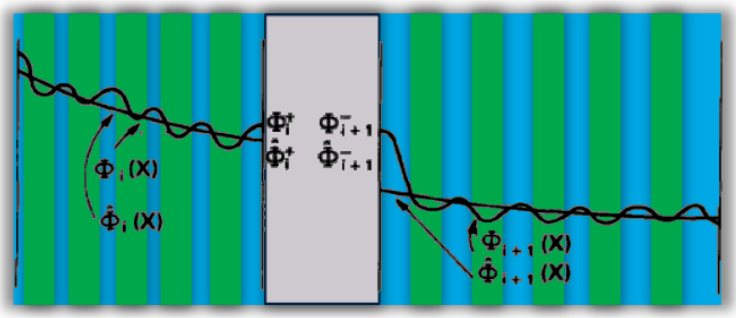

Figure 1.2: Heterogeneous and homogeneous solution to the two-node problem [12]

To illustrate the difficulty of enforcing equivalence on a system, consider the two-node, one-dimensional problem in Figure 1.2, in which the blue regions represent water and the green regions fuel. The curvy line represents the heterogeneous flux solution produced from a lattice calculation, and the straight lines represent the diffusion solution. If the left node, node i, is solved independent of either node adjacent to it with given currents for boundary conditions, then the flux distribution can be found exactly based on the generated group constants for that cell. The same can be done for any node with flux weighted cross sections and boundary currents. However, it should be noted that the diffusion solution on the boundary for node i and node i+1 are not continuous for this formulation. If instead the two-node problem is solved using the currents on the left of node i and the right of node i+1, then the diffusion solution will produce a continuous homogeneous flux at the interface surface. This solution will produce a homogeneous surface flux that is not equivalent to the heterogeneous surface flux, as well as a homogeneous surface current that does not reproduce the heterogeneous surface current. In this example, it is clear that the interface continuity conditions are the main source of error [12].

The solution to this problem is to allow the diffusion solution at the interface to be discontinuous, such that the heterogeneous surface flux is continuous and there is a direct relationship between the heterogeneous and homogeneous surface fluxes [12]. This is done through the use of a discontinuity factor defined in Eq. 1-22:

$$\widehat{\Phi}_i^+ f_i^+ = \widehat{\Phi}_{i+1}^- f_{i+1}^-, \tag{1-22}$$

where,

$$f_i^+ = \frac{\Phi_i^+}{\widehat{\Phi}_i^+}, \qquad f_{i+1}^- = \frac{\Phi_{i+1}^-}{\widehat{\Phi}_{i+1}^-}.$$

In these expressions the "+" indicates the right surface of a given cell and the "-" indicates the left surface of a given cell. The variable Φ represents the heterogeneous flux distribution and the variable $\widehat{\Phi}$ represents the homogeneous flux distribution within a given cell of index "i" (same as shown in Figure 1.2). This relationship also ensures that the interface currents are preserved. Conveniently, these discontinuity factors can be computed and stored just like typical homogenized parameters for use in nodal codes. The implementation of these values within the diffusion equations is discussed in Chapter 3.

One final note is that these constants are typically employed in the radial direction of most reactors. This is because the principal source of heterogeneity in LWRs is the radial arrangement of the fuel pins in an assembly. In fact, most fuel assemblies show little variation in the axial direction. Later in this chapter, the RBWR [6] design will be described, which features considerable axial heterogeneity compared to the radial direction. In this case, the concept of discontinuity factors is not brought about due to the homogenization process described above, but more to treat the boundary interface problem, where steep changes in the flux can occur. The use of axial discontinuity factors for this application is discussed in Chapters 2 through 4.

1.4 Monte Carlo Cross Sections

The Monte Carlo method has been used in the nuclear industry for many years, with one of its principal application in reactor analysis as a tool to validate deterministic calculations at steady-state conditions [13] [14]. With the advancement in computational resources, Monte Carlo is now being extended to additional applications, such as cross section generation and selected whole-core simulations often reserved to deterministic codes for their computational efficiency. The principal idea behind using Monte Carlo for cross section generation is that it can provide an accurate three-dimensional representation of the model of interest without making any simplifications to the Boltzmann neutron transport equation. This includes not only the spatial and angular treatments in deterministic methods, but also the approximate methods for resonance self-shielding used in the lattice codes to collapse cross sections. However, the drawback to Monte Carlo has always been the simulation time and relative uncertainty associated with the statistical parameters. Many but not all of these concerns have been alleviated due to advances in computing technology.

1.4.1 Monte Carlo Results

The interactions in Monte Carlo simulations are based on discrete events. This is different from deterministic calculations, where the equations are based on continuous variables. Information is *scored* for each interaction and provides a detailed analysis for the system. The most common calculation is a weighted flux integral such as the following:

$$R = \int_t dt \int_V d^3r \int_E dE\, f(\vec{r}, E)\phi(\vec{r}, E), \tag{1-23}$$

where the response function $f(\vec{r},\mathrm{E})$ can be any physical quantity that is known over the range of integration. A common quantity of interest is the flux-weighted cross section over a cell. This is found from the ratio of the desired reaction rate to the total flux:

$$\Sigma_n = \frac{\sum_{i=1}^{I_n} \Sigma^i \phi^i}{\sum_{i=1}^{I_n} \phi^i}. \tag{1-24}$$

Where *n* represents a cycle of simulated neutrons, I_n is the number of batches in that generation and *i* is an individual neutron score. For criticality simulations, the number of histories (or simulated neutrons) is broken up into cycles to determine the fission source. Since the distribution of neutrons within a system is often unknown at the beginning of a simulation, cycles are used to establish this shape. A specified number of inactive cycles, cycles before results are tallied, are discarded at the beginning of a simulation to first establish the fission source before any information is scored. Equation 1-24 provides an estimate of the cross section for each generation of neutrons. Once all cycles have been simulated, the cross section can be determined using the arithmetic mean [15] [16]:

$$\bar{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} \Sigma_n, \tag{1-25}$$

where the standard deviation of the mean, or uncertainty due to the random sampling process, is defined as [17]:

$$\sigma(\Sigma) = \sqrt{\frac{1}{\mathrm{N}(\mathrm{N}-1)}\left[\sum_{n=1}^{N} \Sigma_n^2 - \frac{1}{N}\left(\sum_{n=1}^{N} \Sigma_n\right)^2\right]}, \tag{1-26}$$

and the relative statistical error of a value is denoted by:

$$RSE(\Sigma) = \frac{\sigma(\Sigma)}{\bar{\Sigma}}. \tag{1-27}$$

The relative error of the product or ratio of two uncorrelated quantities is found using the individual calculated means and their respective relative errors:

$$\begin{gathered} z = xy, \\ or \\ z = \frac{x}{y}, \end{gathered} \tag{1-28}$$

$$\Delta z = z\sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2}, \tag{1-29}$$

where Δz represents the desired relative error of the variable z, and Δx and Δy are the relative errors of the variables x and y. This is useful for determining the relative uncertainty of a quantity based on two Monte Carlo results that have associated relative uncertainties.

This process can be performed for any number of cross sections and other quantities of interest. However, the diffusion coefficient is difficult to compute since there is no equivalent form in general transport theory. The following section discusses the calculation of diffusion coefficients in Monte Carlo.

1.4.2 Monte Carlo Diffusion Coefficients

As mentioned in the previous section, diffusion coefficients are a difficult quantity to determine using Monte Carlo. This is because there is no equivalent continuous-energy counterpart in general transport theory. Instead, this parameter is based on an approximation. Most of the techniques center on generating a transport cross section from which the diffusion coefficient is calculated. One of these methods is illustrated in Eq. 1-30 [18]:

$$\Sigma_{tr,g}(\vec{r}) = \frac{\int_g dE\phi(\vec{r},E)}{\int_g dE\,\dfrac{\phi(\vec{r},E)}{\Sigma_t(\vec{r},E) - \Sigma_{s1}(\vec{r},E)}}, \tag{1-30}$$

$$D_g = \frac{1}{3\Sigma_{tr,g}}. \tag{1-31}$$

This provides a definition for an isotropic diffusion coefficient. In these equations, g refers to the few-group index.

1.5 Physics of Fast Reactors

The principal motivation for the research performed here has been the difficulties of modeling a fast spectrum Boiling Water Reactor (BWR), in which the physics is very different

than the conventional thermal spectrum LWRs. One of the first light water fast reactors built in the U.S. was the Shippingport reactor in Pennsylvania in 1977 [19]. This reactor operated for five years and at the end of its operation had generated a 1.39 percent greater fissile inventory compared to its initial loading. This reactor was specifically aimed at breeding fuel using a fast spectrum. However, fast reactors are useful for both breeder and burner type reactors. The atomic interactions that allow these reactors to operate are discussed in the following sections.

1.5.1 Breeder Reactor Physics

In breeder designs, the amount of fissile material produced meets or exceeds the amount of fissile material burned. To illustrate this concept it is useful to look at plots of the fission and capture cross sections for Uranium-235, Uranium-238 and Plutonium-239.

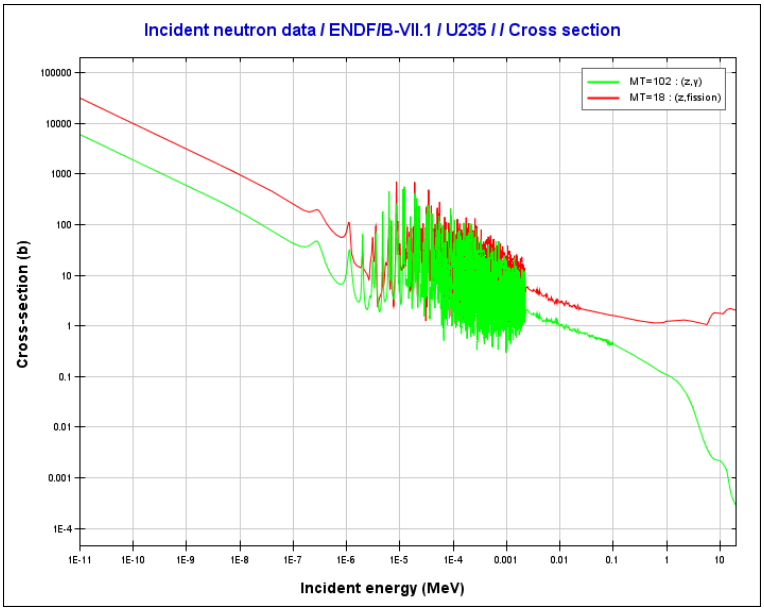

Figure 1.3: Capture and fission cross section for U-235 [20]

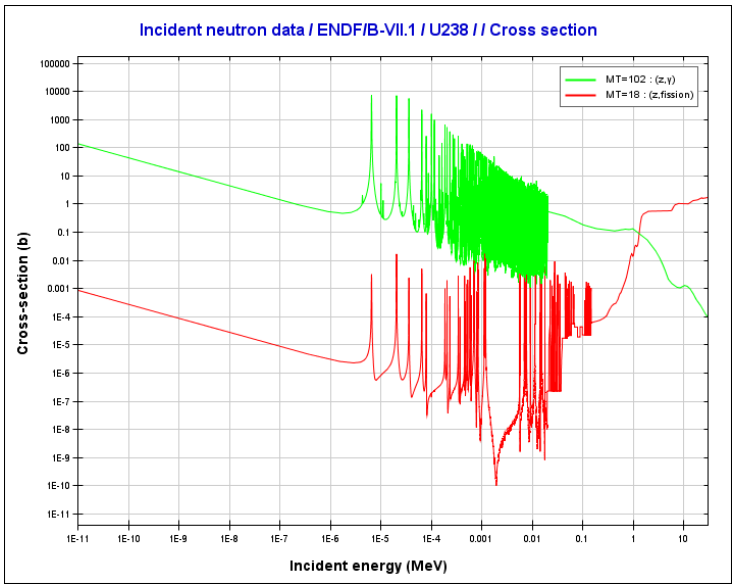

Figure 1.4: Capture and fission cross section for U-238 [20]

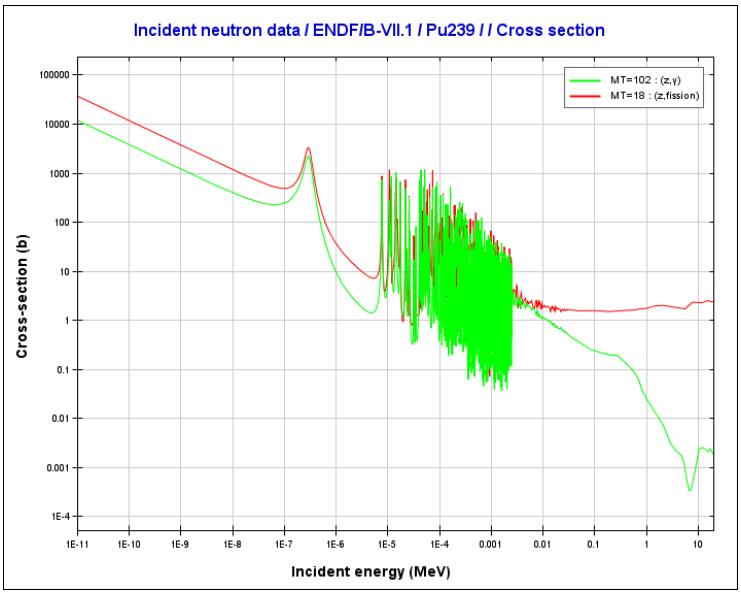

Figure 1.5: Capture and fission cross section for Pu-239 [20]

For high energies, there is an increased ratio of fission to capture for both U-235 and Pu-239. This is due to a decrease in the capture cross section for both isotopes as the energy of an incident neutron increases while the fission cross section shows little variation. Because of these higher fission-to-capture cross sections, a system based on higher-energy neutrons is possible. It is also important to note that the fission cross section for U-238 increases rapidly around 100 keV. The effects of the increase in the ratio of fission-to-capture and the increase in

the fission cross section of U-238 allow a fast spectrum system to maintain a steadier neutron flux than in a thermal spectrum. This is especially important due to the increase in leakage introduced by operating in a fast spectrum. One other important note about U-238 is its ability to capture and convert into Pu-239. All of these properties make these three isotopes ideal for operating a reactor based on high energy neutrons. A breeder system can be devised by introducing a hard enough spectrum and an appropriate fuel composition such that the system can maintain criticality while promoting a high conversion of fertile to fissile fuel. This is typically done through the conversion of U-238 to Pu-239. However, other reactors based on a thorium fuel cycle have been developed.

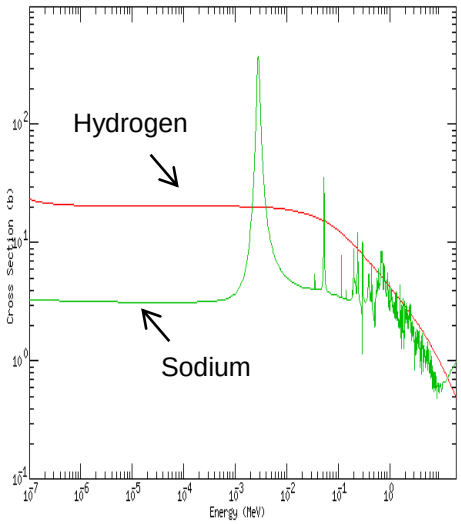

Figure 1.6: Total cross section for hydrogen and sodium

Critical to all fast breeder reactors is hardening the neutron spectrum in order to promote breeding. Most designs have centered on the use of liquid metal cooling, due to its lower cross section compared to hydrogen in light water reactors [21] [22] [23]. An example plot of this can be seen in Figure 1.6 which compares the total cross section of hydrogen and sodium. The lower cross section reduces the probability the coolant will moderate the neutrons in the system. Additionally, the maximum energy lost per collision is larger for light nuclei such as hydrogen. Therefore a heavier nuclide such as sodium is desired for the coolant in order to maintain the hard spectrum.

1.5.2 Physics of Burner Reactors

Similar to breeder reactors, burner reactors depend on a hard neutron spectrum to fission long-lived transuranic elements produced from nuclear reactions. One of the major issues with nuclear energy is the disposal of waste products generated during the chain reaction. Transuranic elements such as Neptunium, Plutonium, Americium and Curium contain isotopes whose half-lives last from hundreds of thousands of years to millions of years (Table 1.1). These pose a significant challenge when designing a method to dispose of this long-lived waste. However, if these isotopes are dealt with either by removal or through fission, then the storage length of nuclear waste could be greatly reduced to hundreds of years.

Table 1.1: Half-lives of important transuranic elements

Element	Isotope	Half-life (years)
Np	237	2.144E6
Pu	238	8.774E1
Pu	239	2.410E4
Pu	240	6.500E3
Pu	241	1.400E1
Pu	242	3.730E5
Am	241	4.322E2
Am	242m	1.410E2
Am	243	7.370E3
Cm	243	2.910E1
Cm	244	1.810E1
Cm	245	8.500E3
Cm	246	4.730E3
Cm	247	1.560E7
Cm	248	3.400E5

Burner reactors operate by fissioning these transuranic isotopes. Consider the fission cross sections for Np-237, Am-241 and Pu-242 displayed in Figure 1.7-Figure 1.9. All three of these isotopes see significant increases in their fission cross section around 1 MeV. By operating a reactor in a fast spectrum, the probability of inducing fission in these isotopes greatly increases. This achieves two goals by destroying the transuranic elements while producing fission to maintain the chain reaction.

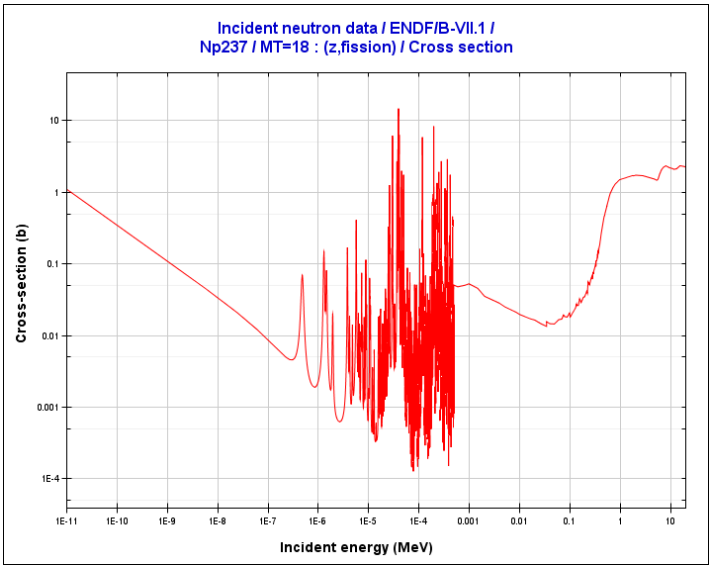

Figure 1.7: Fission cross section for Np237 [20]

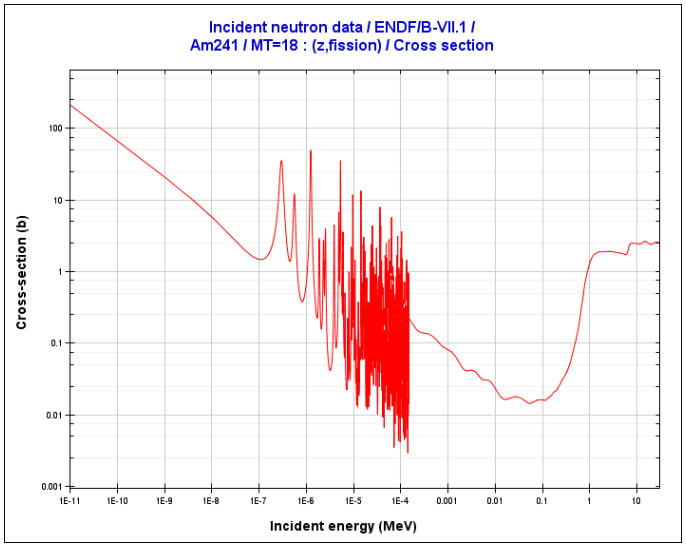

Figure 1.8: Fission cross section for Am241 [20]

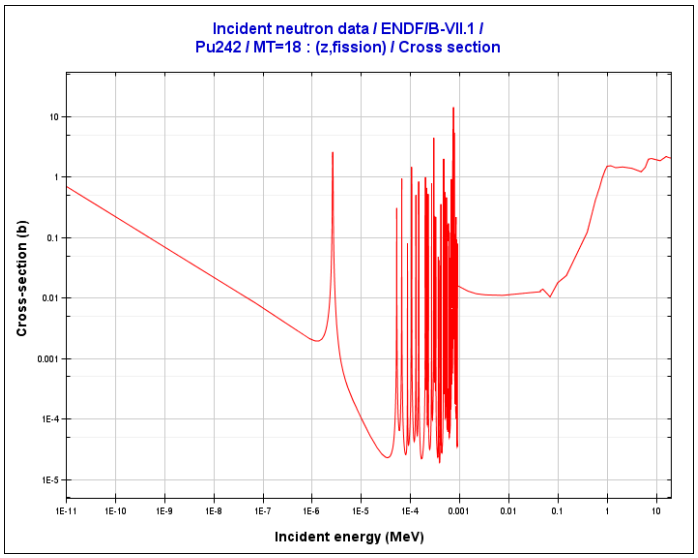

Figure 1.9: Fission cross section for Pu242 [20]

This section discussed the fundamentals of fast reactors. The next section will discuss a new and innovative concept for a breeder and a burner reactor that utilizes existing LWR technology.

1.6 RBWR Design

Though most breeder and burner reactors are designed to use metal coolant, several reduced moderation LWRs have also been developed [24] [25]. One such design is the Resource Renewable Boiling Water Reactor (RBWR) by Hitachi [6]. Relative to most BWRs (and Advanced Boiling Water Reactor (ABWR) [26] [27]), the RBWR core design features a tighter fuel lattice, a shorter core, a smaller coolant mass flow-rate and pressure drop, a larger exit void fraction, and a less negative core void reactivity coefficient. The design of the RBWR is shown in Figure 1.10, and some of the characteristics of the RBWR and the ABWR are compared in Table 1.2. As indicated in the Table, the RBWR has a lower coolant flow rate, higher void fraction, and less negative void coefficient compared to the ABWR. The void coefficient represents the amount of reactivity inserted into the system as the coolant void changes. If the void coefficient is negative, then a negative reactivity feedback is introduced as the coolant void increases. However, if the void coefficient is positive, then a positive reactivity feedback is introduced as the coolant void increases. All reactors in the United States require a negative void coefficient for operation.

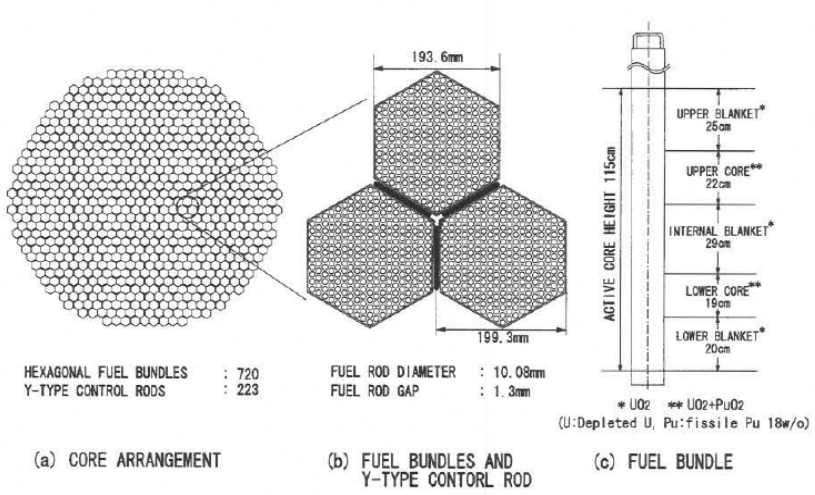

Figure 1.10: Core design of the RBWR [28]

Table 1.2: Design characteristics of the RBWR compared to the ABWR [28]

Item		RBWR-AC	Item		ABWR
Thermal Power	MWt	3926	Thermal Power	MWt	3926
Electrical Power	MWe	1356	Electric Power	MWe	1356
No. of Fuel Bundles		720	Number of Fuel Bundles		872
Core Height	mm	1200	Core Height	mm	3710
Configuration for Height		Parfait			
Coolant Flow Rate	kt/h	22	Coolant Flow Rate	kt/h	58
Core Exit Quality	%	41	Core Exit Quality	%	13
Void Fraction	%	60	Void Fraction	%	36
Pressure Drop	MPa	0.11	Core Pressure Drop	MPa	0.21
HM Inventory	t	131	Uranium Inventory	t	151
Puf/HM in Fissile Zone	w/o	18	Uranium Enrichment	wt%	3.8
Puf Inventory	t	8.9			
Burnup	GWd/t	45	Burnup	GWd/t	45
MLHGR	kW/ft	14.4	MLHGR	kW/ft	12
MCPR		1.3	MCPR		1.3
Void Coef.	△k/k/%void	$-2X10^{-5}$	Void Coefficient	$10^{-4}\frac{\Delta k/k}{\%void}$	$-12X10^{-5}$

The less negative void coefficient in the RBWR is because of the harder (higher-energy) neutron spectrum, resulting from the higher void fraction and the tighter lattice. In general, fast spectrum reactors have the potential for a positive core void coefficient, primarily because the spectrum is already hard and core voiding can result in an even harder spectrum and a corresponding increase in the fast fission probability of ^{238}U and in η of the TRU isotopes. The spectrum of the RBWR is compared to the LWR (thermal spectrum) and ARR (hard spectrum) in Figure 1.11. One of the principal reasons for the much harder spectrum of the RBWR is the large reduction in the Hydrogen scattering cross section at high energies which is also shown in Figure 1.11.

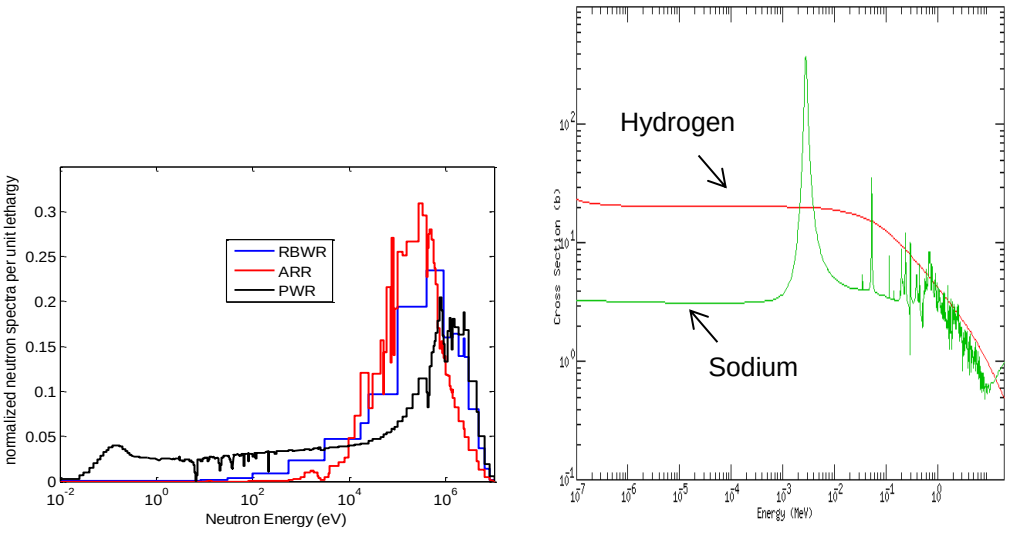

Figure 1.11: Comparison of the spectra of the RBWR, ARR (SFR), and LWR

In order to mitigate the potential for a positive void coefficient, most designs of fast spectrum cores have introduced innovative features, such as large internal blanket regions in order to provide lower importance neutron sinks in the event of core voiding during a transient. The core void and power distribution of the RBWR were previously calculated using RELAP5/PARCS (Purdue Advanced Reactor Core Simulator) and are shown in Figure 1.12 and Figure 1.13 [29]. As indicated, the power in the internal blanket region is significantly lower than the power in the upper and lower fissile regions of the core. The positive reactivity effects of increased neutron production in the fissile regions during voiding would be mitigated by leakage of neutrons into the lower-importance blanket region. There have been numerous sodium fast reactor (SFR) designs which have successfully achieved negative void coefficient values with axial blankets of natural uranium [30]. However, the designers of these reactors have pointed out that achieving a negative void coefficient with large internal blankets can lead to core designs that may be less safe, because they introduce other safety issues. Past work focused on examining the transient safety performance of the RBWR using the U.S. Nuclear Regulatory Commission (NRC) TRACE/PARCS codes [31].

Probably the most unique feature of the RBWR is the large heterogeneity in the axial direction relative to the radial direction. This is illustrated in the axial power profile, where there are significant increases and decreases dependent upon the material region. However,

the radial power distribution shows little to no variation. It is important to emphasize this point because typical LWRs display much more heterogeneity in the radial direction compared to the axial. The axial heterogeneity of this reactor is the motivating factor behind the research discussed in the following chapters. Using typical diffusion theory to solve this problem has significant issues; therefore a more accurate (higher-order) method is proposed to handle this axial treatment.

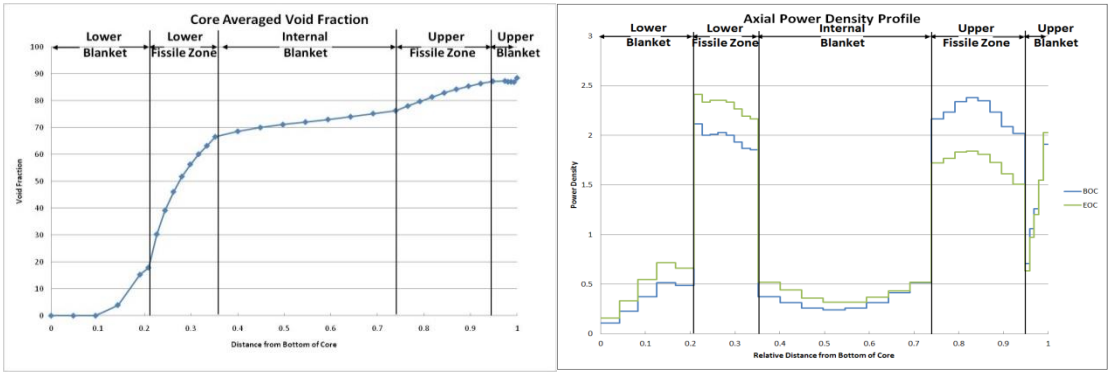

Figure 1.12: Axial core void and power distributions in the RBWR [29]

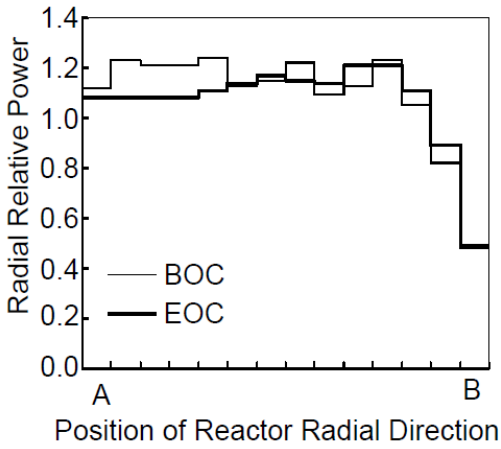

Figure 1.13: Radial power distribution of the RBWR [28]

1.7 Organization of Dissertation

The first chapter discussed some of the fundamental theory that will be used in the remainder of this dissertation. In the next chapter, the current methodology used to solve

axially heterogeneous cores is discussed. An overview of the code system along with assembly and core results are provided. The purpose of this chapter is to illustrate the difficulty of using diffusion theory for problems such as the RBWR and the error that it introduces.

For the following chapter, various higher order Pn methods and Quasidiffusion (QD) are explored to solve the one-dimensional axial problem. The use of discontinuity factors in the P1, P2, P3 and Quasidiffusion equations are demonstrated, along with a mesh refinement scheme. In addition, a scheme for additive discontinuity factors is proposed for each of the methods.

For chapter 4, numerical results for an RBWR-type assembly consisting of five separate axial regions is presented. Eigenvalue comparisons and flux distributions for each approximation are provided along with comparisons of iterations and total run time. The effects of bounding the axial discontinuity factor are investigated along with the magnitudes of the discontinuity factors.

In the final chapter of this dissertation, concluding remarks are provided, along with suggestions for future work to improve upon the introduced methods.

Chapter 2

Core Neutronics with 3-D Monte Carlo Cross Sections

As noted in the previous chapter, the principal heterogeneity in a typical LWR is the radial arrangement of fuel pins in the assembly lattice. This makes it possible to use 2-D homogenized cross sections to simulate the full range of anticipated reactor conditions. However, the axial heterogeneity of the RBWR has motivated the development of Monte Carlo methods to generate 3-D homogenized cross sections for core analysis. This chapter describes the development of 3-D Monte Carlo cross sections and their implementation within the framework of the U.S. NRC code system based on the nodal core simulator PARCS [32]. This chapter describes the overall code package and its various components used for full-core calculations. Both assembly level and full-core calculations are presented to demonstrate the effectiveness of 3-D cross sections, but also to illustrate the need for higher-order transport methods when analyzing axially heterogeneous cores.

2.1 Core Simulator

The computer code developed in this research for full core analysis of the RBWR is depicted in Figure 2.1. It consists of the lattice code Serpent [17], the post-processing code GenPMAXS [33], and the nodal diffusion code PARCS [32] which is coupled to the thermal-hydraulics code PATHS (PARCS Advanced Thermal Hydraulic Solver) [34]. Each of these codes is discussed in further detail in the following sections.

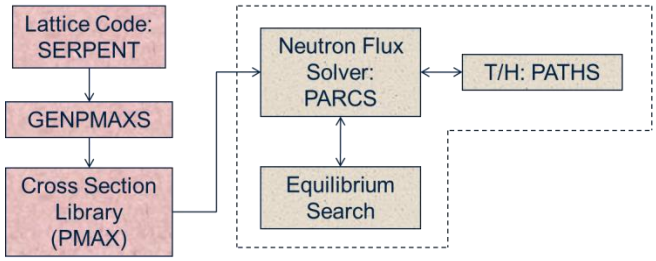

Figure 2.1: Core simulator diagram

2.1.1 3-D Monte Carlo Cross Sections

2.1.1.1 Serpent Monte Carlo Code

In the methods developed in this work, the homogenized, few-group cross sections are generated using the three-dimensional continuous-energy Monte Carlo reactor physics burnup code Serpent [17]. This code utilizes delta tracking to accelerate the calculation, while generating group-wise homogenized parameters. These group constants are collapsed based on ENDF/B-VII continuous-energy data [20]. The reason for using Serpent is due to its development team and broad user group, whose work has aimed at specifically generating Monte Carlo group constants in a convenient format. There is a second version of the Serpent code (Serpent 2), but this code has been in beta testing and is not explored in this document.

Like most other Monte Carlo codes, Serpent uses a universe-based combinatorial solid geometry model. Universes consist of either a lattice or a collection of cells that enables the complexity of the geometry to be divided into smaller parts. These universes can be used repeatedly to simplify the construction of the problem geometry. This allows a user to describe practically any conceived two or three-dimensional problem. The code tracks the geometry through the use of universes which are composed of cells. Each cell describes a particular region, often material differences, and is described through surface cards that represent the boundaries. By specifying a particular universe in the input, a user can generate group constants for the region of interest. This can be done for multiple universes in a single calculation. These group constants are tallied based on interactions within the specified region. The data that governs these reactions is based on ENDF/B-VII ACE (short for A Compact ENDF)

data files, which include S(α,β) and unresolved resonance tables. ACE files contain processed data from the raw evaluated ENDF data for the use in Monte Carlo codes. They contain tabular information for cross sections at different energies for various isotopes and can be created using codes such as NJOY. The S(α,β) tables contain information regarding neutron thermalization for neutron scattering at energies typically less than 4 eV. This includes relations for inelastic scattering and sometimes elastic scattering. Unresolved resonance tables are included within the ACE data files and can be used to sample cross sections where the resonances are so close together that they are "unresolved". This is important for isotopes such as ^{238}U, which can cause significant increases in reactivity if there are high fluxes in the unresolved region. Additional data sets including JEF-2.2, JEFF-3.1, JEFF-3.1.1 and ENDF/B-VI.8 are also available [35] [36] [37] [38] [39]. Serpent also features a built-in depletion module instead of the typical coupled methodology. The Bateman equations are solved using the Chebyshev Rational Approximation Method (CRAM) [40] along with a predictor-corrector time stepping. Additional information such as surface quantities can be generated through tally cards.

One of the advantages of using a Monte Carlo code for cross section generation is the ability to capture axial streaming. This is accomplished using three-dimensional modeling. However, traditional methodology is based on a two-dimensional approach. The following sections outline some of the nuances involved with generating three-dimensional cross sections.

2.1.1.2 Branch Methodology

The generation of cross sections involves defining a set of group constants for various state conditions that are representative of the steady-state and transient problems being solved. Within typical two-dimensional deterministic codes, this is performed by uniformly perturbing properties such as fuel temperature, coolant temperature, coolant density and control rod position. This type of perturbation is referred to as a branch calculation. It is important to capture the group constants for various depletion points, which are referred to as a history calculation. At each depletion point, the branch calculations are repeated to capture all possible scenarios. An illustration of this layout can be seen in Figure 2.2. The combination of

these history and branch calculations composes the cross section set used in most nodal codes. This process is well defined and is used in lattice codes such as HELIOS, CASMO and WIMS [2] [8].

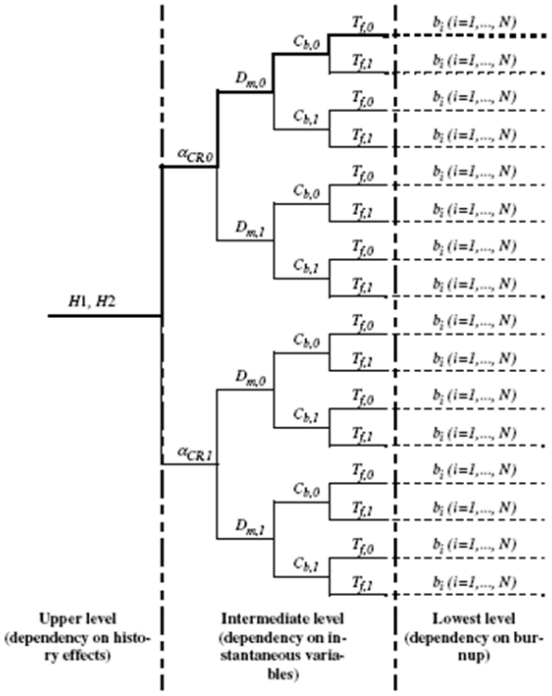

Figure 2.2: PMAXS history and branch structure [41]

The application of this method is similar for three-dimensional cross sections, but with a few key differences. For two-dimensional lattice simulations, state conditions can be uniformly perturbed. With a three-dimensional calculation for a BWR, the coolant density varies as a function of height. Therefore, if one axial region is perturbed, then to maintain a physical void distribution all axial regions must be perturbed. All state variables in a BWR are a function of height, which introduces added complexity to the simulation. To test the impact of this axial variation, a history and branching scheme was developed based on previous experience modeling the RBWR [31]. A small study was performed at Brookhaven National Lab (BNL) that illustrated the effect on the spectrum when perturbing individual regions as opposed to all regions simultaneously (integral method) [42]. A comparison of the spectrum for the upper fissile region is shown in Figure 2.3. From this analysis, significant changes in the spectrum were observed when perturbing the coolant void for individual nodes. The same study was

performed for the spectrum when perturbing the fuel temperature and showed minimal changes. Therefore a system was devised to perturb the entire axial void distribution for branching calculations to maintain a physical distribution within the assembly.

The method used in the work here involved generating an initial set of 2-D Serpent cross sections, which was used in a single assembly PARCS simulation coupled with PATHS to generate an initial guess for the void distribution. PATHS is a drift flux thermal-hydraulics code that solves the mass, momentum and energy equations. This void distribution was then placed within a 3-D Serpent calculation to generate a set of 3-D Serpent cross sections. Using these group constants, a second set of coupled PARCS/PATHS simulations, for a single assembly, were performed to generate the final void distribution to be used for each branch and history. This technique provides a coolant void distribution that is closer to that of the physical system compared to a uniform distribution.

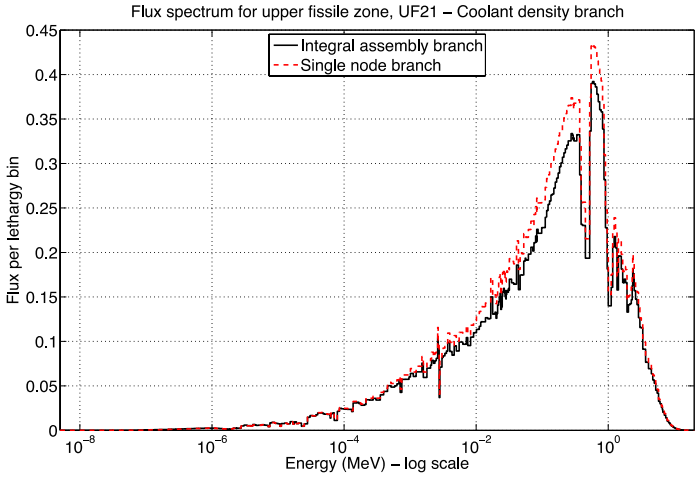

Figure 2.3: Flux spectrum for upper fissile zone [42]

2.1.1.3 Axial Discontinuity Factor Generation

The use of 3-D cross sections for the RBWR was first investigated at the Massachusetts Institute of Technology (MIT) in 2011 [43]. During that study, it was found that 3-D cross sections alone were not sufficient to reproduce a similar 3-D Monte Carlo solution. This led to the creation of axial discontinuity factors (ZDF), which are similar to the conventional assembly

discontinuity factors (ADF), except for the axial direction [44] [45]. For a given axial interface, a ZDF is defined as the ratio of the heterogeneous surface flux to the homogeneous surface flux:

$$f = \frac{\phi_{s,i,g}^{het}}{\phi_{s,i,g}^{\mathrm{hom}}}, \tag{2-1}$$

where f is the ZDF, $\phi_{s,i,g}^{\mathrm{hom}}$ is the homogeneous surface flux and the heterogeneous surface flux is approximated using the partial currents from Serpent:

$$\phi_{s,i,g}^{het} = 2\left(J_{i,g}^{+} + J_{i,g}^{-}\right), \tag{2-2}$$

where $J_{i,g}^{+}$ and $J_{i,g}^{-}$ are the partial incoming and outgoing currents for a given surface.

The homogeneous surface flux is found by solving the one-dimensional axial diffusion equation using the nodal expansion method (NEM) for a single node with the calculated group constants. The homogeneous surface flux is solved using the same method as PARCS (NEM for this case) in order to reproduce the Monte Carlo solution. The NEM approximates the flux solution within each mesh region using a fourth-order Legendre polynomial:

$$\vec{\phi}(\xi) = \sum_{i=0}^{4} \vec{a}_i P_i(\xi), \tag{2-3}$$

where $\vec{a}_i$ are the expansion coefficients and $P_i(\xi)$ are the Legendre polynomials.

The five coefficients associated with the flux expansion derive from the heterogeneous cell average flux, the net currents on the top and bottom surfaces and two weighted residual equations. The homogeneous surface fluxes for the top and bottom surfaces are found using the evaluated coefficients:

$$\vec{a}_0 = \bar{\phi}, \tag{2-4}$$

$$\left(\bar{\bar{A}} + \frac{5\bar{\bar{D}}}{2h^2}\right)\vec{a}_1 = -\frac{5}{4h}\left(\vec{J}^B + \vec{J}^T\right), \tag{2-5}$$

$$\left(\bar{\bar{A}} + \frac{21\bar{\bar{D}}}{2h^2}\right)\vec{a}_2 = -\frac{7}{4h}\left(\vec{J}^B - \vec{J}^T\right), \tag{2-6}$$

$$\vec{a}_3 = -\frac{h}{12\bar{\bar{D}}}\left(\vec{J}^B + \vec{J}^T\right) - \frac{1}{6}\vec{a}_1, \tag{2-7}$$

$$\vec{a}_4 = \frac{h}{20\bar{\bar{D}}}\left(\vec{J}^B - \vec{J}^T\right) - \frac{3}{10}\vec{a}_2, \tag{2-8}$$

$$\vec{\phi}_{i,g}^{B,Hom} = \vec{a}_0 - \vec{a}_1 + \vec{a}_2 - \vec{a}_3 + \vec{a}_4, \tag{2-9}$$

$$\vec{\phi}_{i,g}^{T,Hom} = \vec{a}_0 + \vec{a}_1 + \vec{a}_2 \mp + \vec{a}_4, \tag{2-10}$$

where

$$\bar{\bar{A}} = \begin{bmatrix} \Sigma_{t,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{t,G} \end{bmatrix} - \begin{bmatrix} \Sigma_{s,1,1}^0 & \cdots & \Sigma_{s,1,G}^0 \\ \vdots & \ddots & \vdots \\ \Sigma_{s,G,1}^0 & \cdots & \Sigma_{s,G,G}^0 \end{bmatrix} - \frac{1}{k}\begin{bmatrix} \chi_1 \\ \cdots \\ \chi_G \end{bmatrix}\begin{bmatrix} \nu\Sigma_{f,1} \\ \cdots \\ \nu\Sigma_{f,G} \end{bmatrix}^T,$$

$$\bar{\bar{D}} = \begin{bmatrix} D_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_G \end{bmatrix}.$$

The matrix $\bar{\bar{A}}$ consists of the diagonal matrix of the group-wise total cross section, the full matrix of the group-to-group scattering cross section, the eigenvalue, the vector of the group-wise fission spectrum and the transposed vector of the group-wise product of the average number of fission neutrons produced per fission and the fission cross section. $\bar{\bar{D}}$ is a diagonal matrix of the group-wise diffusion coefficients and h is the height of cell i. The vectors $\vec{J}^B$ and $\vec{J}^T$ represent the bottom and top neutron currents respectively, and $\vec{\phi}_{i,g}^{B,Hom}$ and $\vec{\phi}_{i,g}^{T,Hom}$ represent the bottom and top homogeneous surface fluxes respectively. The axial discontinuity factors are included in the PMAXS cross section file for the top and bottom surfaces of a given material node. A separate cross section file is created for each nodal region to accommodate the axial discontinuity factors and the 3-D cross sections.

For interfaces with large gradients, such as the region between seed and blanket zones, the homogeneous flux can become negative, which leads to a negative discontinuity factor. This can result in negative fluxes within PARCS and numerical problems during the flux solution. A scheme was developed to avoid this, based on modifying the diffusion coefficient such that the axial discontinuity factor would be bounded within an acceptable range, while still preserving the net current on the node interface [31]. However, the modification of the diffusion

coefficient also affected the radial 2-D calculation within TPEN, causing instabilities within the core calculation. Instead, limits were placed on the axial discontinuity factors. If the calculated value exceeded the specified range, then the quantity was changed to the closest bound. Though this does not preserve neutron balance, it ensures stability for the simulation while introducing only a small amount of error. More detail on the error introduced is discussed later in this chapter.

2.1.1.4 Monte Carlo Tallies

The Serpent v1.1.19 Monte Carlo code is capable of generating cell-averaged fluxes and surface currents. For higher-order methods such as P3, additional angular moments of the flux are required. A couple of tallies were implemented into the Serpent code to calculate the higher order moments required for the P3 analysis. These include surface angular flux moments as well as the average angular cell flux moments. For the surface angular flux moments, each particle traversing the surface is weighted by the Legendre moments and divided by the cosine of the particle's angle with respect to the surface normal [46].

$$\psi_n^s = \sum_{i=1}^{I} \frac{W_i P_n(\mu_i)}{\mu_i}, \tag{2-11}$$

where W_i represents the weight of particle i of I total particles in a simulation, $P_n(\mu_i)$ is the nth-order Legendre polynomial of the scattering cosine (μ_i), and ψ_n^s is the nth-moment surface flux.

However, it is important to note that for the zeroth moment as the cosine of the scattering angle approaches zero the surface moment approaches infinity. This leads to large quantities and erratic uncertainties for the surface flux. Additional detail on how this was treated can be found in Appendix A. To determine the average angular cell flux moments, we perform a similar process but no longer weight by the cosine of the scattering angle:

$$\psi_{n,i} = \sum_{i=1}^{I} W_i P_n(\mu_i). \tag{2-12}$$

Here $\psi_{n,i}$ represents the nth-moment cell-averaged flux, and all other values are the same as in Eq. 2-11. Equations 2-11 and 2-12 provide the heterogeneous values necessary to check neutron balance. The surface quantities are also used to generate the heterogeneous surface moments for discontinuity factor calculations.

2.1.2 GenPMAXS

The post-processing code GenPMAXS serves as an interface between lattice physics codes and the PARCS core simulator [33]. It converts the raw data produced from Serpent into PMAXS format, which provides all of the necessary data to perform core simulation for steady-state and transient applications. This includes macroscopic cross sections, discontinuity factors and kinetics parameters. It supports lattice physics codes such as HELIOS, CASMO, TRITON, Serpent and Serpent 2. The code is also capable of calculating discontinuity factors as well as checking neutron balance. If the neutron balance is not preserved due to statistical error, GenPMAXS can adjust the absorption cross section. An example of this for a one-speed problem is shown in Eq. 2-13:

$$\Sigma_{a,i} = \frac{\frac{1}{k}\nu\Sigma_{f,i}\psi_{0,i}\Delta x_i - \left(\psi_{1,i+1/2} - \psi_{1,i-1/2}\right)}{\psi_{0,i}\Delta x_i}, \tag{2-13}$$

where ψ_0 represents the scalar flux in node *i* and ψ_1 are the surface currents on either side of the node. All of these quantities come from the lattice code. A similar process can be performed to ensure balance for the second moment P3 equation.

Section 2.1.1.2 included a discussion on the generation of branch and history variables. All of this information must be stored within the PMAXS files. The first step involves storing all of the cross sections generated at different history conditions. These conditions are referred to as the reference branches, and all information is explicitly stored for all depletion points. Since core conditions can vary, the branching information is also stored within the PMAXS files. This is done through storing the partial derivative of the branch values relative to the reference branch. The method used within GenPMAXS and PARCS involves second order central differencing:

$$\left.\frac{\partial \Sigma}{\partial x_k}\right|_{X^m} = \frac{\Sigma\left(X^i\right) - \Sigma\left(X^{B(i)}\right)}{x_k^i - x_k^r}, \tag{2-14}$$

where X^i is the vector for the variables of state i, $X^{B(i)}$ is the vector for the variables of the base state of i, X^m is the vector for the mid-point between state i and its base state and x_k is the branch variable of state i. The lower case r represents the reference value for the state variables. For control rod calculations, the denominator is replaced with the partial control rod insertion fraction instead of the difference. Partial derivatives are calculated for each branch case and stored in that section within the cross section file.

To calculate the cross sections for a particular state during a nodal simulation, consider the following example of a density coolant branch, as discussed in the GenPMAXS manual shown in Figure 2.4.

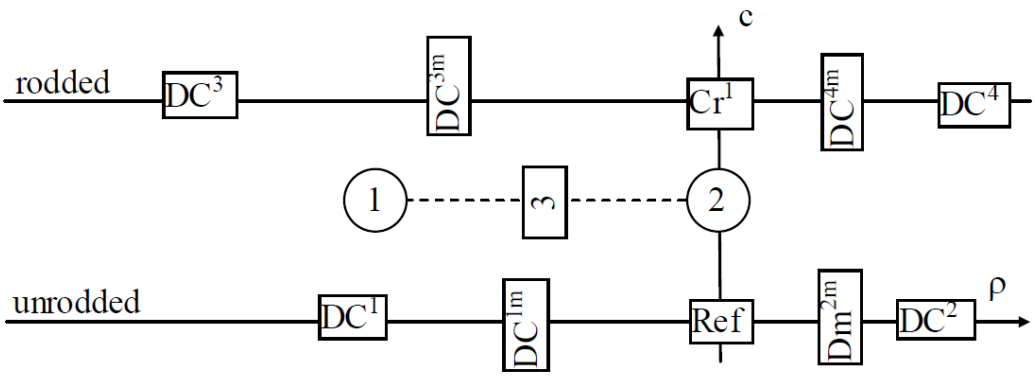

Figure 2.4: Example of computing a cross section at point 1 [33]

The horizontal boxes represent lattice simulations performed at various state conditions. In the figure, “Ref” represents the reference calculation, “Cr” the control rod inserted and the “DCs” as changes in the coolant density. The vertical boxes represent the partial derivatives stored for each of the branches. If the cross section at point 1 is desired, Figure 2.4 shows that no branch calculation was performed for this location. Instead, the cross section can be computed based on the available data from the reference point and the five branches using the following relation:

$$\Sigma(c, DC) = \Sigma^r + c\frac{\partial\Sigma}{\partial Cr}\Big|_{Cr^1/2} + (DC - DC^r)\frac{\partial\Sigma}{\partial DC}\Big|_{c,DC^m}. \quad (2\text{-}15)$$

In this expression, Σ_r is the cross section at the reference state, the second term is the contribution from insertion of the control rod, and the third term is the contribution from the change in coolant density. The control rod term is a simple interpolation between the Ref and Cr positions based on the depth the control rod is inserted and is depicted as point 2. However, the density coolant contribution is a function of the four branches and represents the remaining contribution depicted as point 3:

$$\frac{\partial\Sigma}{\partial DC}\Big|_{c,DC^m} = w_1\frac{\partial\Sigma}{\partial DC}\Big|_{0,DC^{1m}} + w_2\frac{\partial\Sigma}{\partial DC}\Big|_{0,DC^{2m}} + w_3\frac{\partial\Sigma}{\partial DC}\Big|_{1,DC^{3m}} + w_4\frac{\partial\Sigma}{\partial DC}\Big|_{1,DC^{4m}}, \quad (2\text{-}16)$$

where the weights for the four points are found from linear interpolation:

$$w_1 = (1-c)\frac{DC - DC^2}{DC^1 - DC^2}, \qquad w_2 = (1-c)\left(1 - \frac{DC - DC^2}{DC^1 - DC^2}\right),$$
$$w_3 = c\frac{DC - DC^4}{DC^3 - DC^4}, \qquad w_4 = c\left(1 - \frac{DC - DC^4}{DC^3 - DC^4}\right).$$

A similar procedure can be applied to calculate cross sections that are dependent upon burnup. The cross sections at a given depletion point are found using piece-wise linear interpolation between the two nearest tabulated burnup points. Further details can be found in the GenPMAXS manual [33].

2.1.3 PARCS

After GenPMAXS produces the PMAXS cross section files, this data is used within the core simulator PARCS. PARCS [32] is a three-dimensional reactor core simulator, which solves steady-state and time-dependent, multi-group neutron diffusion and SP3 transport equations in Cartesian, cylindrical, and hexagonal geometries [32]. PARCS is coupled directly to the thermal-

hydraulics code PATHS [34] (for equilibrium cycle simulation) and TRACE [47] (for transient simulation), which provide the temperature and flow field information to PARCS. The major calculation features in PARCS include the ability to perform eigenvalue calculations, transient (kinetics) calculations, Xenon transient calculations, decay-heat calculations, pin-power calculations, and adjoint calculations for commercial LWRs.

One of the most important features of the PARCS code is its ability to solve the eigenvalue problem. For a steady-state calculation, the multigroup diffusion equation is discretized to form the following linear system:

$$\bar{\bar{M}}\bar{\phi} = \lambda\bar{\bar{F}}\bar{\phi} = \frac{1}{k_{eff}}\bar{\bar{F}}\bar{\phi}, \tag{2-17}$$

where $\bar{\bar{M}}$ represents the migration matrix that contains all the nonfission terms and $\bar{\bar{F}}$ is the fission matrix that consists of the fission terms. The term k_{eff} represents the largest criticality eigenvalue. This problem can be solved using the fission source iteration method, which involves the solution of the following source problem:

$$\bar{\bar{M}}\bar{\phi}^{\mathrm{n+1}} = \frac{1}{k_{eff}^{n}}\bar{\bar{F}}\bar{\phi}^{n}, \tag{2-18}$$

where n represents the index of the previous iteration, and n+1 the index of the next iteration. To accelerate this calculation, PARCS uses the Wielandt shift method [48]. This technique involves subtracting a fission source term from both sides of Eq. 2-18:

$$\left(\bar{\bar{M}} - \frac{1}{k_s}\bar{\bar{F}}\right)\bar{\phi} = \left(\frac{1}{k_{eff}} - \frac{1}{k_s}\right)\bar{\bar{F}}\bar{\phi}, \tag{2-19}$$

where,

$$k_s = k_{eff} + \delta k.$$

By doing this, the fission source iteration changes form to the following:

$$k_A^n \bar{\psi}^{n+1} = \bar{\bar{F}} \bar{\bar{A}}^{-1} \bar{\psi}^n, \tag{2-20}$$

where

$$\bar{\bar{A}} = \bar{\bar{M}} - \frac{1}{k_s^n} \bar{\bar{F}}, \qquad \bar{\psi}^n = \bar{\bar{F}} \bar{\phi}^n, \qquad and$$

$$\frac{1}{k_A^n} = \frac{1}{k_{eff}^n} - \frac{1}{k_s^n}.$$

The iteration scheme used for this method corresponds to the power method [49], which provides an updated eigenvalue through the following expression:

$$k_A^{n+1} = k_A^n \frac{\langle \psi^{n+1}, \psi^{n+1} \rangle}{\langle \psi^{n+1}, \psi^n \rangle}. \tag{2-21}$$

Using these expressions, a derivation for the next l-th iteration of the eigenvalue can be defined:

$$\frac{1}{k_A^{n+1}} = \frac{1}{k_{eff}^{n+1}} - \frac{1}{k_s^n} = \frac{\gamma}{k_A^n}, \tag{2-22}$$

where

$$\gamma = \frac{\langle \psi^{n+1}, \psi^n \rangle}{\langle \psi^{n+1}, \psi^{n+1} \rangle}.$$

Equation 2-22 is used to solve for a new estimate of k_{eff}:

$$k_{eff}^{n+1} = \left[\frac{\gamma}{k_A^n} + \frac{1}{k_s^n} \right]^{-1} = \left[\frac{\gamma}{k_{eff}^n} + \frac{1-\gamma}{k_s^n} \right]^{-1}. \tag{2-23}$$

The effectiveness of the Wielandt shift method depends on the choice of the eigenvalue shift, δk. Typically, fewer iterations are necessary to converge the outer iteration scheme as δk decreases. However, the problem becomes less diagonally dominant for smaller δk. Empirically, it has been found that $\delta k = 0.04$ performs well for a range of LWR applications [32].

Since the RBWR features a hexagonal lattice, a special radial solver is required. PARCS uses the TPEN method [50] [32], or Triangular Polynomial Expansion method, to solve two transverse-integrated neutron diffusion equations for a hex-octahedron node. The transverse-integrated 2-D equation is found by integrating the multi-group transport equation in the axial direction (z-direction):

$$-D_g\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\phi_g^R(x,y)+\Sigma_{rg}\phi_g^R(x,y) - \sum_{g'<g}\Sigma_{g'g}\phi_{g'}^R(x,y)=Q_g^R(x,y)-L_g^Z(x,y), \tag{2-24a}$$

where the superscript R represents an axially integrated quantity and L_g^Z is the axial transverse leakage term. This produces the radial equation within a plane of interest, where the terms in Eq. 2-24a are defined as:

$$\phi_g^R(x,y)=\frac{1}{z^T-z^B}\int_{z^B}^{z^T}\phi_g\,(\vec{r})dz, \tag{2-24b}$$

$$Q_g^R(x,y)=\frac{1}{z^T-z^B}\int_{z^B}^{z^T}Q_g\,(\vec{r})dz, \tag{2-24c}$$

$$\Sigma_g\,(x,y)=\frac{1}{z^T-z^B}\,\frac{\int_{z^B}^{z^T}\Sigma_g(\vec{r})\phi_g(\vec{r})dz}{\phi_g^R(x,y)}, \tag{2-24d}$$

$$L_g^Z(x,y)=\frac{1}{z^T-z^B}\int_{z^B}^{z^T}\mu\frac{\partial}{\partial z}\,\phi_g(\vec{r})dz. \tag{2-24e}$$

The terms z^B and z^T represent the heights of the bottom and top planes of integration respectively. The transverse-integrated 1-D equation is found by integrating the multi-group transport equation in the radial direction (x- and y-direction):

$$-D_g\frac{\partial^2}{\partial z^2}\phi_g^z(z)+\Sigma_{rg}\phi_g^z(z)-\sum_{g'<g}\Sigma_{g'g}\phi_{g'}^z(z)=Q_g^z(z)-L_g^R(z). \tag{2-25a}$$

where the superscript z represents a radially integrated quantity and L_g^R represents the radial transverse leakage. Equation 2-25a represents the axial expression and the terms are defined as:

$$\phi_g^z(z) = \frac{1}{A} \iint_A \phi_g\,(\vec{r})dxdy, \tag{2-25b}$$

$$Q_g^z(z) = \frac{1}{A} \iint_A Q_g\,(\vec{r})dxdy, \tag{2-25c}$$

$$\Sigma_g\,(z) = \frac{1}{A} \frac{\iint_A \Sigma_g(\vec{r})\phi_g\,(\vec{r})dxdy}{\phi_g^z(z)}, \tag{2-25d}$$

$$L_g^R(z) = \frac{1}{A} \iint_A sin\theta \left(cos\alpha \frac{\partial}{\partial x} + sin\alpha \frac{\partial}{\partial y}\right) \phi_g\,(\vec{r})dxdy, \tag{2-25e}$$

where A is the area of the plane of integration, $\vec{\Omega} = (\mu, \alpha)$ and $\mu = cos\theta$. The combination of Eqs. 2-24 and 2-25 compose the radial and axial equations for the TPEN method. This method uses the 2-D/1-D approach, where spatially three-dimensional problems are decomposed into a 1-D-axial stack and 2-D-radial planes. The solution of the 2-D-radial plane is found by splitting the hexagon into 6 triangles and performing a polynomial expansion on the flux within each triangle. In addition, there are two approximations that must be made. The first involves only using average transverse leakages. For the 2-D problem, the uniform axial leakage is used while for the 1-D problem, the quadratic leakage shape is determined by average leakages of current and two neighbor nodes. The second states that the average cross sections are μ weighted with scalar fluxes instead of angular planes.

There are 36 unknowns defined per energy group for a single hexagonal node. These quantities include 6 hexagon surface outgoing partial currents, 6 inner surface average fluxes, 6 triangle average fluxes, 6 x-moments, 6 y-moments, 1 center point flux, 2 axial partial currents (top and bottom), 1 first-order z-moment, and 1 second-order z-moment, and 1 hexagon average flux. Additional information on the solution of these equations is outlined in the PARCS manual [32].

The solution to Eq. 2-25 is the principal focus of this work. Within the TPEN solver, the axial equation is solved using NEM diffusion and is coupled to Eq. 2-24 through a transverse

leakage term. Typically this method is applied to axially homogeneous problems. However, for an axially heterogeneous assembly, NEM diffusion is not sufficient in resolving the spatial flux at material interfaces. In chapters 2 through 4, the use of discontinuity factors and higher order methods are presented as an alternative approach.

2.1.4 PATHS

Though the majority of the work discussed in this document is related to neutronics, it is important to briefly discuss the thermal-hydraulics code used in the analysis, since the prediction of the void fraction in the core is essential for the accurate simulation of the RBWR. PATHS [34] was developed to calculate a steady-state thermal-hydraulics solution for BWRs. PATHS utilizes a four-equation drift flux model with simplified equations and solution algorithms that considerably reduce the runtime compared to six-equation, two-fluid codes such as TRACE or RELAP5. User-specified boundary conditions include outlet pressure, total core mass flow rate, and inlet enthalpy. The efficiency of PATHS makes it possible to perform one-to-one mapping of neutronics/thermal-hydraulics coupled calculations. The following description is from the PATHS manual [34].

The PATHS methodology is based on the two-fluid model developed by Ishii [51], averaged to consider the mixture instead of two separate fluids. The finite volume method is applied and the equations are cast into a face-based scheme. Since the control volume is set to include the entire cross-sectional area in a channel, the equations are reduced to a one-dimensional flow with heat fluxes and stress terms coming from the boundary. The void fraction and drift velocity are introduced through constitutive relationships instead of a fourth field equation that would normally show up in the drift flux model.

The discretized conservation equations for mass, momentum, and energy, are given by

$$(\rho_m^* v_m A\,)_n - (\rho_m^* v_m A)_s = 0, \tag{2-26}$$

$$\left(\rho_m^*|v_m^*|v_m\left(A+\frac{1}{2}F_z^*\Delta V\right)\right)_n-\left(\rho_m^*|v_m^*|v_m\left(A-\frac{1}{2}F_z^*\Delta V\right)\right)_s+(PA)_n-(PA)_s \tag{2-27}$$

$$=\left(\frac{\alpha^*}{1-\alpha^*}\frac{\rho_f^*\rho_g^*}{\rho_m^*}v_{gj}^{2*}A\right)_S-\left(\frac{\alpha^*}{1-\alpha^*}\frac{\rho_f^*\rho_g^*}{\rho_m^*}v_{gj}^{2*}A\right)_n-\frac{(\rho_m^*)_n+(\rho_m^*)_n}{2}g\Delta V,$$

$$F_z^*=\frac{1}{2}\left(\frac{\Phi_{2\Phi}f}{D_h}+\frac{K_{loss}}{\Delta z}\right), \tag{2-28}$$

$$((\rho_m^*v_m^*hA)_n)-((\rho_m^*v_m^*hA)_s)=q_{wall}^{"}\xi_h\Delta z+\left(\alpha^*\frac{\rho_f^*\rho_g^*}{\rho_m^*}\Delta h_{fg}^*v_{gj}^*A\right)_s-\left(\alpha^*\frac{\rho_f^*\rho_g^*}{\rho_m^*}\Delta h_{fg}^*v_{gj}^*A\right)_n$$

$$+\left(v_m^*+\frac{\alpha^*\left(\rho_f^*-\rho_g^*\right)}{\rho_m^*}\bar{v}_{gj}\right)((P^*A)_n-(P^*A)_s), \tag{2-29}$$

where standard Greek letters are used to refer fluid quantities (e.g. ρ for density) and the subscripts *f,g,* and *m* denote fluid (liquid), gas, and mixture, respectively. The use of * indicates using the previous iteration value, and the subscripts n and s refer to the north and south faces respectively. The fluid quantities are v for velocity, A for channel cross-sectional flow area, ΔV for the volume of the discretized cell, P for pressure, α for void fraction, v_{gj} for the drift velocity, g for gravitational acceleration, $\Phi_{2\Phi}$ for two-phase friction multiplier, f for friction factor, D_h for hydraulic diameter, K_{loss} for loss coefficient, Δz for the height of the cell, h for enthalpy, $q_{wall}^{"}$ for wall heat flux, ξ_h for the heated perimeter, and Δh_{fg} for the specific enthalpy of vaporization.

The user may choose from various void correlations. The void correlations in PATHS are appropriate for standard BWRs. The default is an Electric Power Research Institute (EPRI) void model, which is sufficient for most BWR applications. However, because of the tight-pitch fuel design of the RBWR-AC, the EPRI correlation produces a poor model of the physical void distribution. The available experimental data for tight-pitch BWR assemblies was compared against several different void correlations to obtain a more accurate model [52]. The Lia, Parlos, and Griffith (LPG) model was determined to most closely match experimental data, and was subsequently implemented in PATHS and used for all equilibrium cycle and transient analysis of the RBWR-AC. It should be noted that Hitachi has a different void correlation, which was used

for the AC model. In general, this model predicts a higher void fraction than LPG and shows similar results to the EPRI correlation. Figure 2.5 shows the effect of the void correlation on a single-channel PARCS-PATHS simulation.

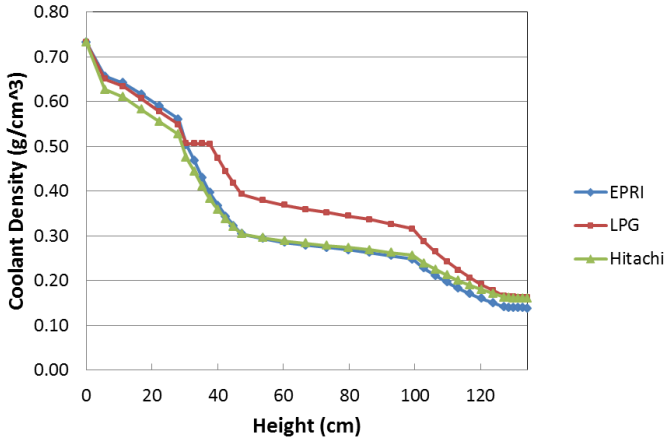

Figure 2.5: RBWR-AC void correlation comparison [31]

2.2 RBWR Assembly Analysis

The above sections outlined the core simulator that was developed for analysis of axially heterogeneous cores. In the remainder of this chapter, the code system is applied to the RBWR. This includes the Monte Carlo modeling of a three-dimensional assembly followed by one-dimensional and three-dimensional diffusion calculations using the PARCS code system.

2.2.1 Monte Carlo Analysis and Modeling

The first step of modeling the RBWR involved generating a set of Monte Carlo cross sections using the Serpent code. An illustration of the assembly is shown in Figure 2.6. In the axial direction, a total of 34 meshes were used to discretize the fuel regions (5 in the lower blanket, 8 in the lower fissile, 8 in the internal blanket, 8 in the upper fissile and 5 in the upper blanket) while using only a single mesh for each axial reflector. Table 2.1 specifies the axial meshing.

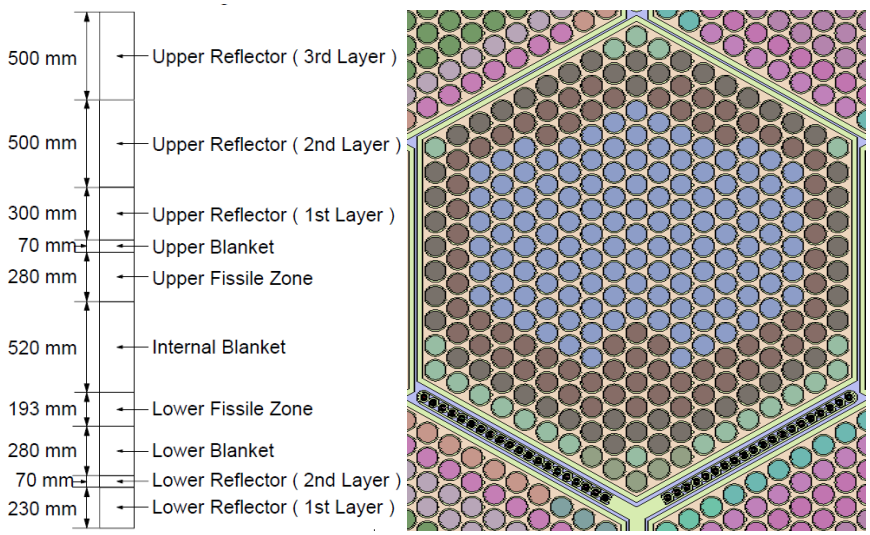

Figure 2.6: RBWR assembly (left) axial view and (right) radial view

Table 2.1: Axial meshing scheme

Region	Number of Mesh	Mesh Size (cm)
Lower Reflector (1st Layer)	1	23.0
Lower Reflector (2nd Layer)	1	7.0
Lower Blanket	5	5.6
Lower Fissile Zone	8	2.4125
Internal Blanket	8	6.5
Upper Fissile Zone	8	3.5
Upper Blanket	5	1.4
Upper Reflector (1st Layer)	1	30.0
Upper Reflector (2nd Layer)	1	50.0
Upper Reflector (3rd Layer)	1	50.0

Due to the complexity of the RBWR assembly design and the limitations of the software used for modeling, several assumptions were made. The first is due to the use of the Serpent Monte Carlo code. Unlike other Monte Carlo software that allows the user to apply a boundary condition to any surface, Serpent requires the user to apply the global boundary condition to a single surface. This places restrictions on the bounding surface used to describe the problem.

Because the RBWR assembly is an irregular hexagon and Serpent can only model regular hexagonal prisms, the regions outside of the assembly can (bypass coolant, control rod sheath and control rod) for the southeast and southwest faces shown in Figure 2.7 were collapsed in order to fit the assembly within a regular hexagon. These changes were performed by reducing the volume of the control rod, sheath and bypass coolant regions while increasing the density of each material to preserve the total mass. The final product of this outer region collapsing is shown in the right image of Figure 2.7 where the RBWR assembly is represented as a regular hexagon.

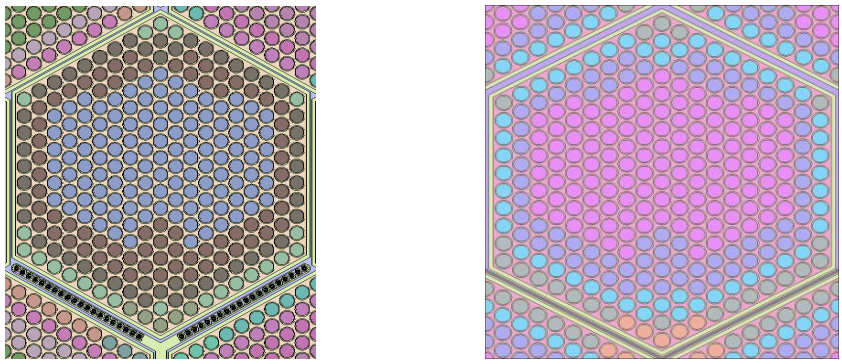

Figure 2.7: Planar assembly view before outer region collapsing (left) and after (right)

The eigenvalues for these two cases were compared to test the impact of the outer region collapsing. This comparison was done using 2-D Serpent models. Table 2.2 shows the two cases for several depletion points. The error can become as large as 450 pcm from the full model. However, it is important to note that we are not trying to reproduce the exact assembly in this work. Instead, the purpose of this collapsing is to develop an assembly model with similar characteristics to the RBWR, but with an outer boundary represented as a regular hexagon. Reflective boundary conditions were applied in Serpent on all sides of the assembly.

Table 2.2: Comparison of heterogeneous and homogeneous models

Burnup (MWd/kg)	Heterogeneous Hex	Homogenized Hex	Difference (pcm)
0	1.25761	1.25584	-177
2	1.25268	1.24811	-457
10	1.23539	1.23245	-294
50	1.15997	1.15602	-395
150	1.01736	1.01882	146

Typical lattice analysis only involves two dimensions, but the advantage of using Monte Carlo is extending the work to three dimensions. However, this also increases the memory burden due to the increased number of burnable regions that are tracked. The axial discretization in the Monte Carlo model tracks 34 separate axial regions. This, combined with the five separate pin enrichment types (illustrated by the different colored circles in Figure 2.8), means that there are a total of 170 different burnable regions. Tracking all of these regions increases the amount of computational memory required (upwards of 70 Gb for this problem). To reduce the amount of memory, the fuel pins were homogenized by volume averaging the five separate radial pin enrichments into a single enrichment as shown in Figure 2.8.

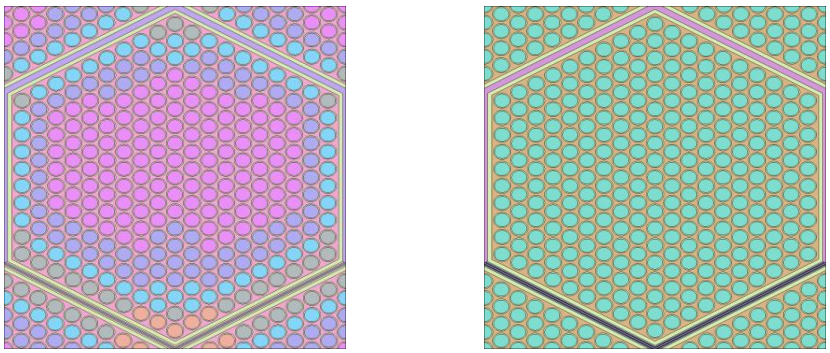

Figure 2.8: Five pin enrichment model (left) and homogenized pin enrichment (right)

Similar to the outer region collapsing, two models were created to compare the effect of homogenizing the fuel. Instead of using a two dimensional model for this case, the Serpent input was extended to three dimensions. Table 2.3 compares the two cases for select burnup points. For this case the largest differences are at low burnup and high burnup at around 80 pcm difference.

Table 2.3: Comparisons of fuel homogenization for a three dimensional assembly

Burnup (MWd/kg)	5 Pin Types	1 Pin Type	Difference (pcm)
0	1.05068	1.05151	83
2	1.05546	1.05520	-26
10	1.03764	1.03787	23
50	0.95574	0.95545	-29
150	0.83657	0.83571	-86

The cross sections were calculated with the P1 infinite spectrum. When developing 3-D Monte Carlo cross sections, the total number of particle histories is crucial in reducing the statistical uncertainty associated with the generated group constants. Each cross section is calculated with tallies from the simulation. If the number of simulated particles is too small, the uncertainty of these tallies will be large. Increased uncertainties may lead to greater errors in the cross sections, which affect the accuracy of the nodal solution. For the work presented here, the statistics were chosen such that the relative uncertainty on all quantities was less than five percent.

2.2.2 Core Diffusion Analysis

The above section developed a set of cross sections for use in the PARCS core simulator. In order to test the validity of these values, a single assembly comparison between the Serpent reference simulation and PARCS using Serpent generated cross sections was performed. The Serpent calculation was performed using 150,000 source particles per cycle, with 300 active cycles and 100 inactive cycles using ENDF/B-VII neutron cross section library. Reflective boundary conditions were applied to all sides of the assembly, and group constants were found for each axial level. Cross sections were collapsed to 12 energy groups based on previous Hitachi studies [28]. This energy group structure is shown in Table 2.5, along with the upper fissile flux spectrum in Figure 2.9. The results of this comparison are shown in the first two rows of Table 2.4. There is over an 800 pcm difference between the reference solution and the one-dimensional diffusion solution. To account for this, discontinuity factors were applied to the same problem in the axial direction as discussed in Section 2.1.1.3. Table 2.4 shows the comparison of the single assembly results.

Table 2.4: Eigenvalue comparison for single assembly case

Method	k	Difference from Serpent (pcm)
3-D Serpent	1.09601	-
3-D PARCS without ZDFs	1.08772	-829
3-D PARCS with ZDFs	1.09601	0

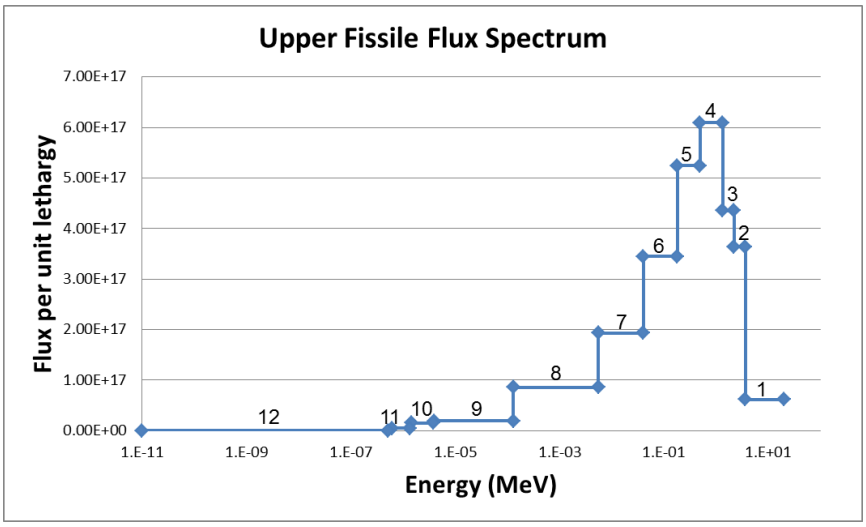

Figure 2.9: Upper fissile flux spectrum based on 12-group energy structure

Table 2.5: 12-group energy structure for cross sections [28]

Group Number	Upper Energy (eV)
1	1.0000E+7
2	3.6788E+6
3	2.2313E+6
4	1.3534E+6
5	4.9787E+5
6	1.8316E+5
7	4.0868E+4
8	5.5308E+3
9	1.3007E+2
10	3.9279E+0
11	1.4450E+0
12	6.2500E-1
Minimum	8.2500E-5

When using only 3-D cross sections, with a relative statistical error of 9.6 pcm on k_{eff} for the Monte Carlo solution, the PARCS solution is over 800 pcm different from the Monte Carlo

solution. As shown in Table 2.4, when axial discontinuity factors are introduced, PARCS is able to reproduce the exact Monte Carlo solution.

A comparison of the normalized fluxes between Serpent and PARCS for an RBWR-like assembly with and without axial discontinuity factors for the fast (group 1) and thermal (group 9) are shown in Figure 2.10. These plots represent the flux over the active core region. From 0-30 cm represents the lower blanket, 30-50 cm the lower fissile, 50-100 cm the internal blanket, 100-130 cm the upper fissile and 130-140 cm the upper blanket. Each of these regions is divided by a solid line in the Figure. For the fast group, the PARCS solution without ZDFs underestimates the flux in the lower fissile zone and overpredicts the flux in the blanket regions near the upper fissile zone. With ZDFs, the solution is consistent with the Serpent flux profile. For the thermal group, the PARCS solution without ZDFs underestimates the flux in the lower fissile region. It also has difficulty reproducing the solution in the upper blanket, where it over predicts the flux. The flux shape in the thermal group is not as smooth as the fast group flux shape and it is much more difficult for the diffusion solution to reproduce the Monte Carlo solution without the use of axial discontinuity factors.

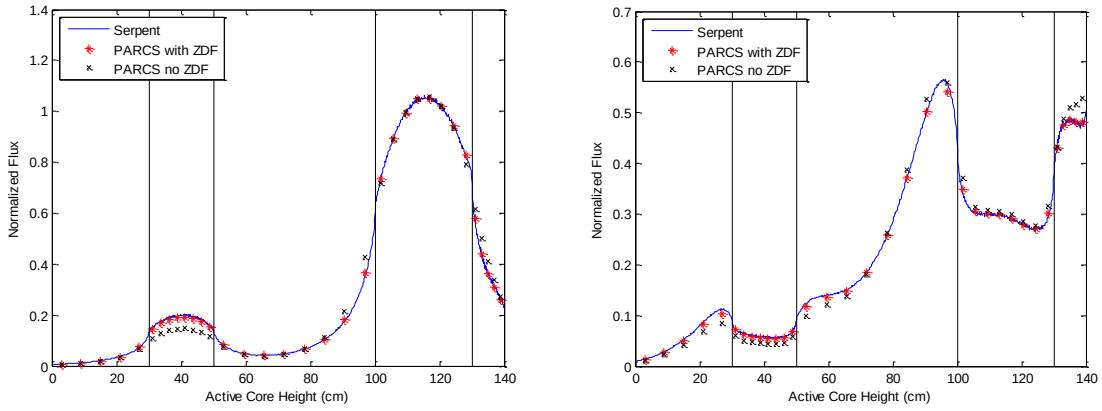

Figure 2.10: Fast group 1 flux (left) and thermal group 9 flux (right)

The use of ZDFs for a single assembly at steady-state conditions is capable of reproducing the reference solution. In the following section, ZDFs are applied to full three-dimensional core diffusion problems with thermal-hydraulic feedback.

2.3 Full-Core Analysis

The methods were then applied to full core analysis and the calculation of an equilibrium cycle. 3-D cross sections were generated for the history and branching scheme shown in Table 2.6 and Table 2.7. ZDFs were created for all nodes and appended to the PMAXS files. A one-third core PARCS model of 240 assemblies was used with a one-to-one channel mapping to the thermal-hydraulics code PATHS. Node-wise burnups were extracted from Serpent for each homogenized region, and PMAXS cross sections were generated for all 39 axial regions.

Table 2.6: RBWR-AC history structure

History	Control Rod State	Core Average Moderator Density (g/cc)	Corresponding Void Fraction (%)	Fuel Temperature (K) (blanket/fissile)
1	0	0.33402	58	600/900
2	0	0.25845	68	600/900
3	0	0.50281	33	600/900
4	1	0.33402	58	600/900

Table 2.7: RBWR-AC cross section branch structure

Branches	Index	Control Rod State	Core Average Moderator Density (g/cc)	Corresponding Void Fraction (%)	Fuel Temperature (K) (blanket/fissile)
RE	1	0	0.33402	58	600/900
CR	2	1	0.33402	58	600/900
DC	3	0	0.00001	100	600/900
DC	4	0	0.25845	68	600/900
DC	5	0	0.50281	33	600/900
DC	6	0	0.76000	0	600/900
DC	7	1	0.76000	0	600/900
TF	8	0	0.33402	58	500/600
TF	9	0	0.33402	58	1200/2000

When performing the coupled simulation, instabilities arose that caused the solution to diverge. If the same simulation was performed but without the use of axial discontinuity factors then the PARCS solution converged without any issues. Investigation of the discontinuity factors

found that for material interfaces with steep gradients in the flux produced ZDFs much larger than 1 and sometimes even negative. The reason for this is due to the estimation of the homogeneous surface flux in the definition of the discontinuity factor. Consider the following situation, in which a given node has a low relative average flux and borders a region with a steep drop in the flux. When the homogeneous solution estimates the surface flux from this location, it can overestimate the drop to the surface and produce a negative surface flux. Similarly, a large discontinuity factor can be found if the homogeneous surface flux becomes very close to zero. These negative discontinuity factors produce negative fluxes within the nodal simulation. For the steady-state simulation with no feedback, this did not cause instabilities because all of the group constants were calculated specifically for a certain set of conditions. Consider the following example illustrated in Figure 2.11.

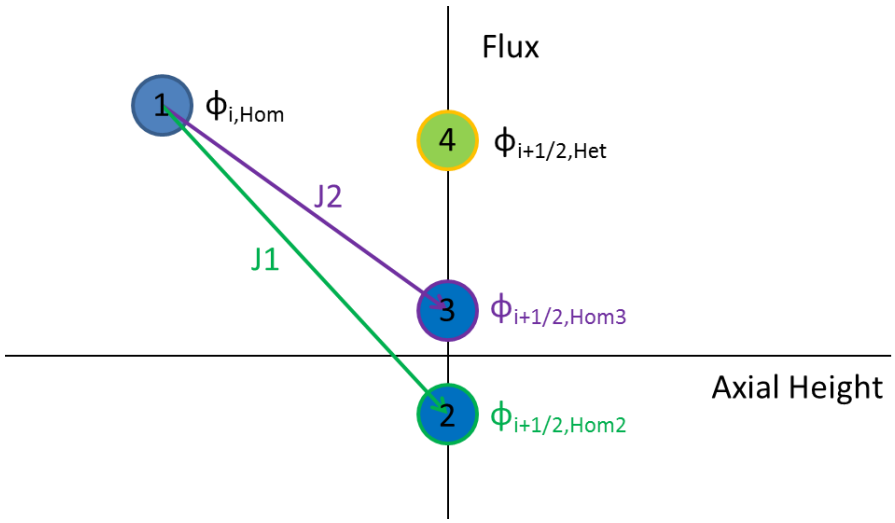

Figure 2.11: Estimation of homogeneous surface flux for two separate cases

Point 1 represents the node i average homogeneous flux. Consider two separate cases that both have the same average flux in node i but separate surface currents on surface i+1/2. These currents will be referred to as J1 and J2 as depicted in Figure 2.11. An estimate of the homogeneous surface flux on surface i+1/2 can be found using the node average flux and group constants along with either currents J1 or J2. The estimate using J1 is illustrated as point 2 and the estimate for J2 is shown as point 3. When the homogeneous surface flux is estimated at point 2, one obtains a negative homogeneous surface flux. Since the heterogeneous surface

flux (point 4) is always positive, this will produce a negative discontinuity factor. If instead the homogeneous surface flux is estimated at point 3, this provides a positive but close to zero homogeneous surface flux. The discontinuity factor for this situation can become quite large as this point approaches zero. Both of these scenarios depict instances in which the homogeneous surface flux can be close to zero. When feedback is introduced into the simulation, the state conditions are allowed to vary. This means the discontinuity factors will change, based on the state conditions. In the example in Figure 2.11, if the state conditions change such that the estimate of the homogeneous surface flux approaches zero, the discontinuity factor approaches infinity. The fluxes produced using these discontinuity factors can easily become erratic, and when coupled to a feedback code such as PATHS, can quickly lead to instabilities. This was found to be the root cause of the divergence in the coupled code solution.

In order to ensure stability in the coupled simulation, the discontinuity factors should ideally be close to one. This can be enforced by introducing bounds on the values of the discontinuity factors. It is important to note, however, that this method will not reproduce the reference solution and will introduce some amount of error into the calculation. The magnitude of this error was studied using the assembly example in the previous section, and a comparison of the eigenvalue is shown in Table 2.8. For this study, the bounds were chosen to be 0.85 for the lower limit and 1.15 for the upper limit. These limits were chosen empirically, based on the stability of several assembly calculations where the discontinuity factor range was allowed to vary [31].

Table 2.8: Eigenvalue comparison for single assembly case

Method	k	Difference from Serpent (pcm)
3-D Serpent	1.09601	-
3-D PARCS without ZDFs	1.08772	-829
3-D PARCS with ZDFs – Bounding	1.09645	44
3-D PARCS with ZDFs	1.09601	0

For this case, an error of 44 pcm was introduced by bounding the range of the discontinuity factors. This technique was then applied to the coupled core calculation. With the bounded

discontinuity factors, PARCS was able to generate a coupled core solution. Results for the axial and radial power distribution are shown in Figure 2.12.

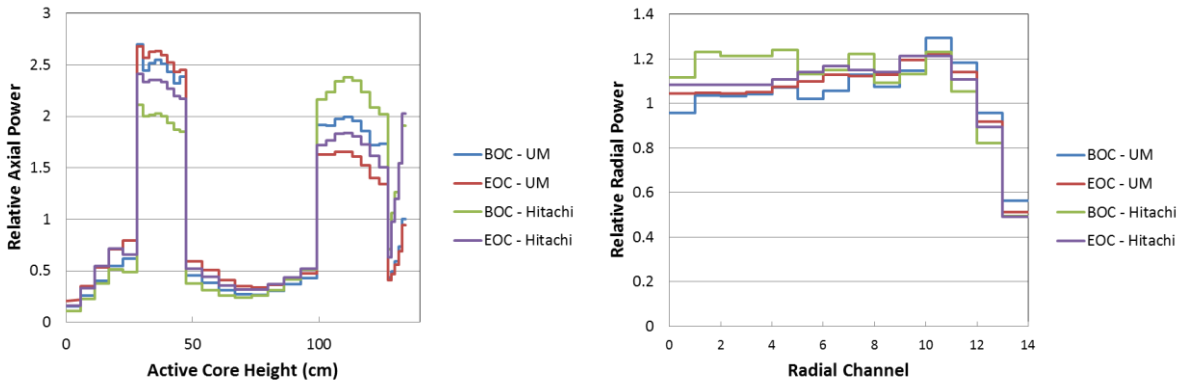

Figure 2.12: Core averaged axial (left) and radial (right) power distributions for the RBWR [31]

It is important to state that the differences between the University of Michigan (UM) and Hitachi results in Figure 2.12 are due to several reasons. The UM results feature the use of three-dimensional cross sections and ZDFs while the Hitachi results are based on two-dimensional cross sections with no discontinuity factors. This difference in methodology produces noticeable differences in the power shape and was the focus of previous work [31].

The use of ZDFs provides a method to improve the solution of nodal codes for axially heterogeneous problems. However, these values required bounding to ensure stability when extended to three-dimensional core problems with feedback and therefore introduced error into the simulation. The next chapter discusses the use of higher order expressions, mainly the Pn equations and Quasidiffusion, to improve the accuracy of the axial solver.

Chapter 3
Pn and Quasidiffusion Methods

In the previous chapter, diffusion theory was used to solve the axial heterogeneous problem within PARCS. However, it became necessary to bound the axial discontinuity factors (ZDFs), which were introduced during the nodal homogenization process, in order to ensure convergence was achieved for all core conditions. The purpose of this chapter is to investigate the use of higher-order Pn and Quasidiffusion methods, to provide a more accurate axial flux solution and to eliminate the need to bound the discontinuity factors.

To test this hypothesis, a one-dimensional simulation within PARCS using diffusion and P3 was performed. The results of this analysis compared to the Serpent reference solution are compared in Table 3.1 and the flux distributions are shown in Figure 3.1 and Figure 3.2.

Table 3.1: Comparison of one dimensional assembly calculations

Method	K-inf	Difference from Serpent (pcm)
3-D Serpent	1.04192	-
PARCS – Diffusion	1.03504	-688
PARCS – Diffusion + ZDFs	1.04192	0
PARCS – SP3	1.04282	90

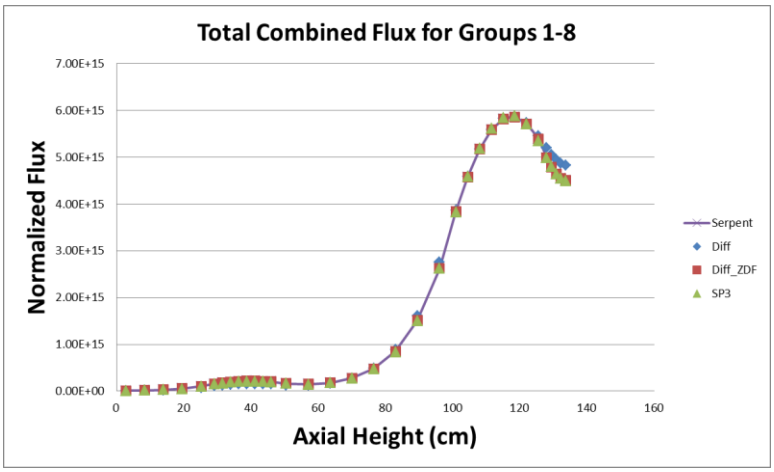

Figure 3.1: Total combined groups 1-8 normalized axial flux comparison

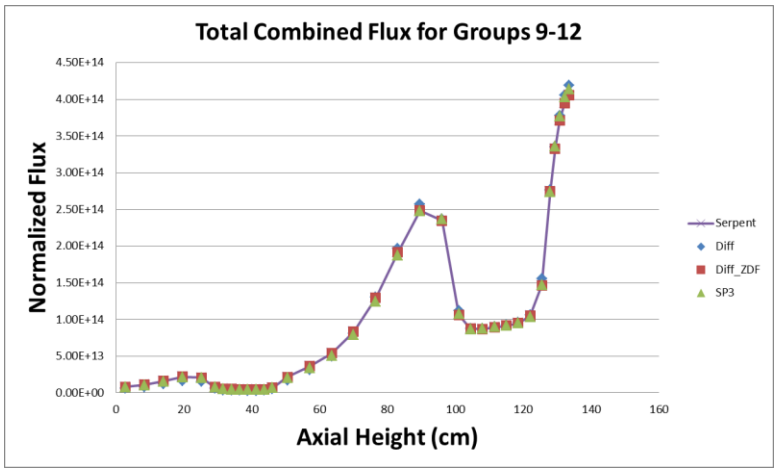

Figure 3.2: Total combined groups 9-12 normalized axial flux comparison

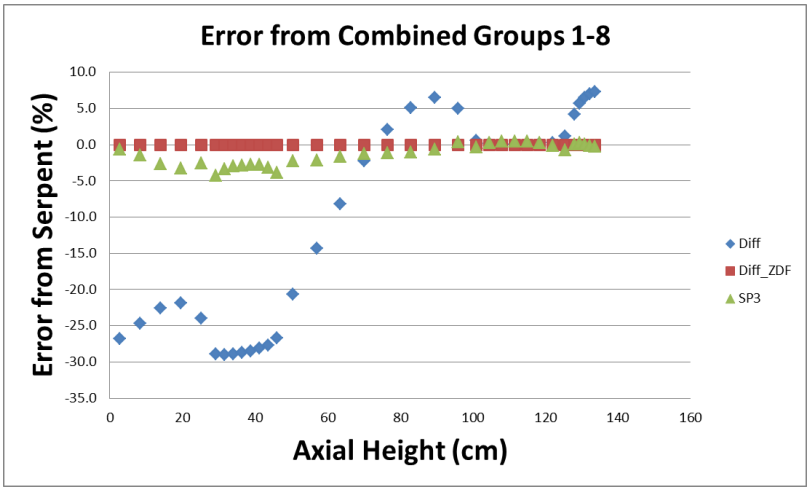

Figure 3.3: Combined groups 1-8 flux error compared to the reference Monte Carlo solution

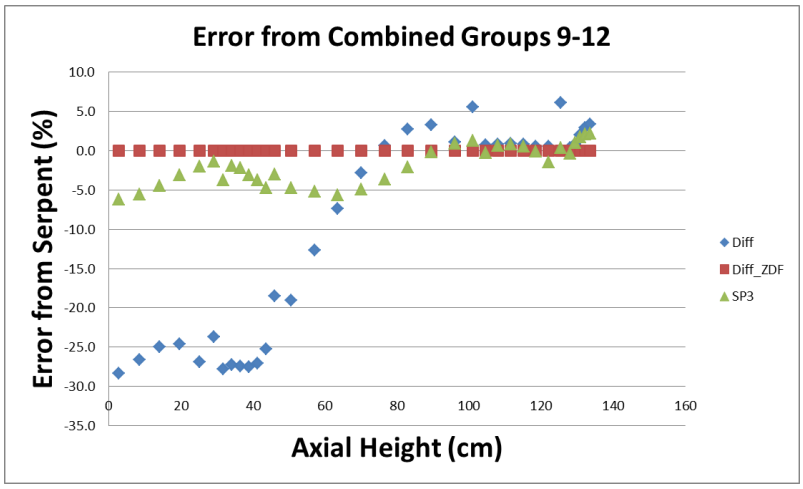

Figure 3.4: Combined groups 9-12 flux error compared to the reference Monte Carlo solution

As indicated in the Tables and Figures, the P3 solution provides a much more accurate eigenvalue and flux compared to the diffusion solution without ZDFs. However, as expected, when discontinuity factors are used with the diffusion solution, a better estimate is provided relative to the P3 solution, since the discontinuity factors are formed such that the diffusion solution reproduces the exact Serpent reference solution.

From this example, it is apparent that P3 provides an improved solution compared to diffusion theory. In the next sections, the one-dimensional Pn and Quasidiffusion equations are applied to the RBWR problem and the solution is compared to the diffusion theory methods described in Chapter 2.

3.1 Pn Equations

Within this chapter there are four equations that are derived to handle the neutron physics within a reactor. These include the P1, P2, P3 and Quasidiffusion equations. Each of these provides different advantages for solving criticality systems. The SPn equations, or simplified spherical harmonics equations, are based on the Pn equations which utilize Legendre polynomial expansions [53] [54] [55]. This is useful when deriving functions for general anisotropic scattering. These equations extend the traditional Pn expressions to three dimensions by replacing the 1-D diffusion operator with the 3-D diffusion operator as well as

the boundary condition derivative terms with the outward normal derivative. It can also be shown that the SPn equations reduce to the Pn equations when performing calculations in one dimension. However, all the applications in this thesis are performed for the 1-D axial problem and therefore only Pn theory is necessary.

3.1.1 Pn Derivation

In order to derive the P1, P2 and P3 equations, the generic form of the Pn expressions are described which begins with the 1-D azimuthally-integrated planar geometry multigroup transport equation [56].

$$\mu \frac{d}{dx}\psi(x,\mu) + \Sigma_t(x)\psi(x,\mu) = \int_{-1}^{1} \Sigma_s(x,\mu,\mu')\psi(x,\mu')d\mu' + \frac{Q(x)}{2}, \tag{3-1a}$$

$$0 < x < X, -1 \le \mu \le 1, 1 \le g \le G,$$
$$\psi(0,\mu) = \psi^b(\mu), 0 < \mu \le 1, 1 \le g \le G,$$
$$\psi(X,\mu) = \psi^b(\mu), -1 \le \mu < 0, 1 \le g \le G,$$

where the term $Q(x)$ is used to simplify the fission term since it has no dependence on the angular variable μ,

$$Q(x) = \lambda\chi(x)\nu\Sigma_f(x)\int_{-1}^{1}\psi(x,\mu')d\mu'. \tag{3-1b}$$

In Eq. 3-1 there is only one direction-of-flight variable (μ). Legendre polynomial expansions are performed on the scattering cross section (see Appendix C for more details) and the angular flux:

$$\Sigma_s(x,\mu,\mu') = \sum_{n=0}^{\infty} \frac{2n+1}{2} P_n(\mu)P_n(\mu')\Sigma_{sn}(x), \tag{3-2}$$

$$\psi_g(x,\mu) = \sum_{m=0}^{\infty} \frac{2m+1}{2}\psi_{m,g}(x)P_m(\mu), \tag{3-3}$$

where

$$\psi_{m,g}(x) = \int_{-1}^{1} P_m(\mu')\psi_g(x,\mu')d\mu'. \tag{3-4}$$

Next, each term of Eq. 3-1 is multiplied by $P_n(\mu)$ and integrated over -1 ≤ μ ≤ 1 while using the above expansions starting with the first term:

$$\frac{d}{dx}\sum_{m=0}^{\infty}\frac{2m+1}{2}\left(\int_{-1}^{1} P_n(\mu)\,\mu P_m(\mu)d\mu\right)\psi_m(x). \tag{3-5}$$

The integral in Eq. 3-5 is simplified using Bonnet's recursion formula and the orthogonality condition:

$$\mu P_n(\mu) = \frac{(n+1)P_{n+1}(\mu) + nP_{n-1}(\mu)}{2n+1}, \tag{3-6}$$

$$\int_{-1}^{1} P_n(\mu)P_m(\mu)d\mu = \frac{2}{2n+1}\delta_{n,m}. \tag{3-7}$$

Using these two relations in the first term reduces the expression to the following:

$$\frac{d}{dx}\sum_{m=0}^{\infty}\frac{2m+1}{2}\left(\int_{-1}^{1} P_n(\mu)\left[\frac{(m+1)P_{m+1}(\mu)+mP_{m-1}(\mu)}{2m+1}\right]d\mu\right)\psi_m(x)$$

$$= \frac{d}{dx}\sum_{m=0}^{\infty}\frac{1}{2}\left(\int_{-1}^{1}[P_n(\mu)\,(m+1)P_{m+1}(\mu) + P_n(\mu)mP_{m-1}(\mu)]d\mu\right)\psi_m(x)$$

$$= \frac{d}{dx}\sum_{m=0}^{\infty}\frac{1}{2}\left[(m+1)\frac{2}{2n+1}\delta_{n,m+1} + m\frac{2}{2n+1}\delta_{n,m-1}\right]\psi_m(x)$$

$$= \frac{d}{dx}\sum_{m=0}^{\infty}\left[\frac{n}{2n+1}\delta_{n-1,m} + \frac{n+1}{2n+1}\delta_{n+1,m}\right]\psi_m(x)$$

$$= \frac{d}{dx}\left[\frac{n}{2n+1}\psi_{n-1}(x) + \frac{n+1}{2n+1}\psi_{n+1}(x)\right]. \tag{3-8}$$

Next, the second term is reduced using the orthogonality condition:

$$\Sigma_t(x)\int_{-1}^{1}\sum_{m=0}^{\infty}\frac{2m+1}{2}\psi_{m,g}(x)P_n(\mu)P_m(\mu)\,d\mu$$
$$=\Sigma_t(x)\sum_{m=0}^{\infty}\frac{2m+1}{2}\psi_{m,g}(x)\frac{2}{2n+1}\delta_{n,m}$$
$$=\Sigma_t(x)\psi_n(x). \tag{3-9}$$

The scattering term is solved for next:

$$\int_{-1}^{1}\sum_{k=0}^{\infty}\frac{2k+1}{2}\,\Sigma_{sk}(x)\left[\int_{-1}^{1}P_n(\mu)\,P_k(\mu')P_k(\mu)d\mu\right]\psi(x,\mu')d\mu'$$
$$=\int_{-1}^{1}\sum_{k=0}^{\infty}\frac{2k+1}{2}\,\Sigma_{sk}(x)\left[\frac{2}{2n+1}\delta_{n,k}\right]P_k(\mu')\,\psi(x,\mu')d\mu'$$
$$=\int_{-1}^{1}\Sigma_{sn}(x)P_n(\mu')\psi(x,\mu')d\mu'. \tag{3-10a}$$

If Eq. 3-4 is appled to Eq. 3-10a, this produces the final form of the scattering term:

$$=\Sigma_{sn}(x)\psi_n(x). \tag{3-10b}$$

The last term to solve for is the fission term:

$$\int_{-1}^{1}P_n(\mu)\frac{Q(x)}{2}d\mu\;=Q(x)\delta_{n,0}. \tag{3-11}$$

Combining each of the terms, the general form of the Pn equations can then be written as:

$$\frac{d}{dx}\left[\frac{n}{2n+1}\psi_{n-1}(x)+\frac{n+1}{2n+1}\psi_{n+1}(x)\right]+\Sigma_t(x)\psi_n(x)$$
$$=\Sigma_{sn}(x)\psi_n(x)+Q(x)\delta_{n,0},\qquad 0\le n\le N. \tag{3-12}$$

where N represents the largest value of n for a given approximation ($N = 1$ for P1 and $N = 3$ for P3). When $n = N$ for Eq. 3-12, the $\psi_{n+1}(x)$ term is found using the "closure relation":

$$\psi_{n+1}(x) = 0, \qquad n = N. \tag{3-13}$$

3.1.2 Discontinuity Factors for the Pn Equations

Because of the higher-order flux expansions, the Pn equations can provide an improved solution compared to diffusion. However, discontinuity factors are still required because of the errors introduced in applying homogenized cross sections with the higher order Pn equations. As described in Chapter 1, generalized equivalence theory provides a correction that allows for the continuity of current while allowing the homogeneous surface fluxes to become discontinuous. The discontinuity factor was defined in Eq. 1-22 as the ratio of the heterogeneous surface flux to the homogeneous surface flux:

$$f = \frac{\phi^{Het,s}}{\phi^{Hom,s}}, \tag{3-14}$$

where the heterogeneous surface flux is calculated during the transport calculation and the homogeneous surface flux is taken as the average cell flux for a problem with reflective boundary conditions. The discontinuity factors are calculated for each side of an interface for all energy groups and are provided with the generated cross sections. In addition to deriving the various Pn equations, the derivation of discontinuity factors will also be discussed for each of the Pn equations in the following sections.

3.2 P1 Equations

The first technique derived is the P1 or diffusion theory method. This equation has been sufficiently accurate for many important practical problems required in the simulation of nuclear reactor cores. However, the P1 equations have limitations when the system contains significant neutron absorption or streaming, which is a concern for RBWR-type systems due to the increased neutron streaming in the axial direction. The purpose of including the P1

equations in this section is simply to provide a consistent basis to evaluate the advantages of the higher order Pn equations.

3.2.1 P1 Derivation

From Eq. 3-12 the two first-order differential equations that represent the P1 planar geometry approximation can be written as:

$$\frac{d}{dx}\psi_1(x) + \Sigma_t\psi_0(x) = \Sigma_{s0}\psi_0(x) + Q_{0,} \tag{3-15}$$

$$\frac{d}{dx}\left[\frac{1}{3}\psi_0(x)\right] + \Sigma_t\psi_1(x) = \Sigma_{s1}\psi_1(x). \tag{3-16}$$

In Eq. 3-16, ψ_1 can be solved for and substituted into Eq. 3-15. This provides a single second-order differential equation that is only dependent upon ψ_0:

$$-\frac{d}{dx}D_0\frac{d}{dx}\psi_0(x) + \Sigma_t\psi_0(x) = \Sigma_{s0}\psi_0(x) + Q_0, \tag{3-17}$$

where

$$D_0 = \frac{1}{3(\Sigma_t - \Sigma_{s1})},$$

$$\psi_1(x) = -D_0\frac{d}{dx}\psi_0(x).$$

If the Σ_{s1} term is assumed to be diagonal, then the P1 equations reduce to the diffusion equation. The next section discusses the use of discontinuity factors in the P1 equations.

3.2.2 P1 Discontinuity Factors

As described in Chapter 1, the discontinuity factors in P1 are determined by using the heterogeneous surface flux from the Serpent transport calculation and the homogeneous surface flux from the P1 expressions. The homogeneous surface flux is found by spatially

discretizing Fick's Law (Eq. 1-16). If finite difference is used, then the homogeneous surface flux is found using the following expressions:

$$\psi_{0,i}^{s,-} = \psi_{0,i} - \frac{\Delta x_i}{2}\left(-\frac{\psi_{1,i}^{s,-}}{D_{0,i}}\right), \tag{3-18}$$

$$\psi_{0,i}^{s,+} = \psi_{0,i} + \frac{\Delta x_i}{2}\left(-\frac{\psi_{1,i}^{s,+}}{D_{0,i}}\right), \tag{3-19}$$

where the plus sign indicates the left surface of a given node i, and the minus side indicates the right surface. If the discontinuity factors for a given surface are defined such that they are the inverse of each other, then the heterogeneous surface flux can be eliminated from the discontinuity factor definition. By doing this, the discontinuity factor becomes a function of the homogeneous surface fluxes from either side. The expression for the P1 discontinuity factor is the following:

$$f_{i+1/2}^{-} = 1/f_{i+1/2}^{+} = \left(\frac{\psi_{0,i+1} + \frac{\Delta x_{i+1}}{2D_{0,i+1}}\psi_{1,i+1/2}}{\psi_{0,i} - \frac{\Delta x_i}{2D_{0,i}}\psi_{1,i+1/2}}\right)^{1/2}. \tag{3-20}$$

The implementation of these discontinuity factors will be discussed later in this chapter with the discretization scheme. The next section discusses the P2 equations.

3.3 P2 Equations

The second Pn approximation that is derived is the P2 equations. Unlike the odd-order Pn approximations, the even-order versions produce a discontinuous solution on material interfaces. This solution is unphysical and is often the reason that even-order Pn solutions are not typically used. However, in cores such as the RBWR, there are large flux gradients at material interfaces. Since the traditional methodology introduces a discontinuity factor that causes the flux to be discontinuous on material interfaces, the expectation is that the natural discontinuity from P2 should improve on the P1 solution.

3.3.1 P2 Derivation

From Eq. 3-12 the three first-order differential equations that represent the P2 planar geometry approximation are the following:

$$\frac{d}{dx}\psi_1(x) + \Sigma_t\psi_0(x) = \Sigma_{s0}\psi_0(x) + Q_0, \tag{3-21}$$

$$\frac{d}{dx}\left[\frac{1}{3}\psi_0(x) + \frac{2}{3}\psi_2(x)\right] + \Sigma_t\psi_1(x) = \Sigma_{s1}\psi_1(x), \tag{3-22}$$

$$\frac{d}{dx}\left[\frac{2}{5}\psi_1(x)\right] + \Sigma_t\psi_2(x) = \Sigma_{s2}\psi_2(x). \tag{3-23}$$

In Eq. 3-23, ψ_2 is solved for in terms of ψ_1:

$$\psi_2(x) = -\frac{2}{5}\frac{1}{\Sigma_t - \Sigma_{s2}}\frac{d}{dx}\psi_1(x), \tag{3-24}$$

where from Eq. 3-21, the derivative of ψ_1 is known. Substituting this expression provides ψ_2 in terms of ψ_0:

$$\psi_2(x) = -\frac{2}{5}\frac{1}{\Sigma_{t2}}\left[(\Sigma_t - \Sigma_{s0})\psi_0(x) - Q_0\right], \tag{3-25}$$

where

$$\Sigma_{t2} = \Sigma_t - \Sigma_{s2}.$$

The term ψ_2 is then replaced in Eq. 3-22 with the expression found in Eq. 3-25. This provides ψ_1 in terms of the derivative of ψ_0:

$$\psi_1(x) = -D_0\frac{d}{dx}M(x), \tag{3-26}$$

where

$$D_0 = \frac{1}{3(\Sigma_t - \Sigma_{s1})},$$
$$M(x) = \left(1 + \frac{4}{5}\frac{\Sigma_t}{\Sigma_{t2}}\right)\psi_0(x) - \frac{4}{5}\frac{Q_0}{\Sigma_{t2}}.$$

If Eq. 3-26 is substituted within Eq. 3-21, then the P2 second-order differential equation can be written as:

$$-\frac{d}{dx}D_0\frac{d}{dx}M(x) + \Sigma_t\psi_0(x) = Q(x), \tag{3-27}$$

where the scattering term has been moved into the Q term for this final expression. To solve the P2 equations, it is useful to define the second expression in Eq. 3-27 in terms of $\bar{M}$ such that the same methodology used for P1 can be employed for P2 to solve for $\bar{M}$. First, Eq. 3-27 is written in matrix notation:

$$-\frac{d}{dx}\,\bar{\bar{D}}_0\frac{d}{dx}\,\bar{M}(x) + \bar{\bar{T}}\bar{\psi}_0(x) = \bar{Q}(x), \tag{3-28}$$

where,

$$\bar{\bar{D}}_0 = diag(D_0),$$
$$\bar{\bar{T}} = diag(\Sigma_t),$$
$$\bar{Q} = \left(\bar{\bar{S}} + \lambda\bar{\bar{F}}\right)\bar{\psi}_0,$$
$$\bar{\bar{S}} = \begin{bmatrix} \Sigma_{1\leftarrow 1} & \cdots & \Sigma_{1\leftarrow G} \\ \vdots & \ddots & \vdots \\ \Sigma_{G\leftarrow 1} & \cdots & \Sigma_{G\leftarrow G} \end{bmatrix},$$
$$\bar{\bar{F}} = \begin{bmatrix} \chi_1 \\ \vdots \\ \chi_G \end{bmatrix}\begin{bmatrix} \upsilon\Sigma_{f1} & \cdots & \upsilon\Sigma_{fG} \end{bmatrix}.$$

If the vector $\bar{M}$ is rewritten as the product of some matrix $\bar{\bar{A}}$ with ψ_0, then the ψ_0 can be replaced in Eq. 3-28 to form a second-order differential equation in terms of $\bar{M}$:

$$\bar{M} = \left(\bar{\bar{I}} + \frac{12}{5}\bar{\bar{E}}\bar{\bar{T}}\right)\bar{\psi}_0 - \frac{12}{5}\bar{\bar{E}}\bar{Q} = \bar{\bar{A}}\bar{\psi}_0, \tag{3-29}$$

where

$$E = diag\left(\frac{1}{3\Sigma_{t2}}\right),$$

$$\bar{\bar{A}} = \left(\bar{\bar{I}} + \frac{12}{5}\bar{\bar{E}}\left(\bar{\bar{T}} - \bar{\bar{S}} - \lambda\bar{\bar{F}}\right)\right),$$

$$\bar{\psi}_0 = \bar{\bar{A}}^{-1}\bar{M}.$$

Using these expressions, Eq. 3-28 is formed in terms of $\bar{M}$:

$$-\frac{d}{dx}\bar{\bar{D}}\frac{d}{dx}\bar{M}(x) + \bar{\bar{T}}\bar{\bar{A}}^{-1}\bar{M}(x) = \left(\bar{\bar{S}} + \lambda\bar{\bar{F}}\right)\bar{\bar{A}}^{-1}\bar{M}. \tag{3-30}$$

One final simplification can be applied to arrive at the final form of the P2 equation:

$$-\frac{d}{dx}\bar{\bar{D}}\frac{d}{dx}\bar{M}(x) + \bar{\bar{T}}\bar{M}(x) = \bar{\bar{B}}\bar{M}(x), \tag{3-31}$$

where,

$$\bar{\bar{B}} = \left(\bar{\bar{S}} + \lambda\bar{\bar{F}} + \bar{\bar{T}}\bar{\bar{A}} - \bar{\bar{T}}\right)\bar{\bar{A}}^{-1}.$$

Equation 3-31 is in the same form as the P1 equations. This allows for the same methodology applied solving the P1 equations to be used in solving the P2 equations for $\bar{M}$. The main difference is that ψ_0 must be solved during each iteration from $\bar{M}$ and the matrices $\bar{\bar{A}}$ and B must be constructed. One unfortunate drawback to P2 is that it requires the inverse of $\bar{\bar{A}}$ to be calculated, which can be a significant increase in runtime for large problems. It should also be restated that solutions of the P2 equations are naturally discontinuous on the interface of nodes. This discontinuity is based purely on the cross sections on either side of the interface and is dependent upon Σ_t and Σ_{t2}. For this reason, P2 is rarely used for LWR applications.

However, P2 may be appropriate for reactors such as the RBWR that feature significant material discontinuities and steep gradients in the flux. The use of discontinuity factors in P2 is discussed in the next section.

3.3.2 P2 Discontinuity Factors

To determine the discontinuity factors used in the P2 equations, a similar approach to P1 is applied. Due to the change of variables introduced in the P2 equations, continuity of the vector $\overline{M}$ is ensured instead of continuity of flux. Therefore the surface quantities of $\overline{M}$ from either side of a given surface are approximated using finite difference from the following expressions:

$$M_i^{s,-} = M_i - \frac{\Delta x_i}{2}\left(-\frac{\psi_{1,i}^{s,-}}{D_{0,i}}\right), \tag{3-32}$$

$$M_i^{s,+} = M_i + \frac{\Delta x_i}{2}\left(-\frac{\psi_{1,i}^{s,+}}{D_{0,i}}\right). \tag{3-33}$$

The same procedure performed for P1 can then be applied for P2 based on the vector $\overline{M}$ for the discontinuity factor. Once more the discontinuity factors for a given surface are defined as the inverse of each other:

$$f_{i+1/2}^{-} = 1/f_{i+1/2}^{+} = \left(\frac{\mathrm{M}_{i+1} + \frac{\Delta x_{i+1}}{2D_{0,i+1}}\psi_{1,i+1/2}}{\mathrm{M}_i - \frac{\Delta x_i}{2D_{0,i}}\psi_{1,i+1/2}}\right)^{1/2}. \tag{3-34}$$

The only difference between the discontinuity factors from P1 and P2 is the use of ψ_0 and $\overline{M}$. The next Pn approximation studied in this chapter is P3, which is presented next.

3.4 P3 Equations

The final Pn method discussed in this thesis is the P3 approximation which should provide an even more accurate transport result. However, this increase in accuracy comes at a computational cost since, for any value of n (the n-th order approximation, n=1 for P1, n=2 for

P2, etc.), the number of unknowns in the Pn approximation in 1-D is (n+1)G. This means that the cost in solving the Pn equations grows linearly as a function of n.

3.4.1 P3 Derivation

From Eq. 3-12 the four first-order differential equations that represent the P3 planar geometry approximation can be written as:

$$\frac{d}{dx}\psi_1(x) + \Sigma_t\psi_0(x) = \Sigma_{s0}\psi_0(x) + Q_0, \tag{3-35}$$

$$\frac{d}{dx}\left[\frac{1}{3}\psi_0(x) + \frac{2}{3}\psi_2(x)\right] + \Sigma_t\psi_1(x) = \Sigma_{s1}\psi_1(x), \tag{3-36}$$

$$\frac{d}{dx}\left[\frac{2}{5}\psi_1(x) + \frac{3}{5}\psi_3(x)\right] + \Sigma_t\psi_2(x) = \Sigma_{s2}\psi_2(x), \tag{3-37}$$

$$\frac{d}{dx}\left[\frac{3}{7}\psi_2(x)\right] + \Sigma_t\psi_3(x) = \Sigma_{s3}\psi_3(x). \tag{3-38}$$

Using the following definitions for the zeroth and second moment diffusion coefficients, as well as the first and third moment angular surface flux, the four first-order differential equations are combined to arrive at two second-order differential equations that are dependent upon only ψ_0 and ψ_2:

$$-\frac{d}{dx}D_0\frac{d}{dx}(\psi_0 + 2\psi_2) + (\Sigma_t - \Sigma_{s0})\psi_0 = Q, \tag{3-39}$$

$$-\frac{d}{dx}D_2\frac{d}{dx}\psi_2 + (\Sigma_t - \Sigma_{s2})\psi_2 = \frac{2}{5}[(\Sigma_t - \Sigma_{s0})\psi_0 - Q], \tag{3-40}$$

where,

$$D_0(x) = \frac{1}{3}[\Sigma_t(x) - \Sigma_{s1}(x)]^{-1},$$

$$D_2(x) = \frac{9}{35}[\Sigma_t(x) - \Sigma_{s3}(x)]^{-1},$$

$$\psi_1(x) = -D_0(x)\frac{d}{dx}[\psi_0(x) + 2\psi_2(x)],$$

$$\psi_3(x) = -\frac{5}{3}D_2(x)\frac{d}{dx}\psi_2(x).$$

These are the expressions that will be used in this chapter. However, an additional substitution can be made that simplifies the equations:

$$\Psi_0(x) = \psi_0(x) + 2\psi_2(x), \tag{3-41}$$

$$\Psi_2(x) = \psi_2(x). \tag{3-42}$$

This modifies the two second-order differential equations so that the differentials are only in terms of Ψ_0 and Ψ_2:

$$\begin{aligned} -\frac{d}{dx}D_0(x)\frac{d}{dx}\Psi_0(x) + [\Sigma_t(x) - \Sigma_{s0}(x)]\Psi_0(x) \\ = Q(x) + 2[\Sigma_t(x) - \Sigma_{s0}(x)]\Psi_2(x), \end{aligned} \tag{3-43}$$

$$\begin{aligned} -\frac{d}{dx}D_2(x)\frac{d}{dx}\Psi_2(x) + [\Sigma_t(x) - \Sigma_{s2}(x)]\Psi_2(x) \\ = \frac{2}{5}\{[\Sigma_t(x) - \Sigma_{s0}(x)][\Psi_0(x) - 2\Psi_2(x)] - Q(x)\}. \end{aligned} \tag{3-44}$$

Once the solution has converged, then the solution is converted back to its original form. In this form, if ψ_2 is zero then Eq. 3-43 reduces to the P1 equations. The next section discusses the use of discontinuity factors in the P3 equations.

3.4.2 P3 Discontinuity Factors

Discontinuity factors for P3 are based on the change of variables to Ψ_0. The surface values of Ψ_0 are determined based on the same methodology discussed in the P1 and P2 sections and are shown below:

$$\Psi_{0,i}^{s,-} = \Psi_{0,i} - \frac{\Delta x_i}{2}\left(-\frac{\psi_1^{s,-}}{D_{0,i}}\right), \tag{3-45}$$

$$\Psi_{0,i}^{s,+} = \Psi_{0,i} + \frac{\Delta x_i}{2}\left(-\frac{\psi_1^{s,+}}{D_{0,i}}\right). \tag{3-46}$$

Using these surface definitions, the definition for the discontinuity factor on either side of the surface is found to be the following:

$$f_{i+1/2}^{-} = 1/f_{i+1/2}^{+} = \left(\frac{\Psi_{0,i+1} + \frac{\Delta x_{i+1}}{2D_{0,i+1}}\psi_{1,i+1/2}}{\Psi_{0,i} - \frac{\Delta x_i}{2D_{0,i}}\psi_{1,i+1/2}}\right)^{1/2}. \tag{3-47}$$

One important note is that the discontinuity factor defined above is only applied to the zeroth moment equation. No discontinuity factors are used for the second moment equation. This is discussed further in a later section dealing with enforcing second and third moment continuity. The next section discusses the final method, which implements the use of Eddington factors for Quasi-diffusion theory.

3.5 Eddington Factors and Quasidiffusion

The method of using Eddington factors for nuclear applications was first investigated by Gol'din [57] [58] [59] [60]. This concept explored adding a transport correction term to a lower-order method as an acceleration scheme [5]. However, these methods have had limited practical application in modern reactor physics. One of the reasons is that Eddington factors used as an acceleration scheme do not ensure the same angular flux as the transport solution.

3.5.1 Eddington Factors in the Neutron Transport Equation

The derivation of the Eddington factors begins again with the time-independent neutron transport equation shown previously in Chapter 1:

$$\vec{\Omega}\cdot\nabla\psi(\vec{r},E,\vec{\Omega})+\Sigma_t(\vec{r},E)\psi(\vec{r},E,\vec{\Omega}) = \int_{4\pi}d\vec{\Omega}'\int_0^{\infty}dE'\,\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\cdot\vec{\Omega})\psi(\vec{r},E',\vec{\Omega}') + \lambda\frac{\chi(E)}{4\pi}\int_{4\pi}d\vec{\Omega}'\int_0^{\infty}dE'\,\nu\Sigma_f(\vec{r},E')\psi(\vec{r},E',\vec{\Omega}'). \quad (3\text{-}48)$$

As in Eq. 1-9, the angular integrated transport equation can be written as:

$$\nabla\vec{J}(\vec{r},E)+\Sigma_t(\vec{r},E)\phi(\vec{r},E) = \int_0^{\infty}dE'\,\Sigma_{s0}(\vec{r},E'\to E)\phi(\vec{r},E') + \lambda\chi(E)\int_0^{\infty}dE'\,\nu\Sigma_f(\vec{r},E')\phi(\vec{r},E'). \quad (3\text{-}49)$$

Introducing the Eddington tensor, which is defined as follows:

$$E_{uv}=\frac{\int_{4\pi}d\,\vec{\Omega}[\Omega_u\Omega_v\psi]}{\int_{4\pi}d\,\vec{\Omega}[\psi]},\qquad u,v=x,y,z, \quad (3\text{-}50)$$

$$\vec{E}_u=E_{ux}\hat{x}+E_{uy}\hat{y}+E_{uz}\hat{z}, \quad (3\text{-}51)$$

and multiplying Eq. 3-48 by $\vec{\Omega}$ and integrating over angle, results in the following expression:

$$\left(\hat{x}\nabla\cdot\left(\phi\vec{E}_x\right)+\hat{y}\nabla\cdot\left(\phi\vec{E}_y\right)+\hat{z}\nabla\cdot\left(\phi\vec{E}_Z\right)\right)+\Sigma_t(\vec{r},E)\vec{J}(\vec{r},E) = \int_0^{\infty}d\,E'\Sigma_{s1}(\vec{r},\mathrm{E}'\to\mathrm{E})\vec{J}(\vec{r},\mathrm{E}'). \quad (3\text{-}52)$$

To develop the scattering term in Eq. 3-52, a polynomial expansion is performed on the scattering term and the following relations are used:

$$\Sigma_s\left(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}, \mathrm{E}' \to \mathrm{E}\right) = \sum_{i=0}^{\infty} \frac{2i+1}{4\pi} \Sigma_{si}(\vec{r}, \mathrm{E}' \to \mathrm{E})\, P_i\left(\vec{\Omega}' \cdot \vec{\Omega}\right), \tag{3-53}$$

$$P_i\left(\vec{\Omega}' \cdot \vec{\Omega}\right) = P_i(\mu')P_i(\mu) + 2\sum_{m=1}^{i} \frac{(i-m)!}{(i+m)!} p_i^m(\mu')p_i^m(\mu)\cos \mathrm{m}(\alpha' - \alpha), \tag{3-54}$$

$$p_i^m(\mu) = (-1)^m(1-\mu^2)^{m/2}\frac{d^m}{d\mu^m}P_i(\mu). \tag{3-55}$$

If an additional simplification is made to Eq. 3-52 with the scattering term:

$$\int_0^{\infty} d\,E'\Sigma_{s1}(\vec{r}, \mathrm{E}' \to \mathrm{E})\vec{J}(\vec{r}, \mathrm{E}') \approx \int_0^{\infty} d\,E'\Sigma_{s1}(\vec{\mathrm{r}}, \mathrm{E} \to \mathrm{E}')\vec{J}(\vec{r}, \mathrm{E}), \tag{3-56}$$

then Eq. 3-52 can be rewritten such that the current can be solved in terms of the transport cross section and the gradient of the scalar flux and Eddington factor:

$$\vec{J} = -\frac{1}{\Sigma_{tr}(\vec{\mathrm{r}}, \mathrm{E})}\left(\hat{x}\nabla \cdot \left(\phi\vec{E}_x\right) + \hat{y}\nabla \cdot \left(\phi\vec{E}_y\right) + \hat{z}\nabla \cdot \left(\phi\vec{E}_Z\right)\right). \tag{3-57}$$

This expression for the neutron current is similar to Fick's law, which was derived in Eq. 1-16. In fact, if Eq. 3-57 is multiplied and divided by a factor of 3, the only difference between Eq. 1-16 and Eq. 3-58 is the gradient term:

$$\vec{J} = -D(\vec{\mathrm{r}}, \mathrm{E})\left(\hat{x}\nabla \cdot \left(3\phi\vec{E}_x\right) + \hat{y}\nabla \cdot \left(3\phi\vec{E}_y\right) + \hat{z}\nabla \cdot \left(3\phi\vec{E}_Z\right)\right). \tag{3-58}$$

It should be noted that if the Eddington factor for all directions, regions and energy groups is one-third, then Eq. 3-58 reduces to Fick's law, which is used in the derivation of the neutron diffusion equation. When the Eddington factor in a given region and energy group is one-third for all directions, this implies that streaming is isotropic. Therefore, allowing the Eddington factor to vary from one-third allows for angular weighting. The use of this Eddington factor captures some of the transport physics that are lost through the traditional use of the diffusion equation. The final form of the diffusion equation with Eddington factors is found

when Eq. 3-58 is substituted into Eq. 3-49. This form of the diffusion equation in Eq. 3-59 is often referred to as the Quasidiffusion Method:

$$-\sum_{u\in\{x,y,z\}}\frac{\partial}{\partial u}\left(D(\vec{\mathrm{r}},\mathrm{E})\sum_{v\in\{x,y,z\}}\frac{\partial 3\phi E_{uv}}{\partial v}\right)+\Sigma_t(\vec{r},\mathrm{E})\phi = \int_0^\infty d\,E'\Sigma_s(\mathrm{E}'\to\mathrm{E})\phi(\vec{r},\mathrm{E}')+\lambda\chi(\mathrm{E})\int_0^\infty d\,E\upsilon\Sigma_f(\mathrm{E}')\phi(\vec{r},\mathrm{E}'). \tag{3-59}$$

3.5.2 Eddington Factor Calculation

The application of Eddington factors in the work here required an additional calculation from the Monte Carlo code Serpent. An additional tally that weights the angular flux was implemented to calculate the numerator of Eq. 3-50:

$$\phi_{2,uv}=\int_{4\pi}d\,\vec{\Omega}[\Omega_u\Omega_v\psi]. \tag{3-60}$$

Since there are three spatial components (x,y,z), the values for $\phi_{2,uv}$ compose a 3x3 matrix for each spatial region and energy group. Each of these quantities is tallied by tracking the directional components for each neutron when scoring the flux using the collision-estimator method:

$$\phi_{2,uv}=\begin{bmatrix}\phi_{2,xx} & \phi_{2,xy} & \phi_{2,xz}\\ \phi_{2,yx} & \phi_{2,yy} & \phi_{2,yz}\\ \phi_{2,zx} & \phi_{2,zy} & \phi_{2,zz}\end{bmatrix}. \tag{3-61}$$

In the above matrix, the off-diagonal terms are the product of the positive and negative contributions from the separate directional vectors and are often several orders of magnitude smaller than the diagonal terms. Since the diagonal values are a single directional component squared, each contribution is a positive quantity. The average Eddington factor for each region and each energy group is then found by dividing Eq. 3-61 by the scalar flux:

$$E_{uv} = \begin{bmatrix} \frac{\phi_{2,xx}}{\phi_0} & \frac{\phi_{2,xy}}{\phi_0} & \frac{\phi_{2,xz}}{\phi_0} \\ \frac{\phi_{2,yx}}{\phi_0} & \frac{\phi_{2,yy}}{\phi_0} & \frac{\phi_{2,yz}}{\phi_0} \\ \frac{\phi_{2,zx}}{\phi_0} & \frac{\phi_{2,zy}}{\phi_0} & \frac{\phi_{2,zz}}{\phi_0} \end{bmatrix}. \tag{3-62}$$

Some important properties of these Eddington factors are that they always vary between zero and one, and the diagonal components are often close to one-third while the off-diagonal components are close to zero. As noted earlier, if the diagonal terms are equal to one-third, then neutrons travel isotropically. However, if a diagonal value (xx-, yy- or zz-) is greater than one-third, then neutrons tend to stream in that direction. For example, when the zz-component is greater than one-third, then neutrons tend to stream in the axial direction, while when the xx- and yy-components are greater than one-third, neutrons tend to stream in the radial direction. The next section discusses the solution of the one-dimensional Quasidiffusion equation.

3.5.3 Quasidiffusion solution in One-Dimensional Systems

The solution of the Quasidiffusion equations begin by first rewriting Eq. 3-59 in one-dimensional form similar to the diffusion equation but with the axial component of the Eddington factor (E_{zz} term):

$$\begin{aligned} -\frac{d}{dx}\frac{1}{\Sigma_{tr}(x,E)}\frac{d}{dx}\, E_j(x,E)\phi_0(x,E) + \Sigma_t(x,E)\phi_0(x,E) \\ = \int_0^\infty dE'\Sigma_s(x,E'\to E)\phi_0(x,E') \\ + \lambda\chi(x,E)\int_0^\infty dE\nu\Sigma_f(x,E')\phi_0(x,E'). \end{aligned} \tag{3-63}$$

In one dimension, there are at least two ways to solve this equation. The first involves a change of variables similar to the method used to solve the P2 and P3 equations. Instead of solving for the flux, the product of the flux and Eddington factor is now the quantity of interest.

In order to cast Eq. 3-63 in a suitable form, each term except the first is multiplied and divided by the Eddington factor:

$$-\frac{d}{dx}\frac{1}{\Sigma_{tr}(x,E)}\frac{d}{dx}\,E_j(x,E)\phi_0(x,E)+\frac{\Sigma_t(x,E)}{E_j(x,E)}E_j(x,E)\phi_0(x,E)$$
$$=\int_0^\infty dE'\,\frac{\Sigma_s(x,E'\to E)}{E_j(x,E)}E_j(x,E)\phi_0(x,E') \qquad (3\text{-}64)$$
$$+\lambda\chi(x,E)\int_0^\infty dE\,\frac{\nu\Sigma_f(x,E')}{E_j(x,E)}E_j(x,E)\phi_0(x,E').$$

Since the Eddington factor is assumed to be a constant scalar over a given cell and energy group in one dimension, then the cross sections in the second, third and fourth terms are adjusted by dividing by the Eddington factor. This creates an equation in which all four terms are functions of the product of the Eddington factor and the flux. If Eq. 3-64 is solved for this product, the same solution scheme as used for the diffusion solution can be applied to the solution of the Quasidiffusion equations. The final result will produce the product of the average flux and Eddington factor for each region and energy group.

Alternatively, the traditional method of solving for the flux can be used by treating the Eddington factor as a discontinuity factor. If the first term of Eq. 3-63 is multiplied and divided by 3, this produces the diffusion equation with a discontinuity factor of *3E* applied to the derivative of the flux. This is different from the traditional discontinuity factor in that the value is the same on the two bounding surfaces for a given cell in a given energy group. Since the Eddington factor is usually close to one-third, the discontinuity factor of *3E* produces a value close to one. This provides a stable discontinuity factor unlike the traditional definition which is capable of becoming negative:

$$-\frac{d}{dx}D(x,E)\frac{d}{dx}\,3E_j(x,E)\phi_0(x,E)+\Sigma_t(x,E)\phi_0(x,E)$$
$$=\int_0^\infty dE'\Sigma_s(x,E'\to E)\phi_0(x,E') \qquad (3\text{-}65)$$
$$+\lambda\chi(x,E)\int_0^\infty dE\nu\Sigma_f(x,E')\phi_0(x,E').$$

Both methods produce equivalent results. However, because the work here is focused on one dimensional applications, Eq. 3-64 will be used for the solution of the Quasidiffusion equations.

3.5.4 Eddington Factor Method with Discontinuity Factors

Just like the other Pn methods discussed in the previous sections, discontinuity factors can be applied to the Quasidiffusion equation to reproduce the transport solution. The definition for the discontinuity factor when using Eddington factors is given by Eq. 3-66:

$$f_{i+1/2}^{-}=1/f_{i+1/2}^{+}=\left(\frac{E_{zz,i+1}\psi_{0,i+1}+\frac{\Delta x_{i+1}\Sigma_{tr,i+1}}{2}\psi_{1,i+1/2}}{E_{zz,i}\psi_{0,i}-\frac{\Delta x_i\Sigma_{tr,i}}{2}\psi_{1,i+1/2}}\right)^{1/2}. \qquad (3\text{-}66)$$

This expression is similar to the other Pn methods except that there is no factor of one-third so the transport cross section appears instead of the diffusion coefficient. The following section discusses the methodology implemented to solve the Pn and Quasidiffusion equations.

3.6 Solution of the Pn Equations

Each of the four methods discussed in the previous section, P1, P2, P3, and Quasidiffusion, were used to solve the RBWR assembly problem. In order to assess the effectiveness of each method for the RBWR problem, a simple finite difference discretization was used [61]. For brevity, only the discretization of the P3 equations is shown here, since it is the most complicated. Similar steps can be performed to arrive at the discretized equations for P1, P2 and Quasidiffusion.

3.6.1 Finite Difference Methodology

The numerical solution of the P3 equations begins with the mesh-centered finite difference approximation to account for the differentials in the P3 expressions. The node averaged flux is defined over a given region, resulting in the following expression for the flux:

$$\psi_{n,i} = \psi_n(x_i) = \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \psi_n(x)dx. \tag{3-67}$$

If Eqs. 3-43 and 3-44 are integrated over a single node, then the finite difference expressions for the P3 equations for a single node are described by:

$$-D_0\left[\left.\frac{d\Psi_0}{dx}\right|_{x^-_{i+\frac{1}{2}}} - \left.\frac{d\Psi_0}{dx}\right|_{x^+_{i-\frac{1}{2}}}\right] + [\Sigma_t - \Sigma_{s0}]\Psi_0\Delta x = Q\Delta x + 2[\Sigma_t - \Sigma_{s0}]\Psi_2\Delta x, \tag{3-68}$$

$$\begin{aligned} -D_2\left[\left.\frac{d\Psi_2}{dx}\right|_{x^-_{i+\frac{1}{2}}} - \left.\frac{d\Psi_2}{dx}\right|_{x^+_{i-\frac{1}{2}}}\right] + [\Sigma_t - \Sigma_{s2}]\Psi_2\Delta x \\ = \frac{2}{5}\{[\Sigma_t - \Sigma_{s0}][\Psi_0 - 2\Psi_2] - Q\}\Delta x. \end{aligned} \tag{3-69}$$

This technique assumes that there is a linear relationship between the average angular flux from the center of the node to its boundary. Using this approximation, the differentials on either side of the node of interest can be represented using the following expression:

$$\left.\frac{d\psi_n}{dx}\right|_{x^+_{i-\frac{1}{2}}} = \frac{\psi_{n,i} - \psi_{n,i-\frac{1}{2}}}{\frac{\Delta x_i}{2}}, \tag{3-70}$$

$$\left.\frac{d\psi_n}{dx}\right|_{x^-_{i+\frac{1}{2}}} = \frac{\psi_{n,i+\frac{1}{2}} - \psi_{n,i}}{\frac{\Delta x_i}{2}}. \tag{3-71}$$

With the node average flux related to the node surface flux, Eqs. 3-68 and 3-69 can then be rewritten as a system of equations that relates the node average fluxes between adjacent cells. This is achieved by preserving the current on the boundary, which enables the surface flux to be rewritten in terms of the node average fluxes on either side of the boundary:

$$-D_{i-1}\left.\frac{d\psi_n}{dx}\right|_{x^-_{i-\frac{1}{2}}} = -D_i\left.\frac{d\psi_n}{dx}\right|_{x^+_{i-\frac{1}{2}}}, \tag{3-72}$$

$$\psi_{n,i-\frac{1}{2}} = \frac{D_i\Delta x_{i-1}\psi_{n,i} + D_{i-1}\Delta_i\psi_{n,i-1}}{D_{i-1}\Delta x_i + D_i\Delta x_{i-1}}. \tag{3-73}$$

Using this definition of the surface flux, the differentials can now be rewritten in terms of the node average fluxes:

$$\left.\frac{d\psi_n}{dx}\right|_{x^+_{i-\frac{1}{2}}} \cong 2D_{n,i-1}\frac{\psi_{n,i} - \psi_{n,i-1}}{D_{n,i-1}\Delta x_i + D_{n,i}\Delta x_{i-1}}, \tag{3-74}$$

$$\left.\frac{d\psi_n}{dx}\right|_{x^-_{i+\frac{1}{2}}} \cong 2D_{n,i+1}\frac{\psi_{n,i+1} - \psi_{n,i}}{D_{n,i+1}\Delta x_i + D_{n,i}\Delta x_{i+1}}. \tag{3-75}$$

Inserting these expressions in the P3 equations results in a linear system of equations based on the node average fluxes. For the boundaries, only reflective problems are treated initially, which cause the differential on the boundary surface to be zero:

$$\begin{aligned} -D_{0,i}\left[2D_{0,i+1}\frac{\psi_{0,i+1} - \psi_{0,i}}{D_{0,i+1}\Delta x_i + D_{0,i}\Delta x_{i+1}} - 2D_{0,i-1}\frac{\psi_{0,i} - \psi_{0,i-1}}{D_{0,i-1}\Delta x_i + D_{0,i}\Delta x_{i-1}}\right] \\ + \left[\Sigma_{t,i} - \Sigma_{s0,i}\right]\Psi_{0,i}\Delta x_i = Q_{0,i}\Delta x_i + 2\left[\Sigma_{t,i} - \Sigma_{s0,i}\right]\Psi_{2,i}\Delta x_i, \end{aligned} \tag{3-76}$$

$$\begin{aligned} -D_{2,i}\left[2D_{2,i+1}\frac{\psi_{2,i+1} - \psi_{2,i}}{D_{2,i+1}\Delta x_i + D_{2,i}\Delta x_{i+1}} - 2D_{2,i-1}\frac{\psi_{2,i} - \psi_{2,i-1}}{D_{2,i-1}\Delta x_i + D_{2,i}\Delta x_{i-1}}\right] \\ + \left[\Sigma_{t,i} - \Sigma_{s2,i}\right]\Psi_{2,i}\Delta x_i = \frac{2}{5}\left\{\left[\Sigma_{t,i} - \Sigma_{s0,i}\right]\left[\Psi_{0,i} - 2\Psi_{2,i}\right] - Q_{0,i}\right\}\Delta x_i. \end{aligned} \tag{3-77}$$

The following iterative scheme was implemented to solve for the 0th- and 2nd-moment angular fluxes and eigenvalue of the P3 equations.

1. Provide an initial guess for the 2nd-moment and eigenvalue to be used as a source in the 0th-moment expression.
2. Solve the for the 0th-moment angular flux.
3. Use the 0th-moment angular flux found in step 2 as the value in the source term to solve for the 2nd-moment angular flux.
4. Solve for the updated 2nd-moment angular flux.
5. Update eigenvalue estimation using the updated fluxes from steps 2 and 4.
6. Check for convergence of the 0th- and 2nd-moment angular fluxes and eigenvalue.
7. If the solution is not converged, use the updated angular fluxes and eigenvalues as the initial guess for the next iteration and repeat steps 1-6.

This follows the traditional power iteration technique used for eigenvalue problems. It is also possible to solve for both the 0th- and 2nd-moment angular fluxes simultaneously, but it can become quite cumbersome for large scale problems with many regions and energy groups. However, the linear systems for the 0th- and 2nd-moments can be formed in such a way that the A matrices are as diagonally dominant as possible by altering the order of the region or energy structure. Illustrations of the structure of the 0th- and 2nd-moment matrices are shown in Figure 3.5 and Figure 3.6.

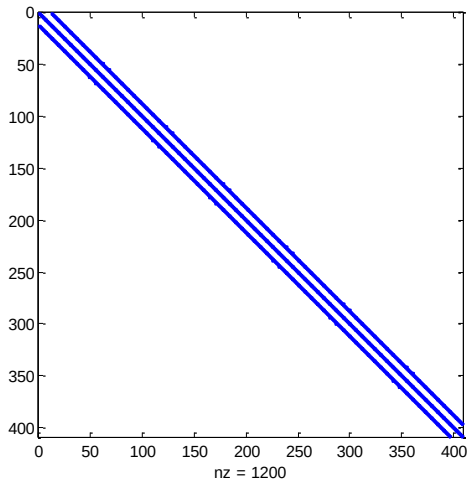

Figure 3.5: Structure of the 0th-moment matrix

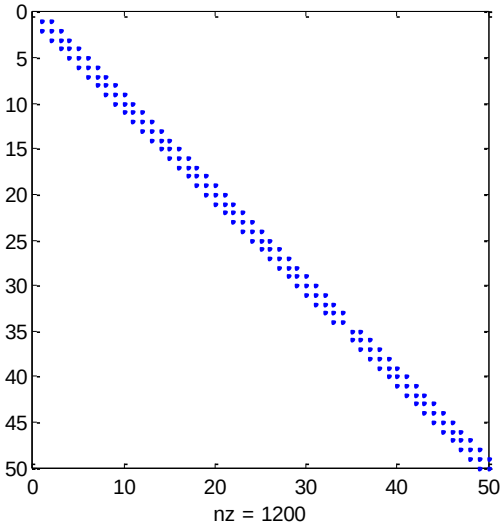

Figure 3.6: Structure of the 2nd-moment matrix

3.6.2 Discontinuity Factors in Finite Difference

For each of the Pn methods and the Quasidiffusion method, the discontinuity factors were defined based on the finite difference approximation. Due to the similarity of implementing discontinuity factors for each method with finite difference, only the P1 method will be demonstrated in this section. First, the homogeneous surface flux term is replaced in Eqs. 3-70 and 3-71:

$$f_n^{\mp}\psi_{n,i-\frac{1}{2}}^{\mp} = \psi_{n,i-\frac{1}{2}}^{Het}, \tag{3-78}$$

$$\left.\frac{d\psi_n}{dx}\right|_{x_{i-\frac{1}{2}}^{+}} = \frac{\psi_{n,i} - \frac{1}{f_n^{-}}\psi_{n,i-\frac{1}{2}}^{Het}}{\frac{\Delta x_i}{2}}, \tag{3-79}$$

$$\left.\frac{d\psi_n}{dx}\right|_{x_{i-\frac{1}{2}}^{-}} = \frac{\frac{1}{f_n^{+}}\psi_{n,i-\frac{1}{2}}^{Het} - \psi_{n,i-1}}{\frac{\Delta x_{i-1}}{2}}. \tag{3-80}$$

Next, using continuity of current, the heterogeneous surface flux is solved in terms of the average homogeneous node fluxes on either side of the surface:

$$\psi_{n,i-\frac{1}{2}}^{Het} = \frac{D_{n,i}\Delta x_{i-1}\psi_{n,i} + D_{n,i-1}\Delta x_i\psi_{n,i-1}}{\frac{1}{f_n^{+}}D_{n,i-1}\Delta x_i + \frac{1}{f_n^{-}}D_{n,i}\Delta x_{i-1}}. \tag{3-81}$$

This can be used to update the differential terms, which results in the following expression for the finite difference equations:

$$\left.\frac{d\psi_n}{dx}\right|_{x_{i-\frac{1}{2}}^{+}} = 2D_{n,i-1}\frac{\frac{1}{f_n^{+}}\psi_{n,i} - \frac{1}{f_n^{-}}\psi_{n,i-1}}{\frac{1}{f_n^{+}}D_{n,i-1}\Delta x_i + \frac{1}{f_n^{-}}D_{n,i}\Delta x_{i-1}}, \tag{3-82}$$

$$\left.\frac{d\psi_n}{dx}\right|_{x_{i+\frac{1}{2}}^{-}} = 2D_{n,i+1}\frac{\frac{1}{f_n^{+}}\psi_{n,i+1} - \frac{1}{f_n^{-}}\psi_{n,i}}{\frac{1}{f_n^{+}}D_{n,i}\Delta x_{i+1} + \frac{1}{f_n^{-}}D_{n,i+1}\Delta x_i}. \tag{3-83}$$

Using these expressions, the zeroth moment equations for an interior node in finite difference form can be written as follows:

$$-D_{0,i}\left[2D_{0,i+1}\frac{\frac{1}{f_0^+}\psi_{0,i+1}-\frac{1}{f_0^-}\psi_{0,i}}{\frac{1}{f_0^+}D_{0,i+1}\Delta x_i+\frac{1}{f_0^-}D_{0,i}\Delta x_{i+1}}\right.$$
$$\left.-2D_{0,i-1}\frac{\frac{1}{f_0^+}\psi_{0,i}-\frac{1}{f_0^-}\psi_{0,i-1}}{\frac{1}{f_0^+}D_{0,i-1}\Delta x_i+\frac{1}{f_0^-}D_{0,i}\Delta x_{i-1}}\right]+\left[\Sigma_{t,i}-\Sigma_{s0,i}\right]\Psi_{0,i}\Delta x_i \quad (3\text{-}84)$$
$$=Q_{0,i}\Delta x_i+2\left[\Sigma_{t,i}-\Sigma_{s0,i}\right]\Psi_{2,i}\Delta x_i.$$

This is typically done for the P1 and P2 equations, as well as the P3 equations for the 0th-moment. However, when applying this to the 2nd-moment equation, the 2nd-moment homogeneous flux can be zero leading to division by zero. To avoid this, a technique was applied to enforce the continuity of the 2nd- and 3rd-moments for the P3 equations. This is discussed in the following section; it eliminates the need for any discontinuity factors in the 2nd-moment equation.

3.6.3 Enforcing Second and Third Moment Continuity

The use of Monte Carlo methods for cross section homogenization introduces statistical error because of the finite number of histories. This error can become particularly important for the higher-order moments that were previously discussed. For example, an assembly problem that contains 34 axial regions with 12 energy groups and a total of 150 million histories produces relative errors for the second moment flux that range from 10% to 74% depending upon the energy group and spatial region of interest. On a single processor, this simulation would require about three days to complete. Since the error associated with a Monte Carlo quantity reduces as a function of the square root of the number of simulated histories, to achieve errors of less than 5% for all regions and all energy groups could take up to 225 times as many particles. It is important to note that this kind of simulation can be performed, but only if the computational resources are available. However, the concept of this work is to develop a set of tools that enables a user to generate group-collapsed homogenized parameters for a wide range of branch and histories, while minimizing the run time. Without access to a significant amount of computational resources, this method would not be practical from a

reactor analysis standpoint. Instead, generating higher-order quantities based on the lower-order values produced from the Monte Carlo simulation would be preferred. Unlike the higher-order moments, the lower-order moments generally have relative errors of less than 5% for a similar number of histories. The principal quantities of interest are the second-moment cell angular flux and the third-moment surface angular flux. First, we recall the second-moment equation:

$$-\frac{d}{dx}D_2\frac{d}{dx}\psi_2(x)+(\Sigma_t-\Sigma_{s2})\psi_2(x)=\frac{2}{5}[(\Sigma_t-\Sigma_{s0})\psi_0(x)-Q_0]. \tag{3-85}$$

All of the quantities on the right-hand side can be extracted from Serpent including the zeroth-moment cell-averaged fluxes. Previously, the second-moment cell-averaged fluxes were produced from tallies, but instead an alternative option is available. If the second-moment equation is discretized, we obtain a linear system in which the only unknowns are the cell-averaged second-moment fluxes:

$$\begin{aligned}-D_{2,i}\left[\left.\frac{d\psi_2}{dx}\right|_{x^-_{i+\frac{1}{2}}}-\left.\frac{d\psi_2}{dx}\right|_{x^+_{i-\frac{1}{2}}}\right]+\left(\Sigma_{t,i}-\Sigma_{s2,i}\right)\psi_{2,i}\Delta x_i \\ =\frac{2}{5}\left[\left(\Sigma_{t,i}-\Sigma_{s0,i}\right)\psi_{0,i}-Q_{0,i}\right]\Delta x_i.\end{aligned} \tag{3-86}$$

An advantage to this approach is that the second moment can be forced to be continuous by using Eqs. 3-82 and 3-83 and setting the discontinuity factors to one. This produces the following expressions for the derivative terms:

$$\left.\frac{d\psi_2}{dx}\right|_{x^+_{i-\frac{1}{2}}}=2D_{2,i-1}\frac{\psi_{2,i}-\psi_{2,i-1}}{D_{2,i-1}\Delta x_i+D_{2,i}\Delta x_{i-1}}, \tag{3-87}$$

$$\left.\frac{d\psi_2}{dx}\right|_{x^-_{i+\frac{1}{2}}}=2D_{2,i+1}\frac{\psi_{2,i+1}-\psi_{2,i}}{D_{2,i}\Delta x_{i+1}+D_{2,i+1}\Delta x_i}. \tag{3-88}$$

Using these definitions, a linear system can be used to directly solve for the second-moment fluxes in each region. Similarly, the third-moment surface fluxes can then be solved based on the second-moment quantities. The continuity of the third-moment on either side of the surface can be preserved using the following steps:

$$\psi^{+}_{3,i-\frac{1}{2}} = \psi^{-}_{3,i-\frac{1}{2}}, \tag{3-89}$$

$$\psi^{+}_{3,i-\frac{1}{2}} = -\frac{5}{3} D_{2,i} \frac{\psi_{2,i} - \psi_{2,i-\frac{1}{2}}}{\frac{\Delta x_i}{2}}, \tag{3-90}$$

$$\psi^{-}_{3,i-\frac{1}{2}} = -\frac{5}{3} D_{2,i-1} \frac{\psi_{2,i-\frac{1}{2}} - \psi_{2,i-1}}{\frac{\Delta x_{i-1}}{2}}. \tag{3-91}$$

With these relations, the third-moment surface flux can be found by eliminating the second-moment surface terms, which provides the following expression:

$$\psi_{3,i-\frac{1}{2}} = \frac{\psi_{2,i-1} - \psi_{2,i}}{\frac{3}{5}\left(\frac{\Delta x_{i-1}}{2D_{2,i-1}} + \frac{\Delta x_i}{2D_{2,i}}\right)}. \tag{3-92}$$

Since the current problem possesses reflective boundary conditions on the boundaries, the above equation is used only to solve the interior third-moment surface fluxes. This process can be used to ensure balance of the second-moment equation while also preserving continuity on the surface. This approach provides the second and third-moments using quantities from Serpent, but not directly calculated from the tallies in the Monte Carlo code. This approach can be used to develop zeroth-moment discontinuity factors for the P3 equations. From there the P3 expressions are solved using only a zeroth-moment discontinuity factor since the second-moment is continuous on the surface.

3.7 Subplane Method

In the previous sections, the finite-difference approximation was used to discretize each of the methods discussed. As the mesh size decreases, a finer representation of the spatial flux is produced, which improves the solution. Based on the methodology proposed, this would require generating cross sections for each of the regions that are represented within the model. However, creating such a cross section set with Monte Carlo can become computationally intensive as the number of meshes increases. As the region size decreases, the number of interactions taking place within each region will decrease and thus the relative errors of the group constants will increase. Since the error decreases as a function of the square root of the number of simulated particles, this can lead to large simulation times to achieve errors that are comparable to larger mesh problems. Therefore, a balance must be achieved between choosing a mesh size that accurately captures the physics of the problem and one that reduces the Monte Carlo simulation time and error.

An alternative approach is to use the subplane method [62] [63]. The concept involves generating cross sections using a coarse mesh and then using a fine mesh for the Pn or Quasidiffusion solution. This is accomplished by discretizing each of the coarse mesh regions while maintaining constant group constants within each coarse mesh region. This effectively reduces the runtime and error associated with the Monte Carlo solution for group constant generation while improving the spatial solution of the lower-order method. However, it is important to note that fixing the cross sections over a set of fine-mesh regions is typically unphysical and likely will introduce some amount of error in the simulation. If the cross sections do not vary much over a given coarse mesh, then using the subplane method should introduce a minimal amount of error. Figure 3.7 provides a simple illustration of the subplane method. The orange and blue regions represent the coarse mesh nodes where group constants are generated using Monte Carlo. Each of the coarse mesh regions are broken into ten fine mesh regions. All of the fine mesh regions that are blue use the same cross section set and the same is applied to all of the orange fine mesh regions. Though the cross sections are fixed within a given coarse mesh, the spatial flux is allowed to vary over the fine mesh.

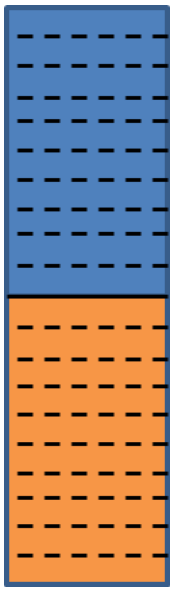

Figure 3.7: Example 2 region problem employing the subplane method

When employing discontinuity factors with the subplane method, a fixed-source problem using the same spatial discretization method is used in the lower order method to solve for the fine-mesh flux. This is because the surface flux on the coarse-mesh boundary is now estimated based on the closest fine-mesh flux instead of the coarse-mesh flux from the transport solution.

3.8 Additive Discontinuity Factors

As discussed in Section 2.3, discontinuity factors can cause numerical instabilities when performing power reactor calculations with feedback. This is sometime due to the estimate of the homogeneous surface flux becoming close to zero, in which case the magnitude of the discontinuity factor can become large since the homogeneous surface flux is in the denominator. An alternative relationship was investigated in which the discontinuity factor is based on an additive relationship of the surface fluxes instead of a multiplicative ratio. This definition of an additive discontinuity factor was examined for each of the Pn approximations and Quasidiffusion.

3.8.1 P1 Additive Discontinuity Factor

The definition of the additive discontinuity factor for P1 for the left and right sides of a surface respectively are the following:

$$f_{i+1/2}^{-} = \frac{\phi_{i+1/2}^{-} - \phi_{i+1/2}^{ref}}{\phi_i}, \tag{3-93}$$

$$f_{i+1/2}^{+} = \frac{\phi_{i+1/2}^{+} - \phi_{i+1/2}^{ref}}{\phi_{i+1}}. \tag{3-94}$$

The group indexing has been omitted for simplicity. In this form, the surface fluxes of the homogeneous and heterogeneous solutions are both located in the numerator, while the average flux in a cell is placed in the denominator as a scaling factor. This avoids potential issues with division by a small number.

There are a few differences that are immediately apparent between the additive and multiplicative definition of the discontinuity factor. First, if the heterogeneous and homogeneous surface fluxes are the same, the discontinuity factor for the additive relation is zero instead of one as in the case of the multiplicative relation. Second, the discontinuity factor for the additive relation can be negative, positive or even zero and still provide a stable solution while the multiplicative definition requires a positive discontinuity factor. If the discontinuity factor is chosen such that the value on the left side of a given surface is equal to the negative of the value on the right side of that same surface, then Eqs. 3-93 and 3-94 are combined to provide the following expression:

$$f_{i+1/2}^{-} = -f_{i+1/2}^{+} = \frac{\phi_{i+1/2}^{-} - \phi_{i+1/2}^{+}}{\phi_i + \phi_{i+1}}. \tag{3-95}$$

In this equation, the discontinuity factor is now a function of the homogeneous surface fluxes and cell averaged fluxes. The neutron current can be found in terms of the additive discontinuity factor by eliminating the surface flux terms:

$$J_{i+1/2} = \frac{\left(1 - f_{i+1/2}^{-}\right)\phi_i - \left(1 - f_{i+1/2}^{+}\right)\phi_{i+1}}{\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}}, \tag{3-96}$$

where the neutron current is now expressed as a function of the cell-averaged fluxes and additive discontinuity factors. Using both Eqs. 3-95 and 3-96 provide a new equation for the additive discontinuity factors based on cell average fluxes and the current on a given surface:

$$f_{i+1/2}^{-} = -f_{i+1/2}^{+} = \frac{\phi_i - \phi_{i+1} - \left(\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}\right) J_{i+1/2}}{\phi_i + \phi_{i+1}}. \tag{3-97}$$

Within the P1 solver, the only change from the traditional multiplicative methodology is the relation used to represent the neutron current. The next section discusses the P2 additive discontinuity factor equations.

3.8.2 P2 Additive Discontinuity Factor

The expressions used for the P2 additive discontinuity factors are very similar to the ones derived above for P1. The traditional multiplicative P2 discontinuity factor was based on the change of variable M. The same is done for the additive discontinuity factor on the left and right of a given surface:

$$f_{i+1/2}^{-} = \frac{\mathrm{M}_{i+1/2}^{-} - \mathrm{M}_{i+1/2}^{ref}}{\mathrm{M}_i}, \tag{3-98}$$

$$f_{i+1/2}^{+} = \frac{\mathrm{M}_{i+1/2}^{+} - \mathrm{M}_{i+1/2}^{ref}}{\mathrm{M}_{i+1}}. \tag{3-99}$$

Once again, the discontinuity factor on the left side of a surface is equated to the negative of the value on the right side of that same surface:

$$f_{i+1/2}^{-} = -f_{i+1/2}^{+} = \frac{\mathrm{M}_{i+1/2}^{-} - \mathrm{M}_{i+1/2}^{+}}{\mathrm{M}_i + \mathrm{M}_{i+1}}. \tag{3-100}$$

The same methodology that was applied to the P1 derivation is applied for P2 which leads to the final expressions for the neutron current and additive discontinuity factor:

$$J_{i+1/2} = \frac{\left(1 - f^{-}_{i+1/2}\right)\mathrm{M}_i - \left(1 - f^{+}_{i+1/2}\right)\mathrm{M}_{i+1}}{\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}}, \tag{3-101}$$

$$f^{-}_{i+1/2} = -f^{+}_{i+1/2} = \frac{\mathrm{M}_i - \mathrm{M}_{i+1} - \left(\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}\right) J_{i+1/2}}{\mathrm{M}_i + \mathrm{M}_{i+1}}. \tag{3-102}$$

3.8.3 P3 Additive Discontinuity Factor

For P3, the same steps used for P1 and P2 were applied. The additive discontinuity factor relations for the left and right sides of a surface are based on the change of variable Ψ:

$$f^{-}_{i+1/2} = \frac{\Psi^{-}_{i+1/2} - \Psi^{ref}_{i+1/2}}{\Psi_i}, \tag{3-103}$$

$$f^{+}_{i+1/2} = \frac{\Psi^{+}_{i+1/2} - \Psi^{ref}_{i+1/2}}{\Psi_{i+1}}. \tag{3-104}$$

Once again the additive discontinuity factors are equated such that the value on the left side of a surface is equal to the negative of the value on the right side of the same surface:

$$f^{-}_{i+1/2} = -f^{+}_{i+1/2} = \frac{\Psi^{-}_{i+1/2} - \Psi^{+}_{i+1/2}}{\Psi_i + \Psi_{i+1}}. \tag{3-105}$$

Following the same steps as the previous two sections provides the final neutron current and additive discontinuity factor relations for the P3 approximation:

$$J_{i+1/2} = \frac{\left(1 - f^{-}_{i+1/2}\right)\Psi_i - \left(1 - f^{+}_{i+1/2}\right)\Psi_{i+1}}{\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}}, \tag{3-106}$$

$$f^{-}_{i+1/2} = -f^{+}_{i+1/2} = \frac{\Psi_i - \Psi_{i+1} - \left(\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}\right) J_{i+1/2}}{\Psi_i + \Psi_{i+1}}. \tag{3-107}$$

3.8.4 Quasidiffusion Additive Discontinuity Factor

Additive discontinuity factors for the Quasidiffusion method for the left and right sides of a surface depend on the product of the Eddington factor and the flux:

$$f_{i+1/2}^{-} = \frac{E_{i+\frac{1}{2}}\psi_{i+1/2}^{-} - E_{i+\frac{1}{2}}\psi_{i+1/2}^{ref}}{E_i\psi_i}, \tag{3-108}$$

$$f_{i+1/2}^{+} = \frac{E_{i+\frac{1}{2}}\psi_{i+1/2}^{+} - E_{i+\frac{1}{2}}\psi_{i+1/2}^{ref}}{E_{i+1}\psi_{i+1}}. \tag{3-109}$$

Equations 3-108 and 3-109 can be written such that the discontinuity factor on the left side of a surface is set equal to the negative of the discontinuity factor on the right side of the same surface:

$$f_{i+1/2}^{-} = -f_{i+1/2}^{+} = \frac{E_{i+\frac{1}{2}}\psi_{i+1/2}^{-} - E_{i+\frac{1}{2}}\psi_{i+1/2}^{+}}{E_i\psi_i + E_{i+1}\psi_{i+1}}. \tag{3-110}$$

These relations are used to determine the neutron current and additive discontinuity factor relations for the Quasidiffusion method:

$$J_{i+1/2} = \frac{\left(1 - f_{i+1/2}^{-}\right)E_i\psi_i - \left(1 - f_{i+1/2}^{+}\right)E_{i+1}\psi_{i+1}}{\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}}, \tag{3-111}$$

$$f_{i+1/2}^{-} = -f_{i+1/2}^{+} = \frac{E_i\psi_i - E_{i+1}\psi_{i+1} - \left(\frac{\Delta x_i}{2D_i} + \frac{\Delta x_{i+1}}{2D_{i+1}}\right)J_{i+1/2}}{E_i\psi_i + E_{i+1}\psi_{i+1}}. \tag{3-112}$$

Chapter 4

Numerical Results

This chapter focuses on the numerical results for the various methods discussed in the previous chapter. The problem of interest is an RBWR-type assembly that consists of five separate axial fuel regions with no axial reflectors. The model was created within the Serpent Monte Carlo code using a total of 34 axial meshes. Each of the fuel regions consists of several axial meshes, which are shown in Table 4.1. Figure 4.1 illustrates the RBWR assembly from both the axial and radial view. The fuel composition for both the blanket and fissile regions are displayed in Table 4.2. A total of twelve energy groups were used to generate the cross sections and can be seen in Table 2.5. Since the RBWR is a boiling water reactor, a representative axial coolant density profile was chosen based on the previous RBWR studies and is shown in Figure 4.2.

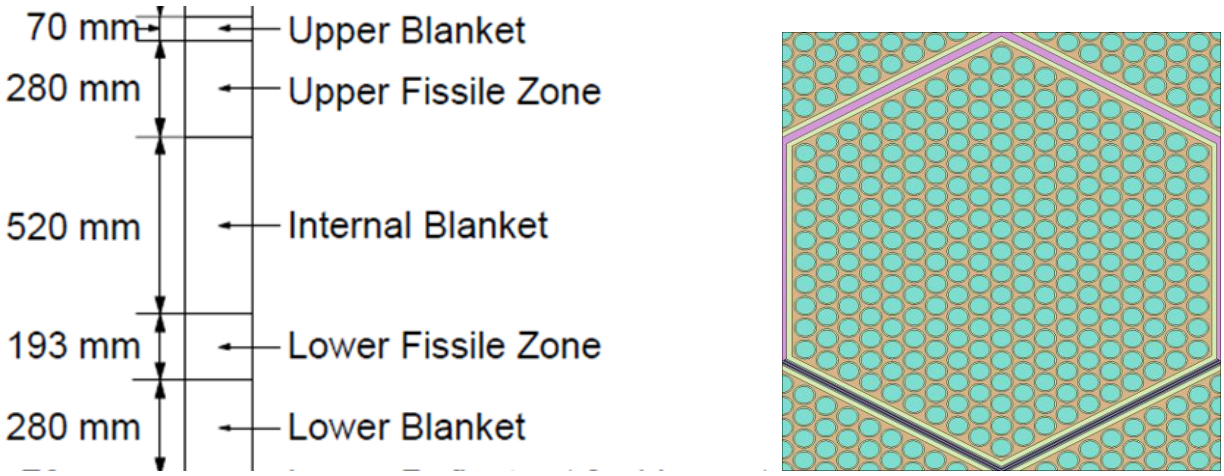

Figure 4.1: RBWR-type assembly axial view (left) and radial view (right)

Table 4.1: Axial discretization

Axial Layer	Total Length (cm)	Number of Meshes	Mesh Length (cm)
Lower Blanket	28	5	5.6
Lower Fissile	19.3	8	2.4125
Internal Blanket	52	8	6.5
Upper Fissile	28	8	3.5
Upper Blanket	7	5	1.4
Total	134.3	34	-

Table 4.2: Fuel compositions

Isotope	Blanket Number Density	Lower Fissile Number Density	Upper Fissile Number Density
U-235	4.738E-05	2.81000E-05	3.22857E-05
U-238	2.335E-02	1.38333E-02	1.59000E-02
O-16	4.679E-02	4.71667E-02	4.70000E-02
Np-237	-	4.30000E-05	3.34571E-05
Pu-238	-	2.83667E-04	2.20143E-04
Pu-239	-	4.23333E-03	3.29714E-03
Pu-240	-	3.52000E-03	2.74429E-03
Pu-241	-	4.94333E-04	3.84000E-04
Pu-242	-	4.85667E-04	3.77143E-04
Am-241	-	3.51000E-04	2.72571E-04
Am-242m	-	1.47333E-05	1.14571E-05
Am-243	-	1.27667E-04	9.96429E-05
Cm-243	-	2.22333E-06	1.72714E-06
Cm-244	-	1.02267E-04	7.93714E-05
Cm-245	-	3.19000E-05	2.47857E-05
Cm-246	-	9.86333E-06	7.69000E-06
Cm-247	-	1.74333E-06	1.35286E-06
Cm-248	-	1.23000E-06	9.57571E-07

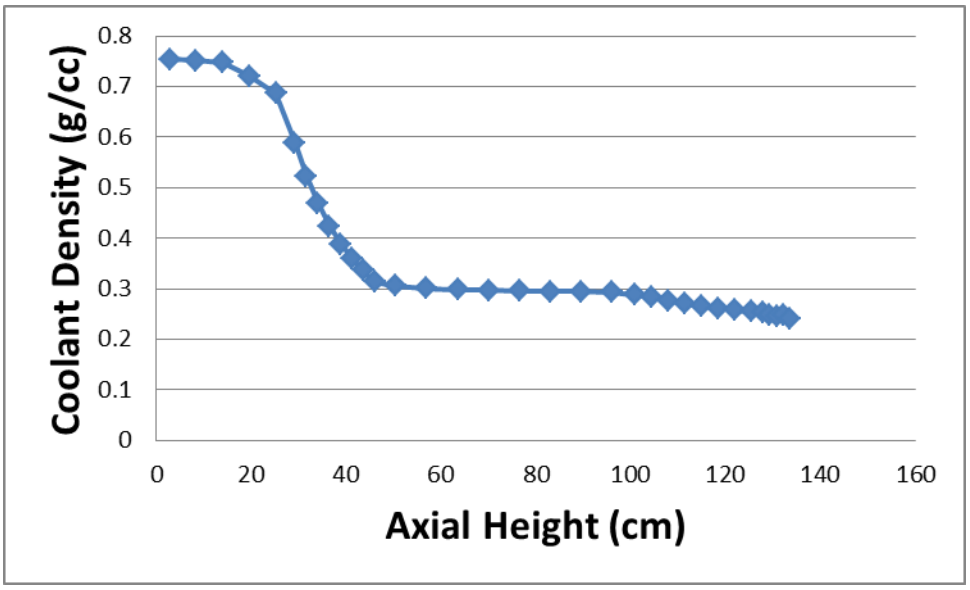

Figure 4.2: Axial coolant density profile for the RBWR-type assembly

To solve each of the methods discussed in Chapter 3, a set of cross sections was required based on the Monte Carlo calculation. A total of 34 cross section sets, one for each axial mesh, was tallied and homogenized. These cross sections were used to produce the solution for the following sections. For each calculation, the Serpent Monte Carlo case is used as the reference solution.

4.1 Finite Difference Solution to the Pn Equations for an Assembly

To assess the accuracy of the various solvers, the assembly problem discussed above was generated within Serpent. As mentioned, the assembly is similar to the RBWR assembly but without axial reflectors and with reflective boundary conditions on all sides. No control rod was inserted, and each of the five main axial regions (lower blanket, lower fissile, internal blanket, upper fissile and upper blanket) was modeled. A total of 34 axial regions and 12 energy groups were used to homogenize the cross sections (same as discussed in Chapter 2). The fissile regions consist of MOX plutonium, while the blanket regions consist of depleted uranium.

All cross sections were generated using Serpent, and tallies were used to calculate the surface quantities as well as the higher-order angular flux moments within each region. A script was generated to solve the fixed source problem to ensure that the quantities produced from Serpent preserve neutron balance. Small corrections were made to the absorption cross section

(Section 2.1.2) due to statistical noise in the Serpent values. Using these cross sections, a P1, P2, P3 and Quasidiffusion calculation were performed using the same model. The eigenvalue comparison can be seen below in Table 4.3.

Table 4.3: Assembly eigenvalue comparison

Solver	Eigenvalue	Difference from Serpent (pcm)
Monte Carlo (Serpent)	1.04192 ± 0.00008	-
P1	1.03429	-763
P2	1.04477	285
P3	1.04209	17
Quasidiffusion	1.04238	46

The results from this analysis illustrate that P3 and Quasidiffusion provide an improved solution over P1 and P2 for the axially heterogeneous problem of the RBWR. This is expected, since P3 provides an improved estimate of the angular component and Quasidiffusion uses Eddington factors based on the angular data from Serpent. Appendix B discusses the Eddington factor values calculated for the assembly problem as well as their behavior in both fuel and non-fuel regions in more detail. The error on this calculation is quite low for P3 and Quasidiffusion, and in most situations would be acceptable. In the RBWR, the beta, or effective delayed neutron fraction, is typically around 350 pcm for full-core calculations. This is determined by summing the Monte Carlo cross section produced betas for a given cell and conditions and weighting by the fission source to determine the full-core value. Therefore, acceptable errors must be smaller than one dollars' worth of reactivity, which is 350 pcm.

In addition to the eigenvalue, the group 1, group 11 and group 12 fluxes were compared for each of the methods. Group 11 is included since group 12 alone is a small portion of the thermal flux as indicated in Figure 4.3. The number of iterations and run time were also calculated to compare the computational requirements. It is important to note that no acceleration schemes were used and that the MATLAB codes generated for each method could

be further optimized. Conversely, each method follows a similar code structure so the comparisons in iteration and run time should provide meaningful insight to the speed of the various techniques. Figure 4.4 through Figure 4.6 illustrate the group 1 (fast), group 11 (thermal) and group 12 (thermal) normalized fluxes. Additionally, Figure 4.7 through Figure 4.9 shows the relative flux errors, the absolute error divided by the Serpent flux, to the Serpent reference calculation. Table 4.4 compares the iterations and run times for the various simulations.

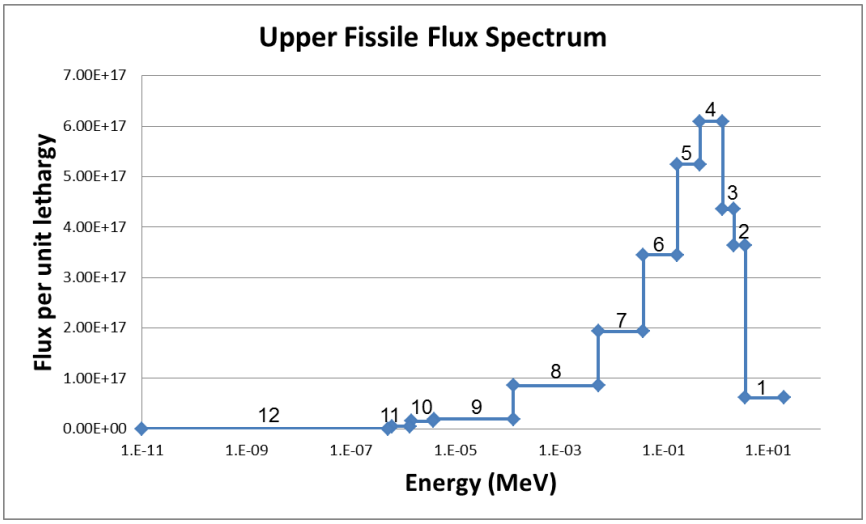

Figure 4.3: Upper fissile flux spectrum based on 12-group energy structure

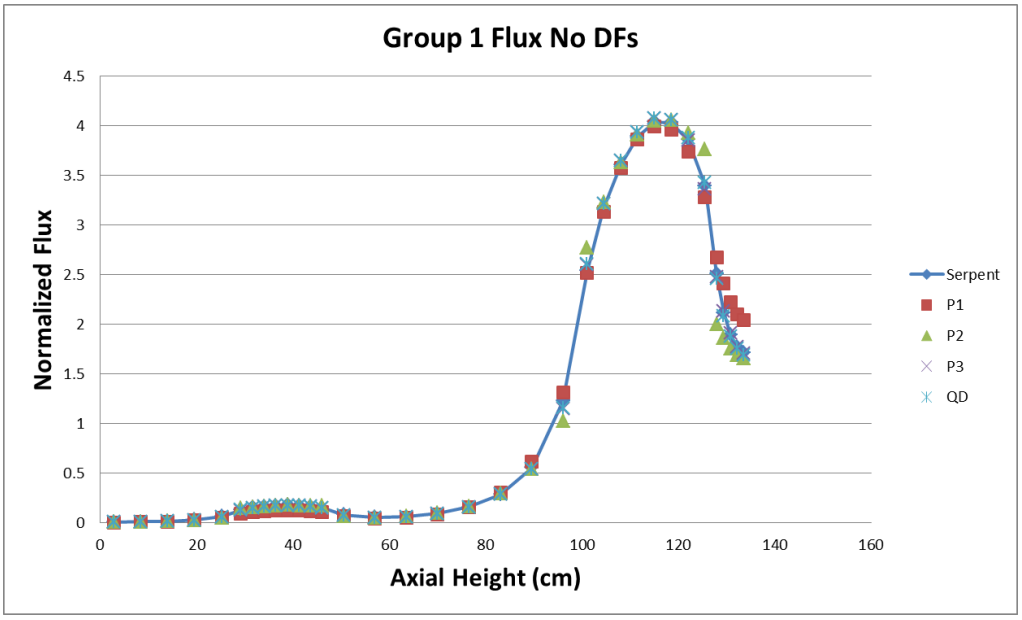

Figure 4.4: Assembly group 1 flux comparison

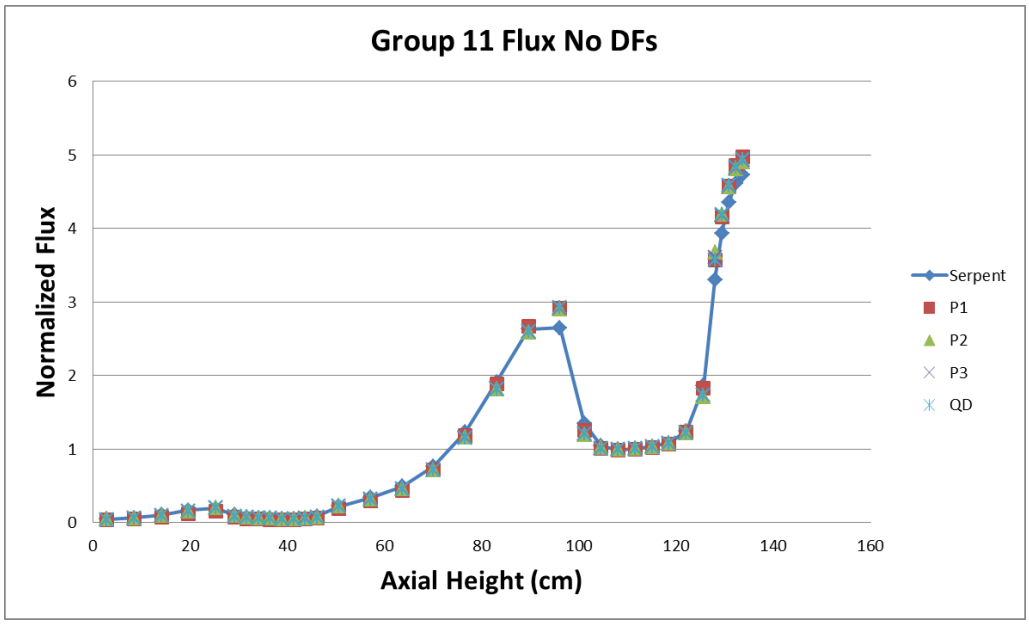

Figure 4.5: Assembly group 11 flux comparison

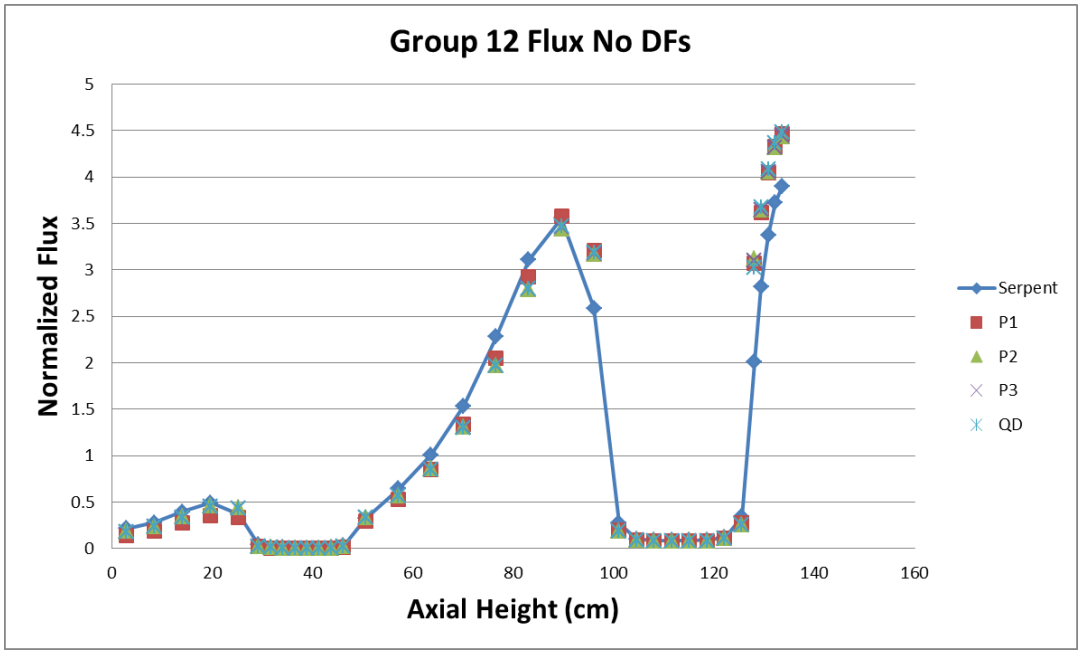

Figure 4.6: Assembly group 12 flux comparison

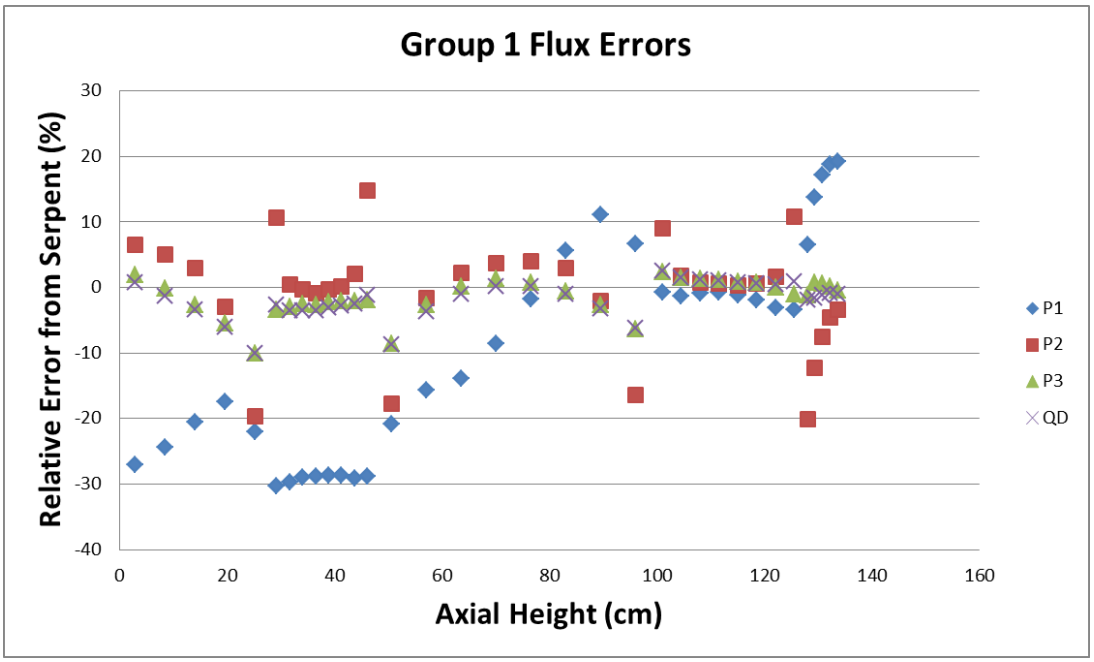

Figure 4.7: Assembly group 1 relative flux errors

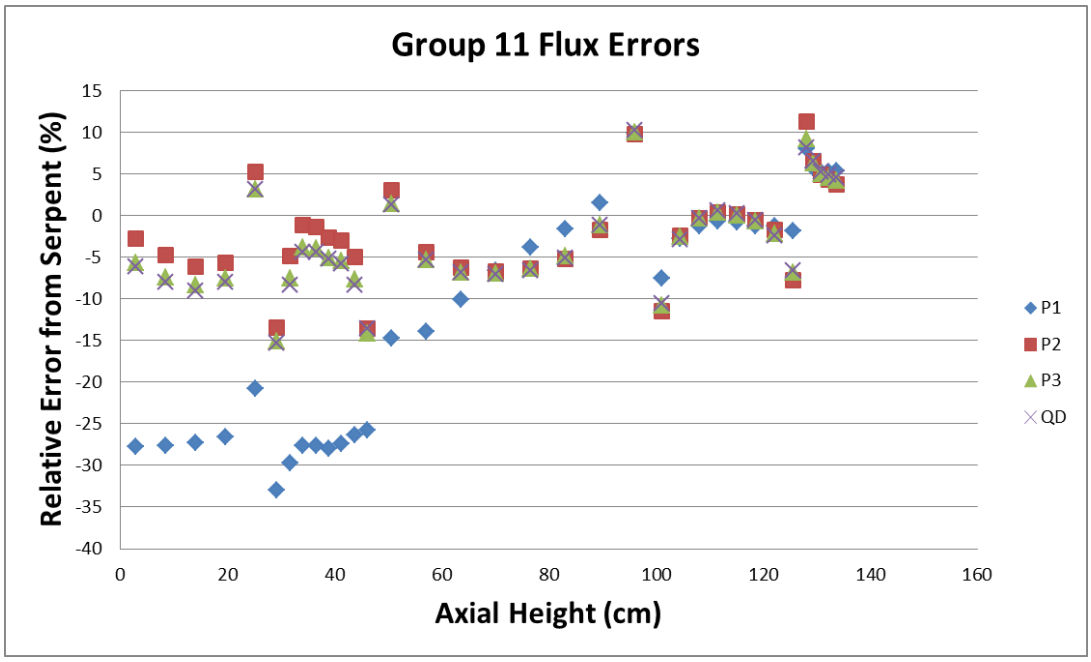

Figure 4.8: Assembly group 11 relative flux errors

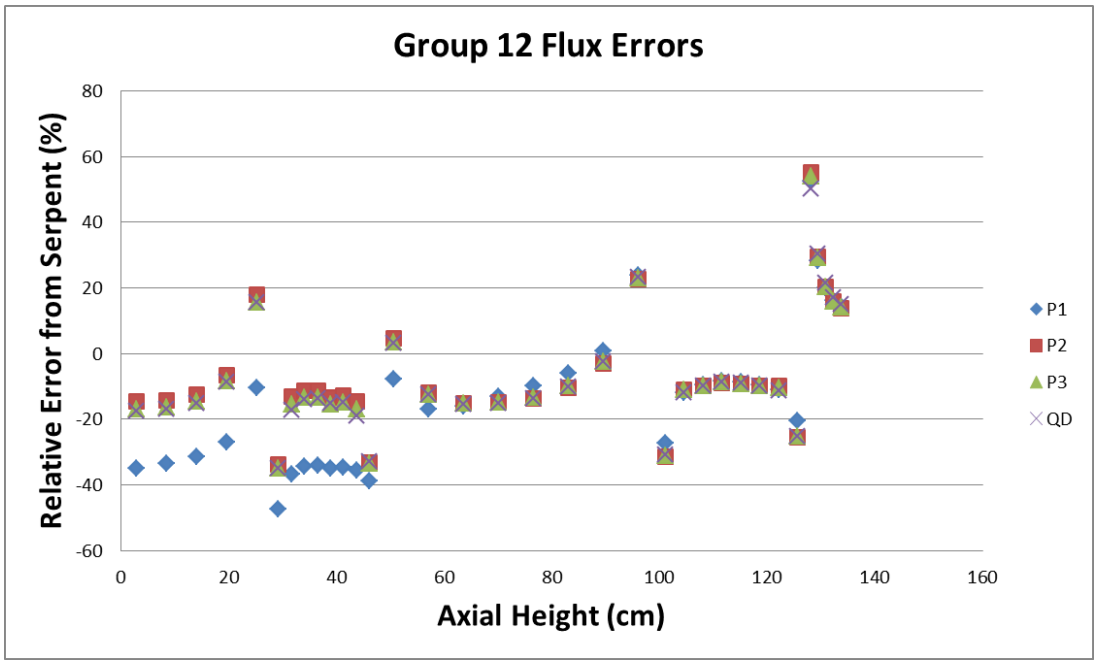

Figure 4.9: Assembly group 12 relative flux errors

Table 4.4: Comparison of iterations and run time for assembly problem

Method	Iterations	Run Time (s)
P1	744	8.98
P2	3854	676.29
P3	1121	13.63
Quasidiffusion	847	9.85

The group 1 and group 11 flux comparisons illustrate the diffusion solution has difficulty capturing the shape in the upper portions of the assembly where neutron streaming is most prevalent due to the lower coolant density. It also differs from 20-30% in the lower blanket and lower fissile regions. P2 also varies from the reference solution especially near the material interfaces. Both P3 and Quasidiffusion are close to the reference solution. However, for the group 12 flux comparison, all of the methods start to vary from the reference calculation. The group 1 through group 11 flux distributions have similar errors when compared to the reference calculation, but the group 12 result has a larger amount of error. This is attributed to

the increased uncertainties in the group 12 values due to the lower neutron population in this energy range.

One of the objectives of this analysis was to determine the most accurate methods while maintaining computational speed. The above discussion has covered the accuracy of each technique but not the cost associated with each. From Table 4.4 the P1 method required the fewest number of iterations and run time, while the P2 method took the largest number of iterations and run time. The significantly longer run time with P2 is due to calculating the inverse of the A matrix every iteration. This is an expensive calculation for a problem with 34 regions and 12 energy groups. The P3 calculation is significantly faster than P2 but still slower than P1. This simulation requires additional iterations to converge not only the eigenvalue and zeroth moment flux but also the second moment flux. Typically, the second moment flux is the last of the three parameters to reach the $1x10^{-10}$ convergence criteria. The Quasidiffusion method is similar to P1 but still requires more iterations and run time. Overall, Quasidiffusion is the second fastest since it is based on the same methodology as P1 with only the addition of the Eddington factors.

However, all four methods fail to reproduce the exact reference solution. To correct the error seen between the lower-order methods and the reference solution, the use of discontinuity factors in the various equations is discussed in the following sections.

4.2 Solution to the Pn Equations for an Assembly with Discontinuity Factors

The previous chapter described the definition of the discontinuity factors for each of the methods. Using the same assembly problem, the effect of the discontinuity factors on the solution was analyzed. Table 4.5 compares the eigenvalues of the reference solution with P1, P2, P3 and Quasidiffusion with and without discontinuity factors. Figure 4.10 through Figure 4.12 compare the group 1 (fast), group 11 (thermal) and group 12 (thermal) fluxes. The iterations and run time were also analyzed and are shown in Table 4.6.

Table 4.5: Assembly eigenvalue comparison with and without discontinuity factors

Solver	Use of Discontinuity Factors	Eigenvalue	Difference from Serpent (pcm)
Reference (Serpent)	-	1.04192	-
P1	No Discontinuity Factors	1.03429	-763
P1	With Discontinuity Factors	1.04192	0
P2	No Discontinuity Factors	1.04477	285
P2	With Discontinuity Factors	1.04192	0
P3	No Discontinuity Factors	1.04209	17
P3	With Discontinuity Factors	1.04192	0
Quasidiffusion	No Discontinuity Factors	1.04238	46
Quasidiffusion	With Discontinuity Factors	1.04192	0

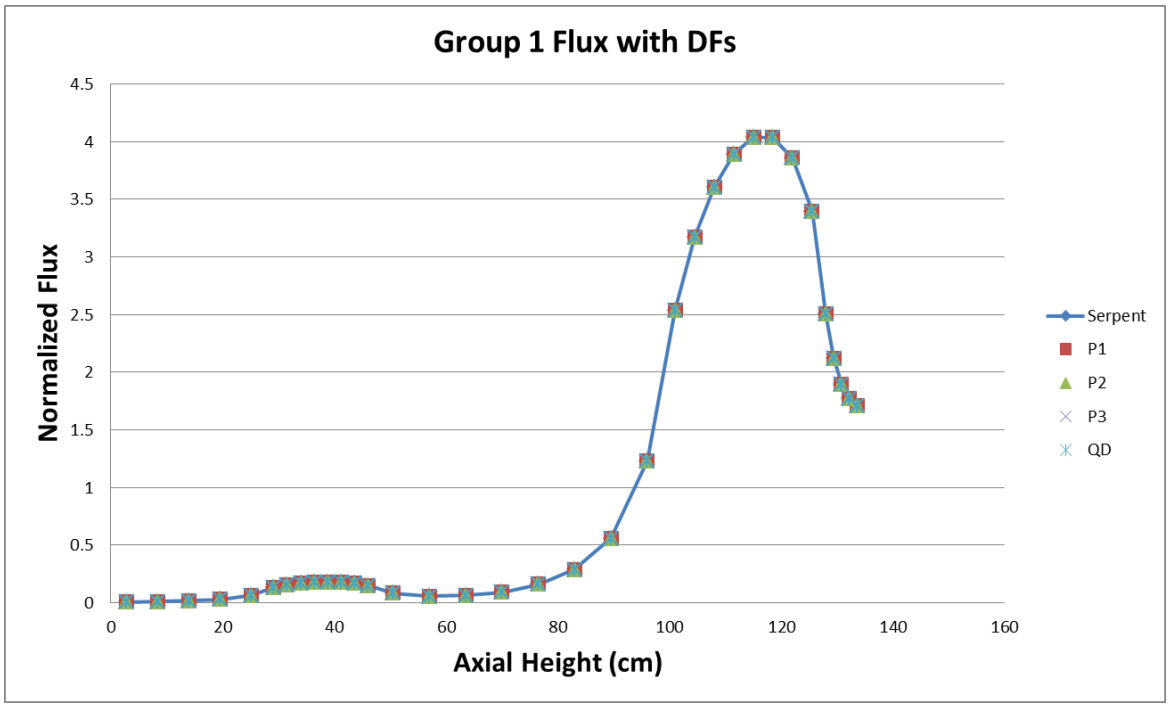

Figure 4.10: Assembly group 1 flux comparison with discontinuity factors

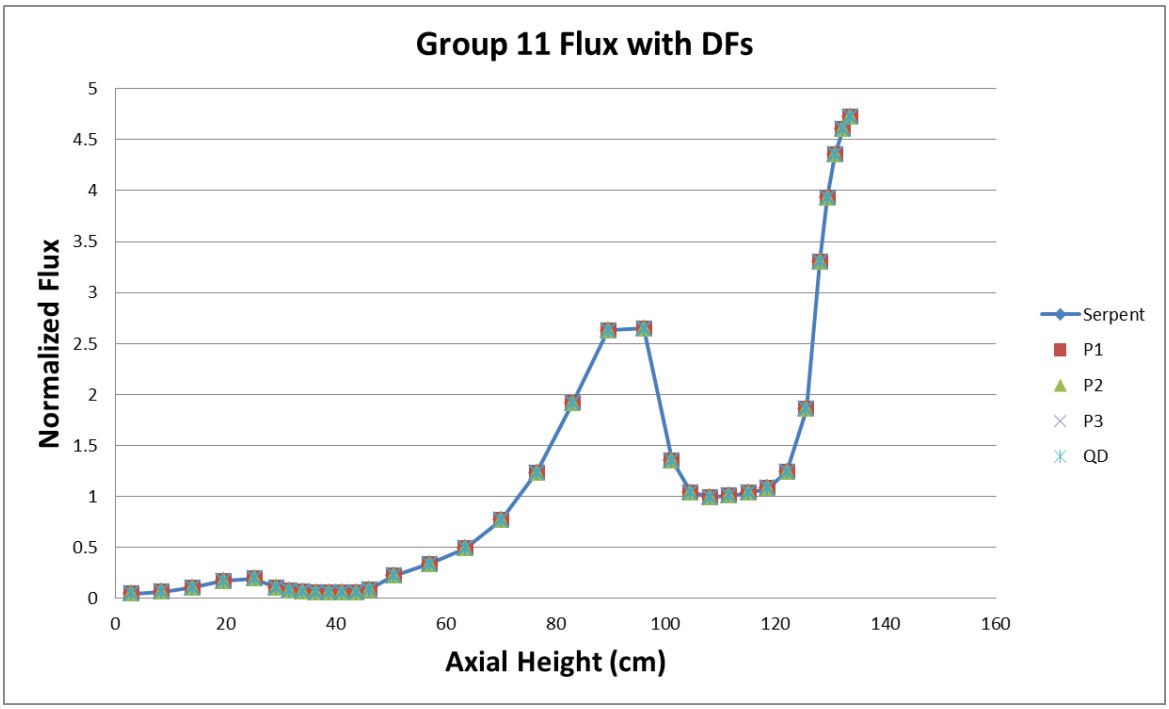

Figure 4.11: Assembly group 11 flux comparison with discontinuity factors

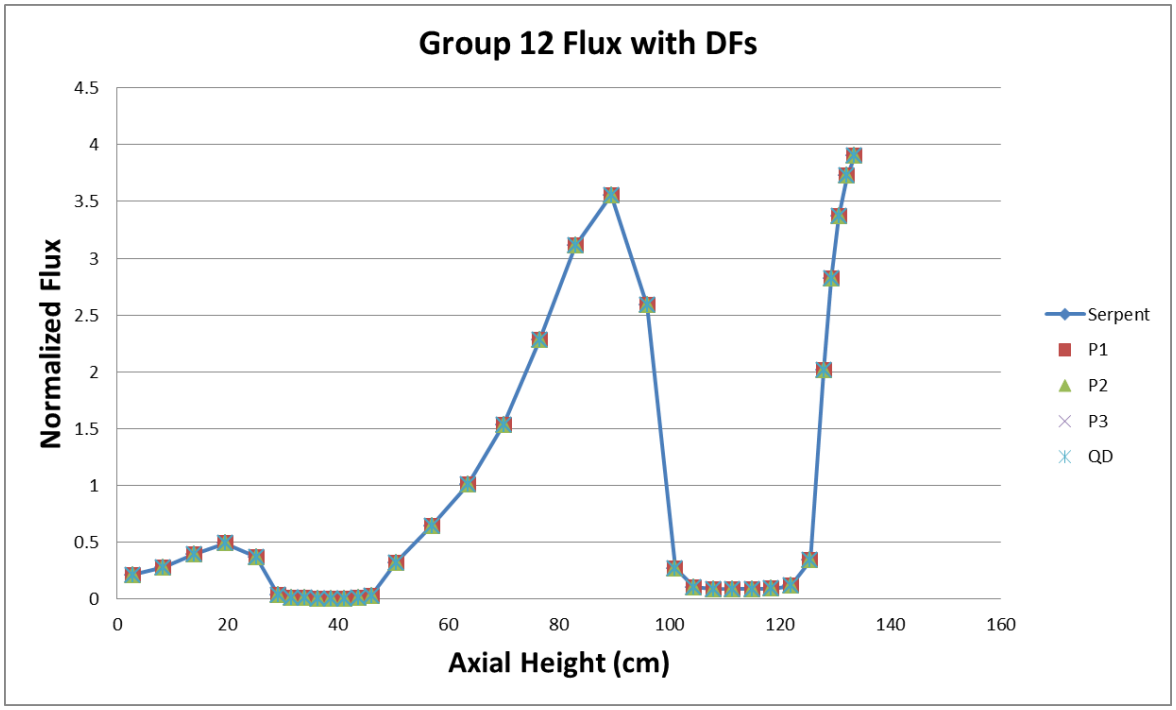

Figure 4.12: Assembly group 12 flux comparison with discontinuity factors

Table 4.6: Iterations and run time for assembly problem with discontinuity factors

Method	Iterations	Run Time (s)
P1	903	10.39
P2	3814	690.10
P3	859	13.52
Quasidiffusion	864	10.18

When using discontinuity factors, the number of iterations to converge for P1, P3 and Quasidiffusion were similar. P3 required the fewest number of iterations, while Quasidiffusion had the shortest run time. Once again P2 took the most iterations and run time to converge.

As expected, the discontinuity factors for the P1, P2, P3 and Quasidiffusion solutions reproduce the reference solution for both the eigenvalue and flux distributions. Previous work with the RBWR has also illustrated this [31]. However, the difficulty with using these discontinuity factors becomes apparent when thermal-hydraulic feedback or any other condition beside the reference solution is used. This relies upon a set of cross sections for various state conditions that account for burnup, fuel temperature, coolant density and control rod insertion. A detailed discussion and example of this issue can be found in the core analysis section of chapter 2. Typically this is not an issue; however, through previous studies on the RBWR, the variation in the axial discontinuity factor can be quite dramatic. By using P3 or Quasidiffusion, one might assume that the discontinuity factor would be closer to 1 compared to diffusion, since the initial estimate is similar to the reference solution. This is due to the P3 and Quasidiffusion flux providing a better estimate to the reference calculation. Figure 4.13 through Figure 4.16 show the discontinuity factors for P1, P2, P3 and Quasidiffusion. Additionally Figure 4.17 through Figure 4.20 show the ratio of the discontinuity factors for both sides of a given surface for all non-boundary surfaces.

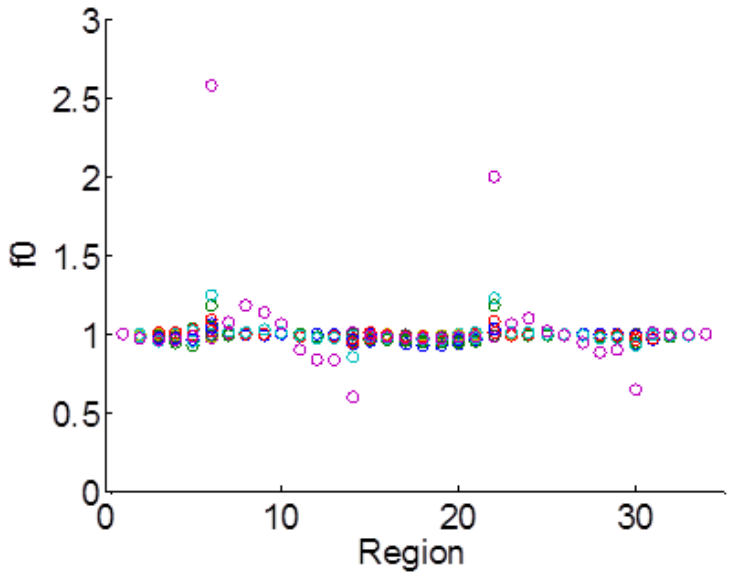

Figure 4.13: Discontinuity factors for P1 calculation

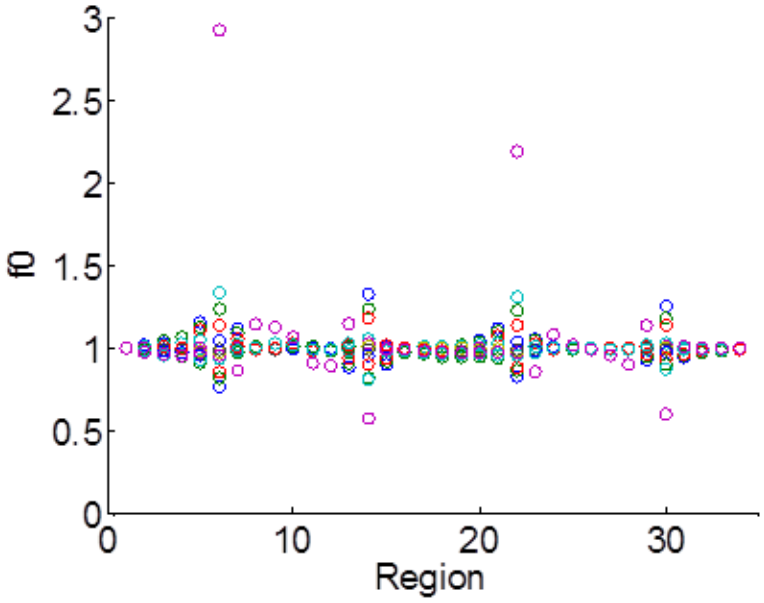

Figure 4.14: Discontinuity factors for P2 calculation

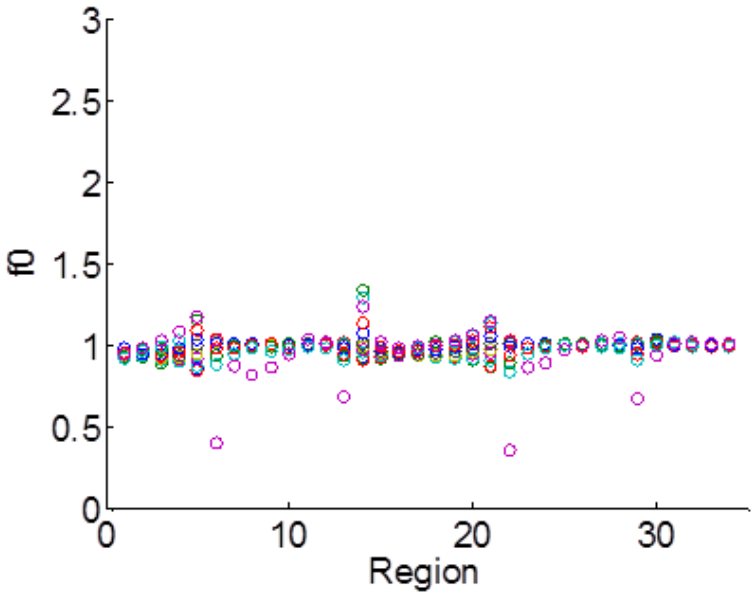

Figure 4.15: Discontinuity factors for P3 calculation

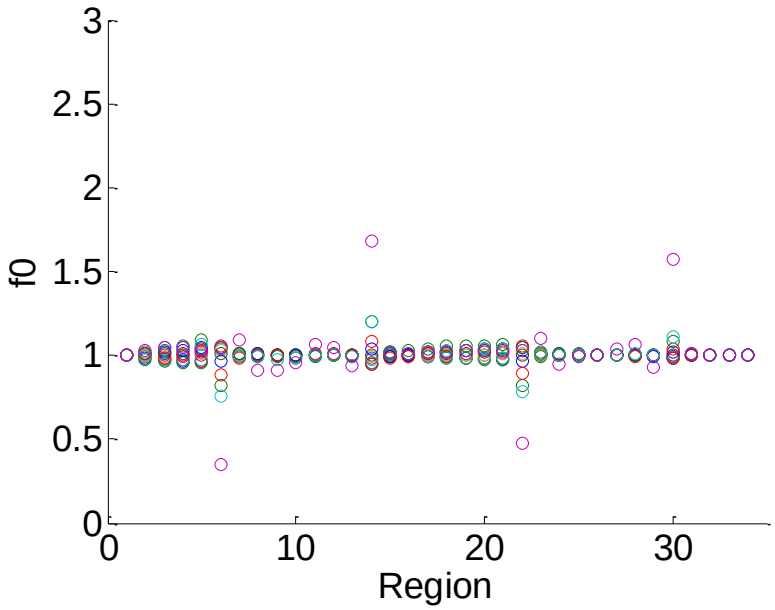

Figure 4.16: Discontinuity factors for Quasidiffusion calculation

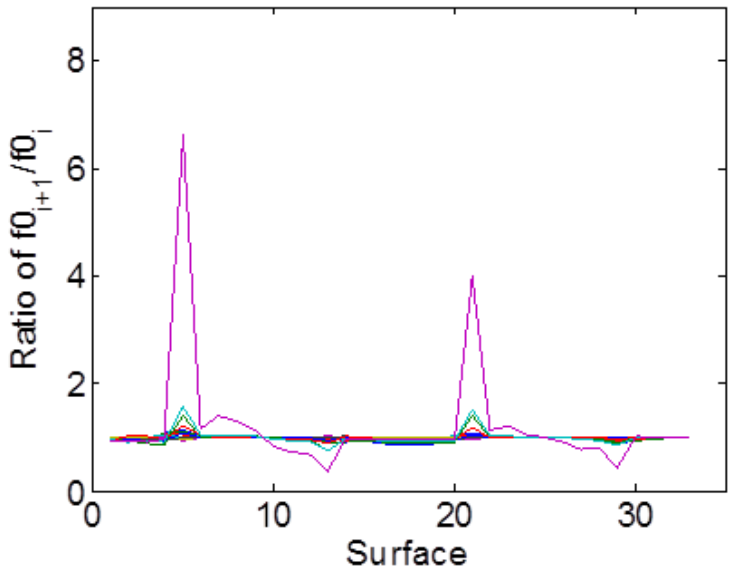

Figure 4.17: Ratio of discontinuity factors for P1 calculation

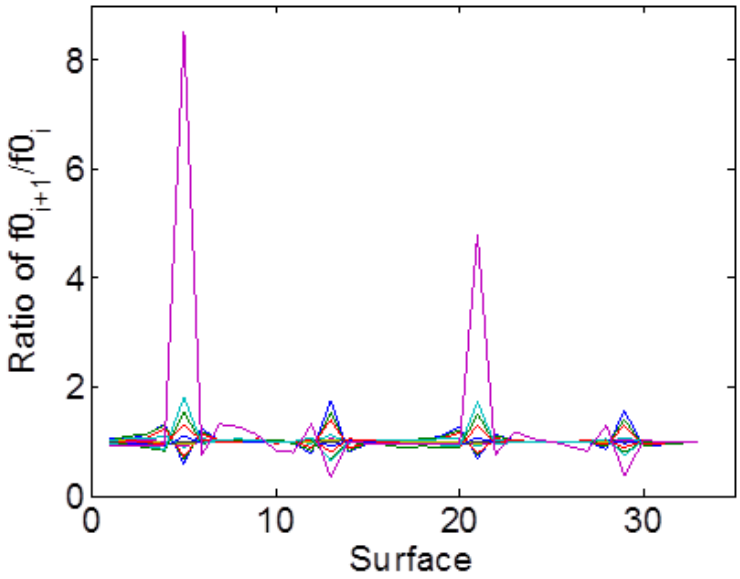

Figure 4.18: Ratio of discontinuity factors for P2 calculation

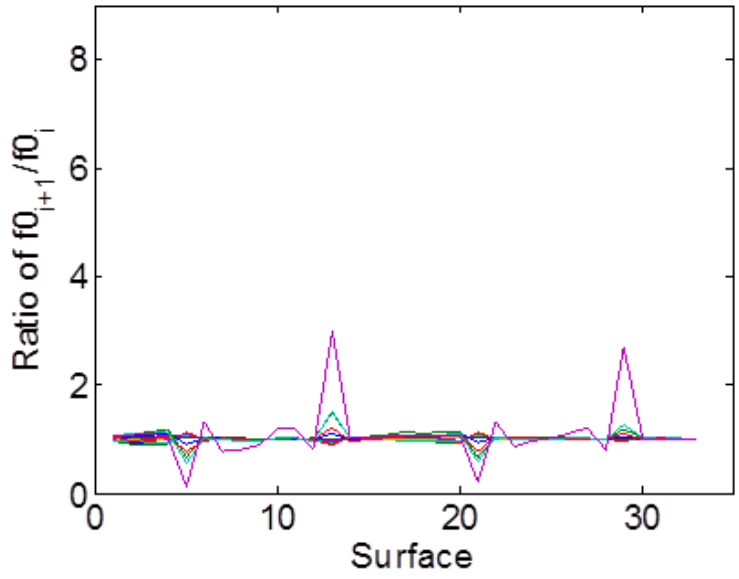

Figure 4.19: Ratio of discontinuity factors for P3 calculation

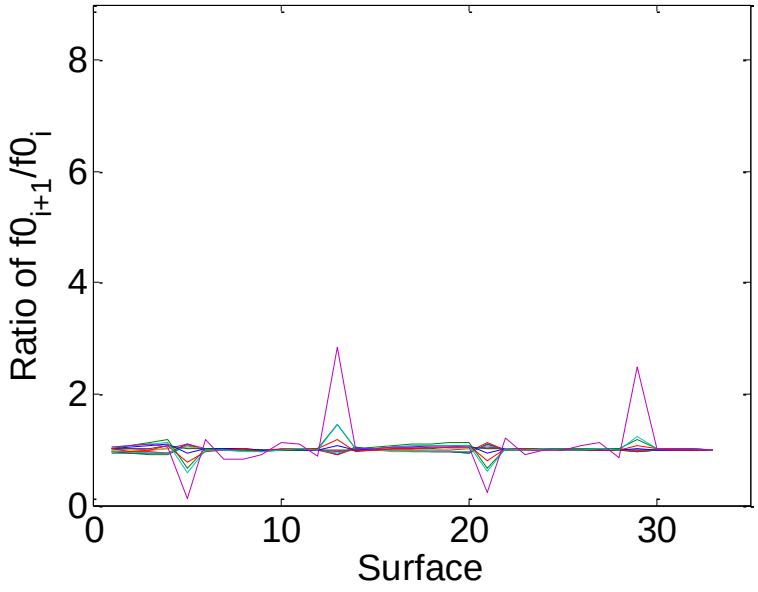

Figure 4.20: Ratio of discontinuity factors for Quasidiffusino calculation

It is important to note in the first four figures that the magnitudes of the discontinuity factors for all methods are generally similar. The majority of the differences are from the twelfth group (light purple color) on the four material interfaces. Both P3 and Quasidiffusion

have discontinuity factors that tend to be closer to one compared to P1 and P2. The ratio of the discontinuity factors also mimics this trend as both P3 and Quasidiffusion have smaller ratios compared to P1 and P2 in the twelfth energy group. In fact, these results indicate that P3 and Quasidiffusion provide similar ratios of discontinuity factors. This is important because the ratio is what is applied to the flux for each method. Overall, both P3 and Quasidiffusion show improvement over P1 and P2 for the range and ratio of discontinuity factors. One other interesting note is that the four smallest/largest discontinuity factors all reverse which side of the horizontal f0=1 axis they appeared on for Quasidiffusion compared to P1. This is also seen in Figure 4.17 and Figure 4.20 with the ratio of the discontinuity factors. The change in position of the discontinuity factors indicates that the Eddington factors are "overcorrecting" the P1 solution. This overcorrection is resolved through the introduction of the traditional discontinuity factor. However, it is possible that the range of discontinuity factors could still lead to instabilities when applied to the full-core problem. Instead, the bounding technique discussed in chapter 2 was applied to the various methods to ensure stability at the cost of introducing some error in the next section.

4.3 Solution of the Pn Equations for an Assembly with Bounded Discontinuity Factors

When a large gradient exists, such as on a material interface, the discontinuity factor can become quite small or quite large, which can cause instabilities in a coupled simulation. These values can even become negative, which leads to negative fluxes in the solution. Originally, this issue was avoided by modifying the diffusion coefficient [31], which once again reproduced the reference solution for a steady-state problem. However, this modification altered both the radial and axial aspects of the diffusion coefficient and caused instabilities in the coupled calculation. Instead, the discontinuity factors were bounded over a range to ensure stability within the simulation, while providing an improved estimate to the solution compared to a case without discontinuity factors. This was performed by generating discontinuity factors as described in the previous sections and then taking any values that lay above or below the specified range and setting them to their closest limit. The concept behind bounding the

discontinuity factors is that as the width of the bounds around 1 is increased, the result approaches the true solution. To illustrate this effect, the discontinuity factors were bounded for all four solutions. In order to provide a uniform metric, the bounds of the discontinuity factor were chosen based on the ratio to the maximum discontinuity factor for each method. If a ratio of 0 is used, that corresponds to not using a discontinuity factor, whereas a ratio of 1 corresponds to using the discontinuity factors with no bounding. Table 4.7 shows the ratio of the maximum discontinuity factor along with the produced eigenvalue, while Figure 4.21 depicts the change in eigenvalue as a function of the ratio of the maximum discontinuity factor.

Table 4.7: Eigenvalue as a function of zeroth moment discontinuity factor bounding

Ratio of Maximum DF	P1	P2	P3	Quasidiffusion
0	1.03429	1.04477	1.04209	1.04238
0.1	1.03972	1.04291	1.03958	1.03921
0.2	1.04063	1.04184	1.04049	1.04026
0.3	1.04085	1.04119	1.04080	1.04064
0.4	1.04105	1.04136	1.04102	1.04111
0.5	1.04123	1.04170	1.04122	1.04146
0.6	1.04141	1.04182	1.04141	1.04170
0.7	1.04158	1.04187	1.04158	1.04183
0.8	1.04173	1.04192	1.04175	1.04192
0.9	1.04186	1.04192	1.04185	1.04192
1.0	1.04192	1.04192	1.04192	1.04192

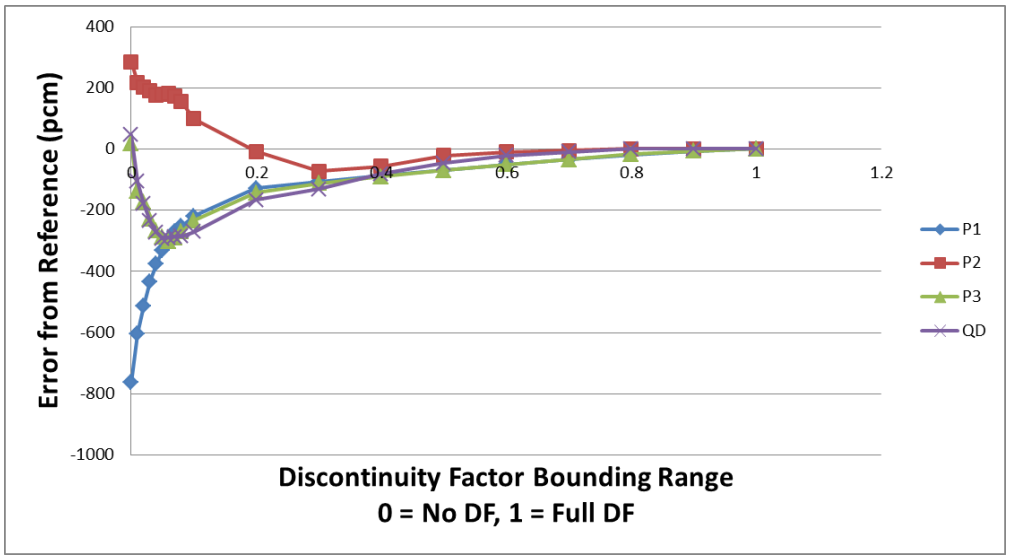

Figure 4.21: Analysis of discontinuity factor bounding on eigenvalue solution

As the width of the bounds increases for P1, the eigenvalue continuously approaches the reference solution. However, for P2 the solution overshoots and then approaches the true solution while for P3 and Quasidiffusion there is a steep increase in error before the eigenvalue shows a similar trend as P1. In fact, P1 outperforms P3 from ratios of 0.1 to 1.0 due to the non-monotonic behavior of the P3 bounded solutions. The behavior of the P3 and Quasidiffusion solutions for ratios of 0.0 to 0.1 are the only points where the solution moves away from the reference value. This trend was unexpected, since all four methods were believed to provide monotonic improvements as the width of the discontinuity factors were broadened. Due to this trend, there was likely more than one source of error involved in this calculation. The possible sources of error were:

1) Radial homogenization of group constants

2) Group collapsing from continuous energy to the prescribed 12-group structure

3) Angular approximation

4) Spatial discretization

The first two error sources are products of the cross section homogenization. For the third error, the angular dependence is expected to improve as the order of the Pn equations increase. This

leaves the fourth error source which deals with the spatial discretization of the problem. In the next section, a mesh refinement scheme was introduced to the various methods to address the behavior of Figure 4.21.

4.4 Subplane Results

The previous sections used finite difference methods to solve the P1, P2, P3 and Quasidiffusion equations. Each of these used the same axial discretization from the Monte Carlo code Serpent. If the number of meshes were increased, then the flux solution would change, which would alter the eigenvalue. As the number of meshes continues to increase, the variation in the flux and eigenvalue calculation should approach some solution. This convergence is due to the smaller variations as the mesh decreases. Since the use of discontinuity factors caused unexpected behavior when bounding, the use of a finer mesh was used to test the accuracy of the P1, P2, P3 and Quasidiffusion solutions.

To find the true converged solution, the number of meshes was varied within the various solvers. The lowest number of meshes matched the number of zones used to generate the cross sections in Serpent (34 axial zones). From there, the number of internal or fine mesh zones was increased for each region until the solution converged to a single value. For example, the first calculation used one region in each of the 34 axial zones, the second used two regions in each of the 34 axial zones, and so forth. The same cross sections were used within each fine mesh region for a single coarse zone. Table 4.8 shows the eigvenvalue as a function of the number of fine mesh regions within each zone while Figure 4.22 through Figure 4.25 illustrates these results.

Table 4.8: Eigenvalue based on number of fine mesh regions (Reference k_{eff} = 1.04192)

Number of Fine Meshes in Each Region	P1 Eigenvalue	P2 Eigenvalue	P3 Eigenvalue	Quasidiffusion Eigenvalue
1	1.03429	1.04477	1.04209	1.04238
2	1.03446	1.04470	1.04181	1.04243
3	1.03471	1.04486	1.04198	1.04264
4	1.03484	1.04494	1.04208	1.04274
5	1.03491	1.04499	1.04214	1.04280
6	1.03495	1.04501	1.04218	1.04283
7	1.03497	1.04503	1.04221	1.04285
8	1.03499	1.04504	1.04222	1.04286
9	1.03500	1.04504	1.04224	1.04287
10	1.03501	1.04505	1.04225	1.04288
11	1.03502	1.04505	1.04225	1.04289
12	1.03502	1.04506	1.04226	1.04289
13	1.03503	1.04506	1.04226	1.04289
14	1.03503	1.04507	1.04227	1.04290
15	1.03504	1.04507	1.04227	1.04290
16	1.03504	1.04507	1.04227	1.04290

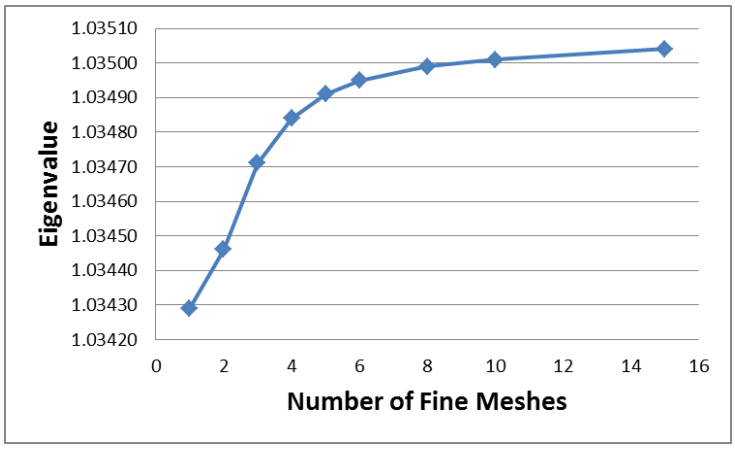

Figure 4.22: Convergence of fine mesh P1 solution

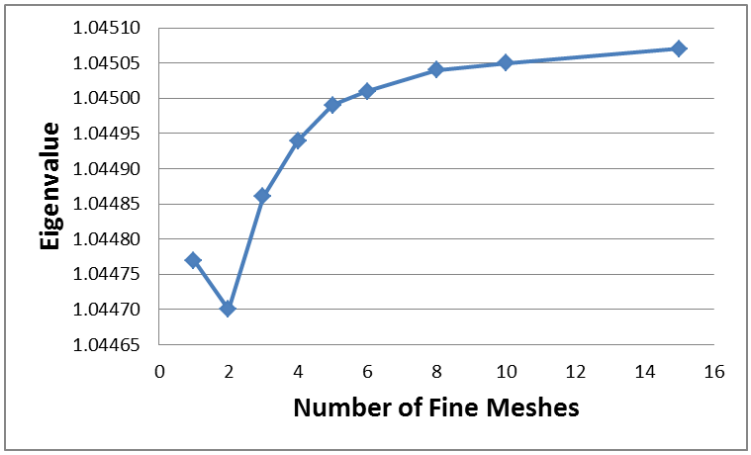

Figure 4.23: Convergence of fine mesh P2 solution

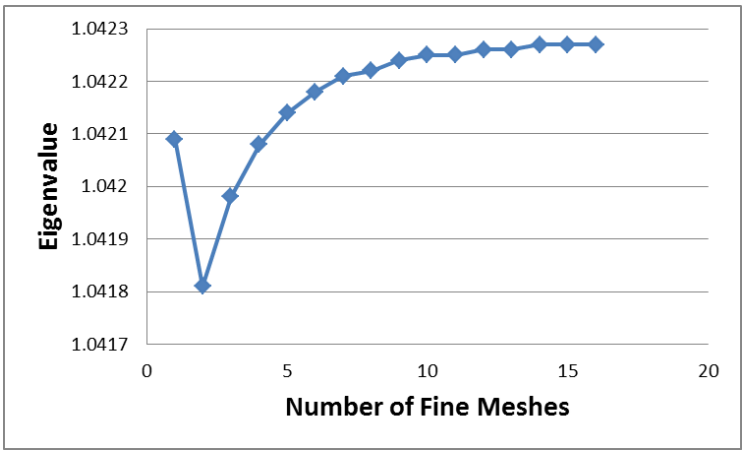

Figure 4.24: Convergence of fine mesh P3 solution

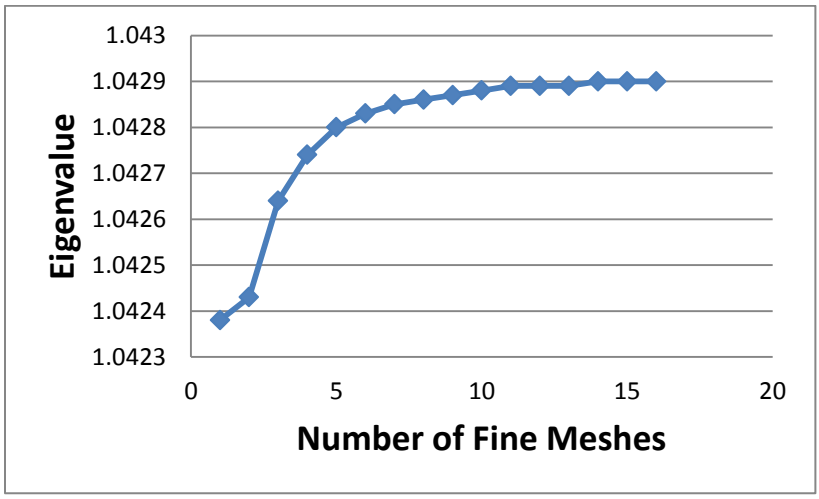

Figure 4.25: Convergence of fine mesh Quasidiffusion solution

As the number of meshes increases, the change in eigenvalue between mesh cases decreases until it converges starting at 14-15 fine meshes, depending on the method. This provides a converged solution that differs from the reference calculation by 688 pcm for P1, 315 pcm for P2, 35 pcm for P3, and 98 pcm for Quasidiffusion. These simulations do not reproduce the reference solution, but they provide the best estimate without discontinuity factors for a one dimensional heterogeneous problem similar to the RBWR.

4.5 Subplane with Discontinuity Factors for Pn

Without the use of a fine mesh, the original simulations using the same coarse mesh from Serpent were not converged spatially. The previous section determined the number of fine meshes required to spatially converge each of the methods. By performing the same bounding analysis with discontinuity factors but using the fine spatial mesh, any error source associated with the spatial discretization was eliminated. This calculation required an additional step when solving each of the equations. Prior to solving the eigenvalue problem, the fine mesh fluxes were calculated based on the coarse mesh information. When the discontinuity factors were calculated, the homogeneous surface fluxes were estimated based on the fine mesh closest to the surface instead of the average coarse mesh flux. It should be noted that discontinuity factors were only generated for the coarse mesh cell surfaces and not for each of the fine mesh cell surfaces.

For each method, a total of ten fine meshes were used within each region. Discontinuity factors were calculated based on the fine mesh fixed source problem, and the bounding treatment discussed in Section 4.3 was applied. For P3, the fixed source problem required additional treatment to handle the second moment. Each other method involved only determining the zeroth moment fine mesh flux solution, while P3 required both the zeroth and second moment solutions. An iterative scheme was implemented to converge the zeroth and second moment fine mesh flux solutions for the discontinuity factor calculation. The results are shown in Table 4.9 and Figure 4.26. Discontinuity factors for each technique, along with the ratio of discontinuity factors for each method with fine mesh treatment are illustrated in Figure 4.27 through Figure 4.34.

Table 4.9: Eigenvalue as a function of discontinuity factor bounding with fine mesh

Ratio of Maximum DF	P1 10 mesh	P2 10 mesh	P3 10 mesh	QD 10 mesh
0	1.03504	1.04505	1.04225	1.04288
0.1	1.04197	1.04533	1.04175	1.04194
0.2	1.04192	1.04461	1.04198	1.04198
0.3	1.04192	1.04338	1.04197	1.04197
0.4	1.04192	1.04220	1.04196	1.04196
0.5	1.04192	1.04194	1.04195	1.04195
0.6	1.04192	1.04193	1.04194	1.04194
0.7	1.04192	1.04192	1.04193	1.04192
0.8	1.04192	1.04192	1.04192	1.04192
0.9	1.04192	1.04192	1.04192	1.04192
1.0	1.04192	1.04192	1.04192	1.04192

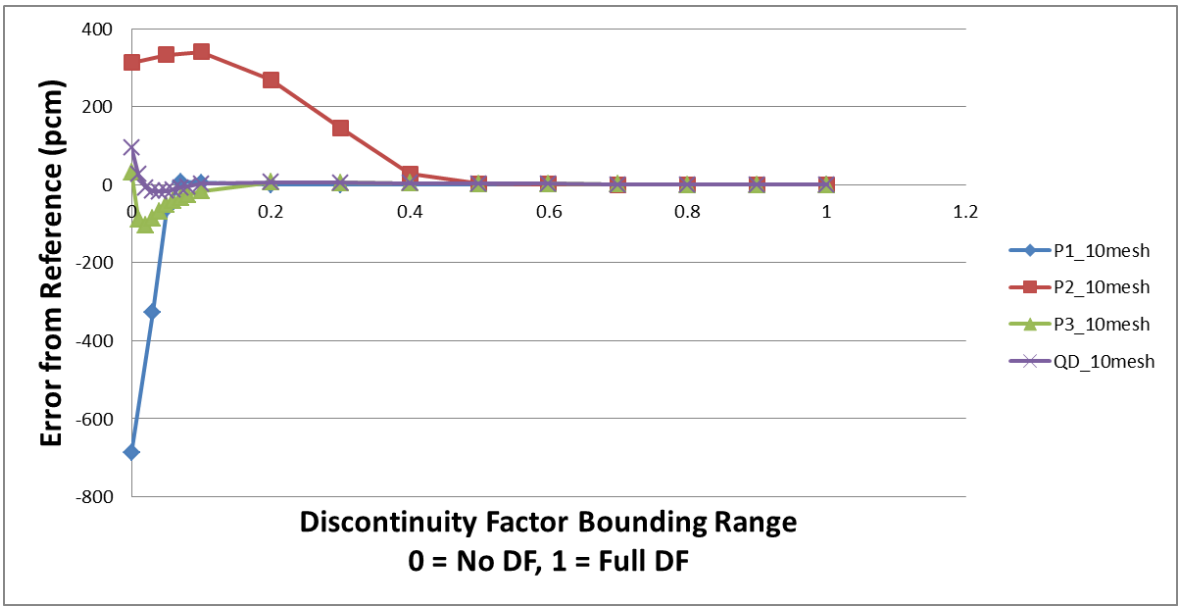

Figure 4.26: Effect of bounding discontinuity factor using 10 fine meshes

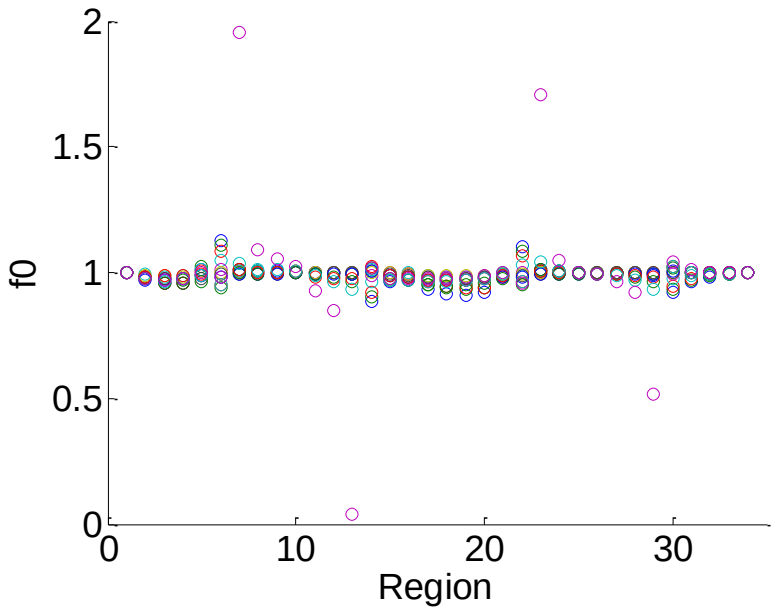

Figure 4.27: Discontinuity factors for P1 with 10 fine meshes

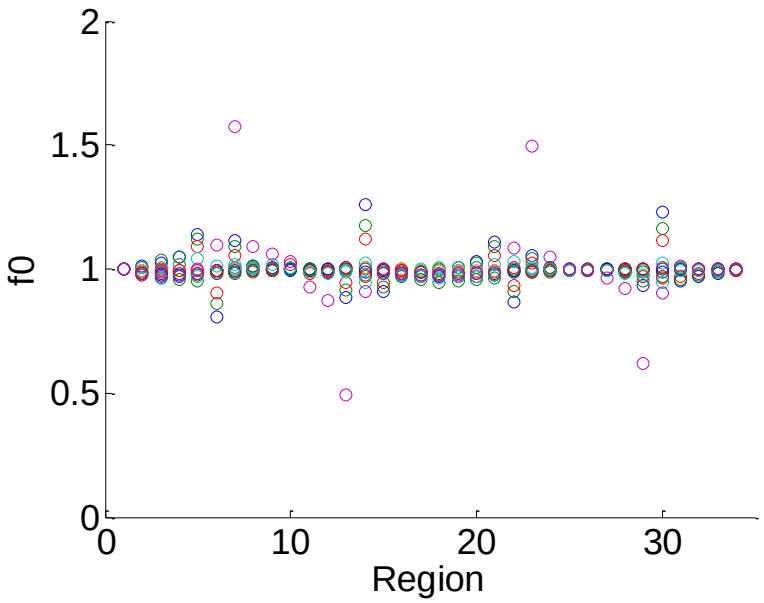

Figure 4.28: Discontinuity factors for P2 with 10 fine meshes

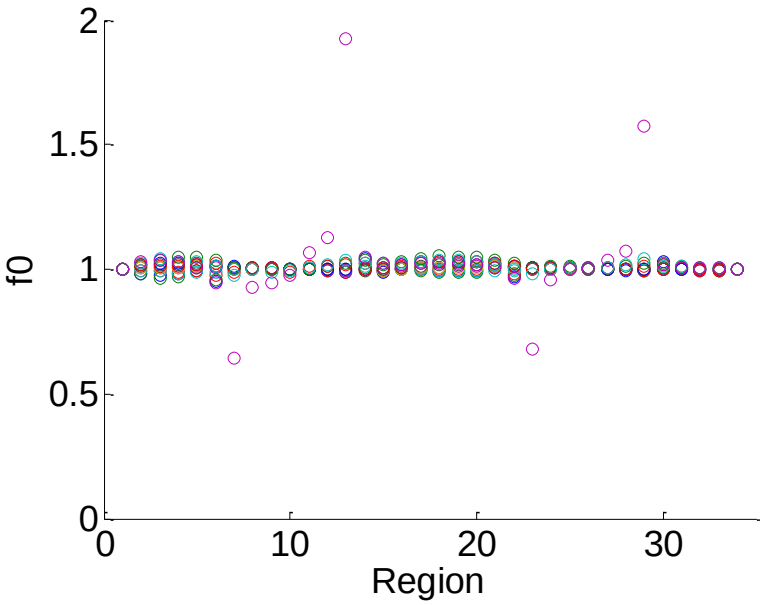

Figure 4.29: Discontinuity factors for P3 with 10 fine meshes

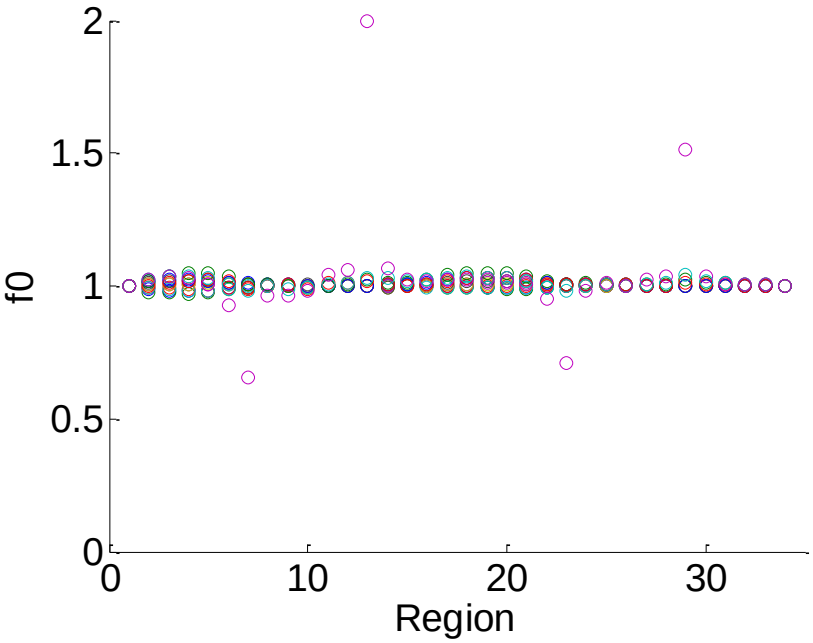

Figure 4.30: Discontinuity factors for Quasidiffusion with 10 fine meshes

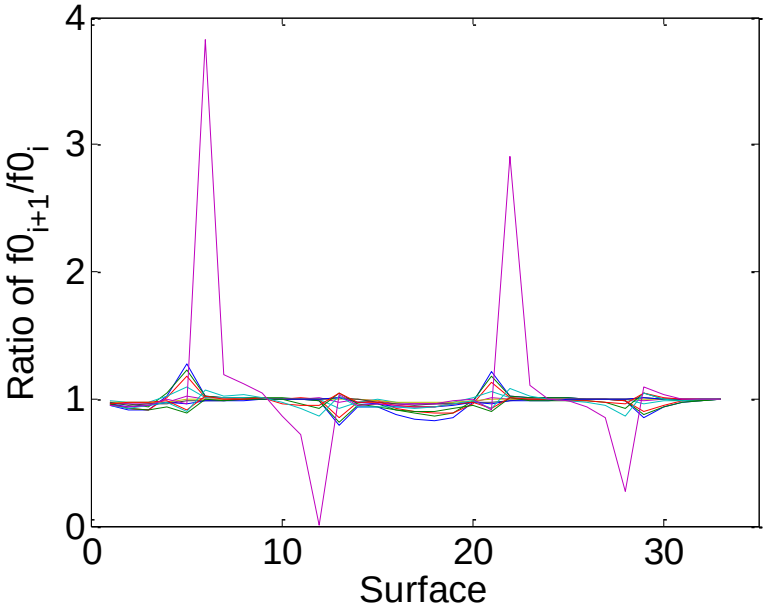

Figure 4.31: Ratio of discontinuity factors for P1 with 10 fine meshes

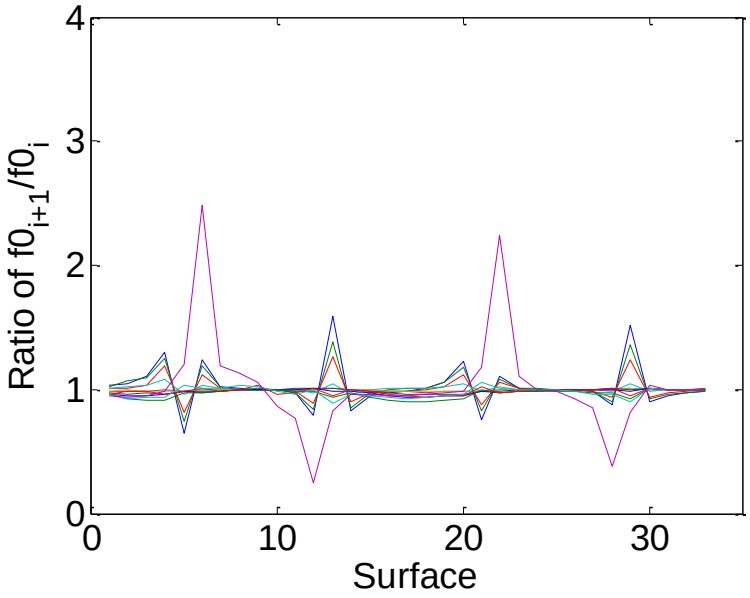

Figure 4.32: Ratio of discontinuity factors for P2 with 10 fine meshes

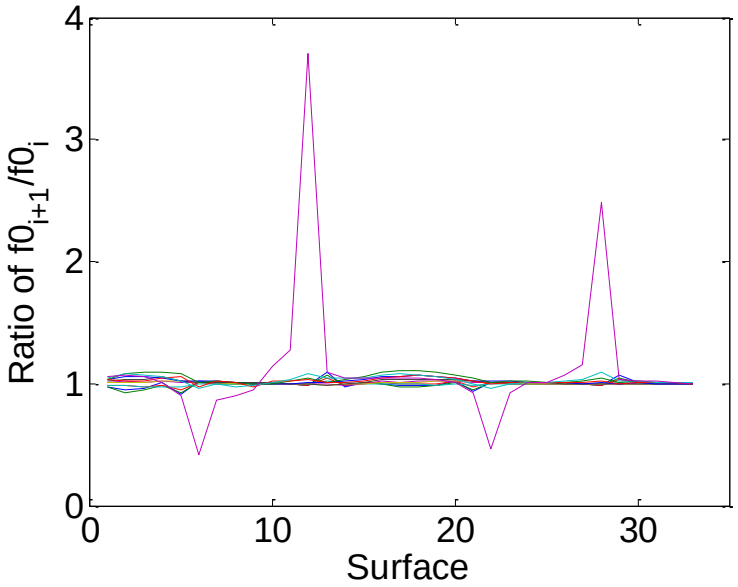

Figure 4.33: Ratio of discontinuity factors for P3 with 10 fine meshes

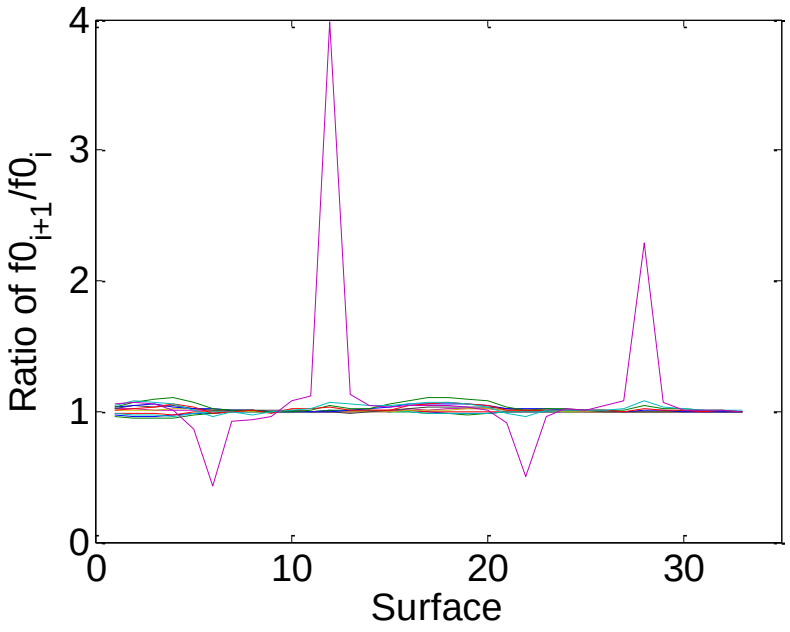

Figure 4.34: Ratio of discontinuity factors for Quasidiffusion with 10 fine meshes

With increased number of meshes applied with the discontinuity factor bounding technique, the P2 method exhibited a much more monotonic trend. There is still a slight increase in the eigenvalue for ratios less than 0.1, but from 0.1 to 1.0 the solution steadily approaches the reference value. When the fine mesh method is applied to P1 with bounding, the solution quickly converges. The method briefly overcorrects by a few pcm, but provides the reference solution with very small bounds. Both P1 and P2 with the fine mesh produce discontinuity factors closer to one for almost all points. In addition, both P3 and Quasidiffusion saw improvements with the finer mesh. For P3, the eigenvalue between 0.0 and 0.1 still increases in error but at a smaller magnitude and converges faster than the coarse mesh solution with bounded discontinuity factors. Of all the methods, Quasidiffusion had the smallest maximum error. Both P1 and Quasidiffusion were the fastest to converge within a few pcm of the reference solution. The ratio of the discontinuity factors also improved with this technique. However, there still appear to be surfaces where larger discontinuity factors continue to persist, even with a finer mesh applied to the problem. Despite this, it is clear that the spatial meshing has a large impact when bounding the discontinuity factors.

4.6 Additive Discontinuity Factors

The final method discussed in this chapter has to do with the use of additive discontinuity factors. Usually discontinuity factors are based on a multiplicative relation between the heterogeneous and homogeneous surface fluxes. Throughout this document, it has been shown that this definition can provide widely varying values that have significant impact on the stability of full core power simulations with feedback. The definitions discussed at the end of chapter 3 are demonstrated for the P1, P2, P3 and Quasidiffusion methods. Table 4.10 illustrates the results of the RBWR assembly problem with and without additive discontinuity factors. Figure 4.35 through Figure 4.38 show the magnitude of the discontinuity factors.

Table 4.10: Eigenvalues for RBWR assembly with and without additive discontinuity factors

Method	Eigenvalue	Error from Reference (pcm)
Serpent	1.04192	-
P1	1.03429	-763
P1 with Additive DF	1.04192	0
P2	1.04477	285
P2 with Additive DF	1.04192	0
P3	1.04209	17
P3 with Additive DF	1.04192	0
QD	1.04238	46
QD with Additive DF	1.04192	0

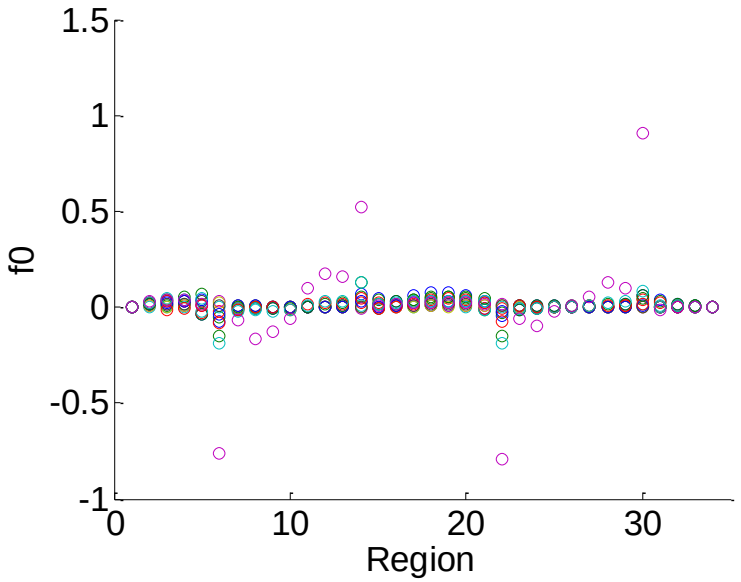

Figure 4.35: P1 additive discontinuity factors

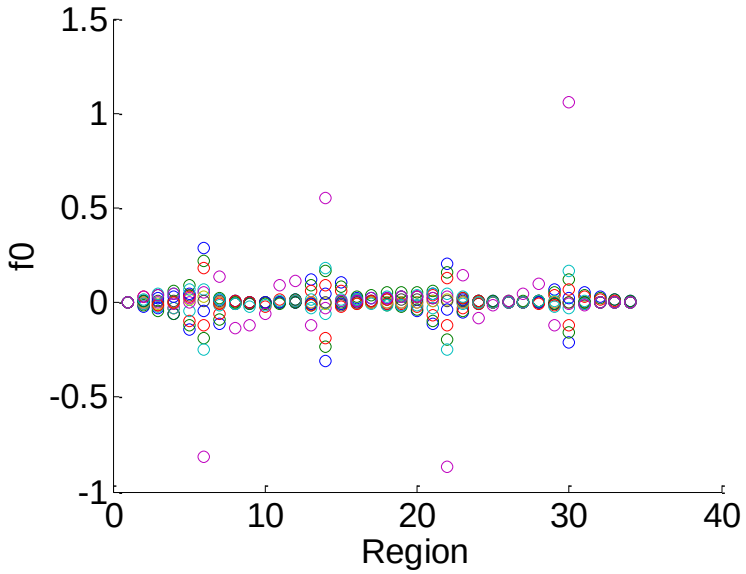

Figure 4.36: P2 additive discontinuity factors

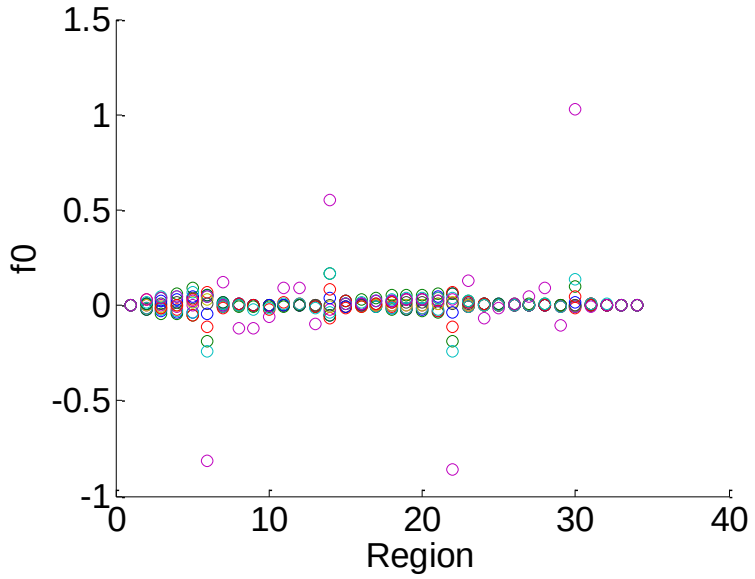

Figure 4.37: P3 additive discontinuity factors

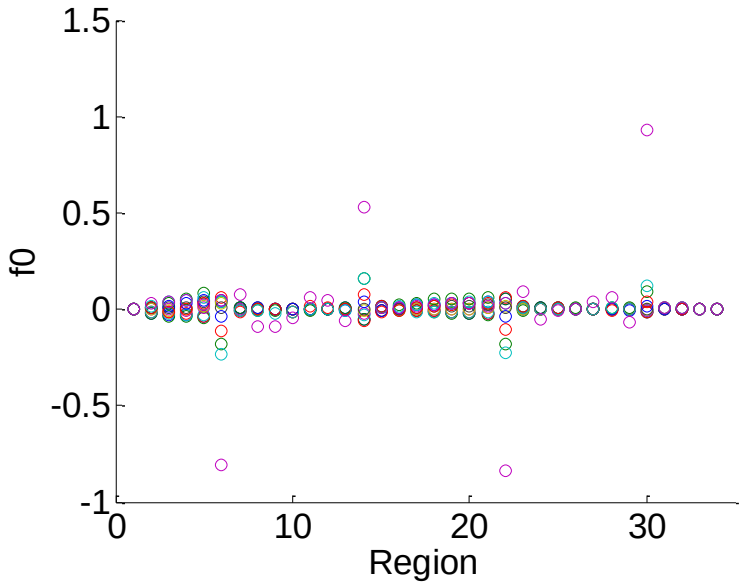

Figure 4.38: Quasidiffusion additive discontinuity factors

For a steady-state problem, the additive discontinuity factor reproduces the reference solution for all four methods. With regards to the actual discontinuity factors, there are both positive and negative values. In addition, the values are centered on zero. This is different from the traditional definition, which typically produces only positive values that are centered on one. For this specific application, all of the discontinuity factors are within a range of -1 to 1.3.

Unfortunately this analysis does not indicate whether this method would be stable when applied to full core power calculations with feedback. However, the range of additive discontinuity factors is small compared to the traditional method. This suggests a potential solution to the stability problem.

Chapter 5
Summary, Conclusions and Future Research

5.1 Conclusions

The reactor physics methods traditionally used for the analysis of Light Water Reactors encounter difficulties when applied to reactors with severe axially heterogeneities, such as the RBWR. In particular, the large axial blankets in the RBWR introduce significant axial neutron streaming, which exposes the limitations of diffusion theory methods based on two-dimensional homogenized group constants. A principal focus of the research here was to investigate the use of methods based on higher spherical harmonics or the Quasidiffusion equations. The objective was to develop a higher-order axial solver that could be coupled with a lower-order radial diffusion solver for practical steady-state and transient analysis of axially heterogeneous reactors such as the RBWR.

In chapter 2, the core analysis methods were described that were originally developed in this research for the RBWR, based on 3-D Monte Carlo cross sections and implemented in the coupled PARCS/PATHS core simulator. This work included the development of an innovative method for treating cross section feedback, which included both instantaneous and history branches for 3-D cross sections. These methods were successfully applied to core depletion, the determination of the RBWR equilibrium cycle, and core transient analysis. An original and significant development of this phase of the research was the introduction of an axial discontinuity factor, which corrected the diffusion solution to reproduce the Monte Carlo reference. However, when this method was applied to full core power problems with feedback, instabilities were introduced, due to large values of the discontinuity factors. A bounding scheme was applied to the discontinuity factors, which increased the stability of the solution but introduced error into the calculation.

In order to eliminate the need for bounding the axial discontinuity factors, four different approximations to the transport equation were investigated and discussed in chapter 4. This included the P1, P2, P3 and Quasidiffusion approximations, which each provided a separate method of approximating the angular component of the transport equation. In addition, discontinuity factors were developed for each approximation using homogeneous surface fluxes derived from finite difference. Numerical results were generated based on an RBWR-type problem, where P3 and Quasidiffusion provided the most accurate solution compared to the Monte Carlo reference case. All four methods reproduced the reference solution with the use of axial discontinuity factors. An assessment of the discontinuity factors found that the largest values occurred in the thermal most group (group 12), and that the magnitude of the discontinuity factors decreased as the angular approximation improved. However, P1 appeared to outperform the various higher-order approximations when the bounding treatment was applied to each method.

Since the discontinuity factor is able to account for many sources of error, a subplane method was introduced to reduce the spatial discretization error. The subplane concept involved using a fine mesh spatial flux distribution based on the coarse mesh group constants that were generated by the Monte Carlo calculation. A set of ten fine meshes were applied to each coarse mesh in the cases analyzed, and when the subplane method was coupled with the discontinuity factor bounding, significant improvements were noticed in the P1, P3 and Quasidiffusion solutions. All three methods produced smaller error compared to the reference solution for tighter bounds on the discontinuity factors. Quasidiffusion produced the lowest average error for this method, even though the range of discontinuity factors was similar for all four methods.

Another approach to minimize numerical stability problems was also investigated, based on the use of an additive discontinuity factor relation instead of the multiplicative version. The intent was to eliminate the large discontinuity factors, which result when dividing by a homogeneous surface flux that is close to zero.

The results presented here suggest the following conclusions:

- For axially heterogeneous cores such as the RBWR, diffusion theory is unable to capture the axial streaming effect. Instead, a higher-order transport correction is required such as P3 or Quasidiffusion for the axial flux solution, which can then be coupled to a lower-order two-dimensional radial method.
- The use of the subplane finite difference method should reduce the possibility of producing homogeneous surface fluxes that are close to zero or negative. This is due to the reduced mesh size which produces a refined spatial flux.
- The generation of Eddington factors is computationally inexpensive using Monte Carlo methods, and since they can be implemented as a discontinuity factor into the diffusion equation, this method would be relatively straightforward to implement in a production code such as PARCS.
- Additive discontinuity factors eliminate the possibility of large discontinuity values. However, their behavior for full core problems requires additional study before their effect on the numerical stability can be determined.
- Throughout the analysis, P3 and Quasidiffusion produced similar results and outperformed the other methods. Either of these approximations, coupled with the subplane method, provided the closest solution to the reference. Due to the similarity to diffusion, Quasidiffusion is the best option to improve the axial solution.

5.2 Future Research and Development

5.2.1 Application to Full Core Analysis

The first recommendation involves extending the analysis presented here to full core PARCS/PATHS calculations. This is important to assess the stability of each of the methods. First, the use of Eddington factors should be introduced into the PARCS code. Within PARCS, the axial component of the TPEN method should be changed from NEM diffusion to finite difference Quasidiffusion. In addition, both GenPMAXS and PARCS would have to be altered to place Eddington factors in the PMAXS files. The capability to use both Eddington factors and

discontinuity factors would have to be included within the PARCS diffusion solvers. For completeness, the radial components of the Eddington factors could be included, which would involve additional changes to the radial component of the TPEN solver.

Implementation of the subplane method should further improve the PARCS/PATHS solution. This requires modifications to the meshing scheme within both codes. The use of the subplane method could be implemented independently using the traditional methodology or in conjunction with the Quasidiffusion method.

Full core analysis is also necessary to determine the impact of using additive discontinuity factors compared to the traditional multiplicative version. It is expected that additive discontinuity factors should improve stability, since there is no longer any restriction on the magnitude and sign of the value. However, additional studies should be performed to conclusively establish their performance for full core, 3-D problems.

5.2.2 3-D Monte Carlo Cross Sections in Serpent 2

The next recommendation for future research involves extending the 3-D cross section generation to the Serpent 2 Monte Carlo code [64]. All of the work presented in this thesis involved the original Serpent code [17]. One of the main advantages to Serpent 2 is the improved parallel computing performance. The original Serpent code possessed parallel capabilities, but varying the number of processors for the same problem could produce different results. This led to all of the cross section calculations being performed in serial, which significantly slowed down the run time. By implementing the new tallies in Serpent 2, the time required to generate cross sections should significantly reduce.

Another advantage to Serpent 2 is the ability to perform multiphysics calculations [65] [66] [67] [68]. This is relevant for determining the axial coolant void distribution for the history and branching scheme. The prior method involved generating 2-D cross sections to be used in the PARCS/PATHS core simulator to determine an appropriate void distribution. By introducing multiphysics capabilities, the axial void distribution can be determined using only the Serpent 2 code reducing the amount of simulations.

5.3 Lessons Learned

In this final section, the major hurdles encountered while performing this research are discussed in order to provide guidance for future work. This section mostly covers various techniques applied to solving the second moment equation for the P3 approximation.

As described in Section 2.1.2, neutron balance is ensured by adjusting the absorption or total cross section using the GenPMAXS code. This involves solving a fixed source problem for the absorption or total cross section and making adjustments to the originally generated value. For this procedure, only the zeroth moment equation is preserved. In order to maintain neutron balance for the second moment equation, a similar scheme to the zeroth moment was derived for the P3 approximation. When solving the fixed source zeroth moment equation, the total cross section was adjusted to preserve the zeroth moment balance. This meant that when preserving the second moment balance, the total cross section could not be modified or else the zeroth moment would no longer be balanced. The only value that could be changed that did not affect the zeroth moment was the second moment scattering cross section. If this quantity was changed in order to preserve the second moment balance based on the generated cross sections and currents, significantly different second moment scattering cross sections were calculated. The difference was a result of the larger uncertainties associated with the cell averaged second moment flux and the third moment surface flux from the Monte Carlo simulation. Generating reliable higher-order moments is difficult for large problems with many energy groups. Performing this type of second moment balance should only be performed with large amounts of computational resources, or for small problems with only a few energy groups.

Another recommendation deals with the use of discontinuity factors for the second moment equation. A similar definition based on the zeroth moment methodology for discontinuity factors was applied to the second moment. However, this complicated the solution since the second moment equation depends on continuity of the second moment flux at an interface. Since the zeroth moment equation preserves continuity of the combination of the zeroth and second moment fluxes, and the second moment equation preserves continuity of the second moment flux, this may lead to inconsistencies in the solution. This is due to introducing a discontinuity to the second moment flux, which is part of the zeroth moment

solution. Instead, it is better to only apply a discontinuity factor to the zeroth moment equation and enforce second and third moment continuity for the second moment equation which was done in Section 3.6.3.

Appendix A

Monte Carlo Methods

A.1 Probability Density Functions and the Inversion Method

The fundamental concept of Monte Carlo for nuclear applications involves simulating the life of a single neutron (or other particle type) from its initial emission until its death by absorption or escaping from the system boundaries. Each interaction and its frequency is randomly sampled, based on interaction laws derived from particle physics. If a large number of particles are simulated, then the calculation provides a detailed analysis of the transport process over the region of interest usually at a large computational expense. Though it can be costly, the beauty of Monte Carlo is its ability to handle complex variations in space and energy without explicitly solving for the flux distribution.

Each reaction is represented by a probability density function, or PDF, *f*(x). This distribution represents the probability that a given event will occur over some range from x to x+dx. These distributions when integrated must provide non-negative finite values. To determine the probability that an event will occur before some certain value, we integrate the PDF to determine a cumulative distribution function, or CDF:

$$F(x) = \int_{-\infty}^{x} f(x')dx'. \tag{A-1}$$

One additional note is that each PDF is assumed to be properly normalized so that the integration of *f*(x) over the range of interest is one. This is important for the random sampling process. Throughout this work, we will refer to ξ as the randomly sampled variable.

The simplest version of sampling a random event is found from the inversion technique. This is performed by sampling the variable x using the CDF. First, the CDF is set equal to the

random variable ξ, which is a uniformly distributed variable. From there, the CDF is inverted such that it becomes a function of ξ which allows one to solve directly for the variable x. This sampled quantity is distributed according to the PDF *f*(x):

$$F(x) = \xi. \tag{A-2}$$

Inverting this function, we obtain:

$$x = F^{-1}(\xi). \tag{A-3}$$

Let us consider an example where we sample the distance a neutron travels in a purely homogeneous system with constant interaction probability. The total cross section Σ_t represents the interaction probability per path length travelled by a neutron. One can express the probability that a neutron will undergo an interaction while moving a distance dx as the following:

$$dP = \Sigma_t dx. \tag{A-4}$$

Now let us define a separate quantity P_0(x), which represents the probability that a neutron has traveled a distance x without having an interaction. If this neutron moves a distance dx from its position at x, then the reduction in probability of having an interaction over this distance is defined as:

$$dP_0 = -P_0(x)dP = -P_0(x)\Sigma_t dx. \tag{A-5}$$

If we solve this differential equation, we get an expression for the probability that a neutron will not interact over a distance x:

$$P_0(x) = e^{-x\Sigma_t}. \tag{A-6}$$

Now we can define an expression where a neutron travels some distance x without colliding before interacting for the first time within the next dx:

$$P_0(x)dP = P_0(x)\Sigma_t dx = \Sigma_t e^{-x\Sigma_t} dx. \quad \text{(A-7)}$$

The PDF that describes the neutron free path length is then defined as:

$$f(x) = \Sigma_t e^{-x\Sigma_t} dx, \quad \text{(A-8)}$$

and the corresponding CDF is:

$$F(x) = \int_0^x \Sigma_t e^{-x'\Sigma_t} dx' = 1 - e^{-x\Sigma_t}. \quad \text{(A-9)}$$

If we use the inversion method, then we equate the CDF to the random variable ξ and solve for x:

$$x = -\frac{1}{\Sigma_t}\ln(1-\xi) = -\frac{1}{\Sigma_t}\ln(\xi'). \quad \text{(A-10)}$$

In Eq. A-10, the quantity $1-\xi$ has been replaced by ξ' which is also a random number. This is done because both $1-\xi$ and ξ' are similarly distributed and $1-\xi$ requires an additional mathematical operation. This provides an expression for the distance a neutron travels in an infinite homogeneous medium through the use of random sampling.

A.2 Surface Tracking

The previous section considered how a neutron behaves in an infinite homogeneous medium. However, in most applications there are several homogeneous regions of interest, or cells. In order to model the behavior of a neutron through these multiple regions, it is necessary to track the distance from the neutron to the surface of the current cell the neutron is in. This is important when the sampled distance a neutron travels is greater than the distance to the cell boundary. To find the distance to the nearest surface, the universe in which the particle is located is first found, based on the minimum and maximum of the x,y, and z coordinate axes. Once the universe is located, the cell in which the particle is located is found using a similar manner by looping over the cells that compose the universe. Next, the distance to each surface

that comprises the cell the particle is located in is determined based on the surface type and the particles position and direction of flight. For a general case, let:

$$S(x, y, z) = 0, \tag{A-11}$$

be the equation of a surface. The distance to this surface is found by solving the following equation for the variable d:

$$\mathrm{S}\big(x + d\Omega_x, y + d\Omega_y, z + d\Omega_z\big) = 0. \tag{A-12}$$

For example, consider a particle positioned at (x,y,z) traveling in direction <u,v,w>. To find the distance (d) to a plane positioned on the x-axis, assuming that the u component is non-zero, is found from:

$$d = -\frac{x}{u}. \tag{A-13}$$

The expression for the distance to a surface changes based on the type of surface. Once all of the distances have been calculated, the minimum distance is returned. This provides the distance to the nearest surface. To account for stopping a neutron at the nearest surface, there are two statistically similar approaches.

To determine the distance to a collision, the optical depth is sampled from the PDF:

$$f(\tau) = e^{-\tau}, \tag{A-14}$$

where

$$\tau = \int_0^d \Sigma_t(x)dx. \tag{A-15}$$

The distance to a collision is found by solving for the variable d using these equations. This method for finding the distance is valid even when the cross section is a continuous function of distance.

Let us consider a problem with two homogeneous cells with different total cross sections. We will call the distance a neutron lies from the edge of its current cell d and the sampled path length in the original cell x_1. If the path length sampled for x_1 is greater than d, then we want to find the remaining path length x_2 in the second cell. We can determine this quantity by preserving the non-interaction probability:

$$e^{-x_2\Sigma_{t2}} = e^{-(x_1-d)\Sigma_{t1}}, \tag{A-16}$$

where the distance the neutron travels in the second region is found to be:

$$x_2 = (x_1 - d)\frac{\Sigma_{t1}}{\Sigma_{t2}}. \tag{A-17}$$

From this expression, we can find the total distance the neutron travels in both regions as the following:

$$x = d + x_2 = d + (x_1 - d)\frac{\Sigma_{t1}}{\Sigma_{t2}}. \tag{A-18}$$

Similarly, we can also determine the distance the neutron travels in the second region by stopping the neutron at the boundary surface. The distance the neutron travels in the second region is then sampled using the same technique used in the first region but by sampling instead over the second region and so forth. This is equivalent, since neutrons are unaware of their past history and sampling from the boundary is the same as sampling from any other starting point. Mathematically, this is true since the distance a neutron travels is exponentially distributed and exponential distributions are “memoryless”. Another way to consider this is based on how the distance to the next collision is found. The sampled distance is based on the probability that the neutron will interact over a given distance. By stopping the neutron at the surface, there are now two probabilities introduced, one based on the probability of interacting in the first region and another based on the probability of interacting in the second region. Therefore the total distance the neutron travels through both regions can

be found based on the separate probabilities of interaction in each region. If multiple surfaces are crossed, the resampling technique can be much simpler. Both these techniques require knowledge of the distance to the surface. The next section discusses a method that calculates the distance to the next neutron interaction without knowledge of the distance to the surface of the current cell.

A.3 Delta Tracking

The delta tracking method is a technique that samples collision points without handling surface crossing and was first introduced by Woodcock [69]. If an interaction is considered where the neutron is not absorbed and its energy and direction of flight remain the same, then this will be considered a virtual collision with cross section Σ_0. This type of collision has no effect on the statistics or outcome of a Monte Carlo simulation. If a virtual collision cross section is added to each region such that the sum of the total cross section and virtual collision cross section are the same for each cell, then this sum can be sampled repeatedly over the entire geometry without tracking surfaces. The sum of these cross sections is often referred to as the majorant cross section (Σ_m) and is equal to the maximum of all total cross sections in the system:

$$\Sigma_m = \Sigma_0 + \Sigma_t. \tag{A-19}$$

The virtual collision cross section from each region can be calculated by subtracting the region's total cross section from the majorant cross section. To sample a particles interaction, the path length of the neutron is sampled using the majorant cross section. The neutron is moved to this location and its type of interaction is sampled, real or virtual. Whether a collision is real or virtual is determined based on the probability of the neutron having a virtual collision. This is found from the ratio of the virtual collision cross section to the majorant cross section:

$$P = \frac{\Sigma_0}{\Sigma_m} = \frac{\Sigma_m - \Sigma_t}{\Sigma_m} = 1 - \frac{\Sigma_t}{\Sigma_m}. \tag{A-20}$$

If the sampled collision is found to be virtual, then the path length is sampled again and the process is repeated until a real collision occurs or the neutron exits the system. The

advantage of this technique is that it simplifies the geometry while performing the simulation and can decrease simulation times when the total cross sections for all regions are similar in magnitude. One of the downsides of delta tracking is that there is no longer any surface tracking, so estimates of the flux must be found through the use of a collision estimator instead of track lengths. This also requires the ability to quickly locate the region the particle is interacting in. For reactor problems that take advantage of repeated lattices, this is simple. However, for more general applications this can cause the simulation to slow down as the code searches for the particles location. Additionally, if there is a large absorber, such as a control rod in the system, then the majorant cross section can become quite large and the number of virtual collisions to real collisions will increase. This can impact the simulation speed due to the increased number of sampled virtual reactions. Within the Serpent code, this is treated by switching between delta- and surface-tracking based on a specified threshold c:

$$\frac{\Sigma_{tot,m}(E)}{\Sigma_{maj}(E)} > 1 - c. \tag{A-21}$$

The constant c can be varied between 0 (no delta-tracking) and 1 (full delta-tracking). When a particle interacts, the ratio of the total cross section to the majorant is compared using Eq. A-21. If the relation in Eq. A-21 is true, delta-tracking is used until the next collision. Otherwise, the code switches to the surface tracking routine. By default, Serpent uses a threshold of $c = 0.9$ [70].

A.4 Neutron Interactions

When an interaction occurs, sampling the type of interaction is a straightforward process. The first step involves sampling which isotope within the given region the neutron is interacting with. Isotopes will be distinguished by using the letter m. All dependencies on space and energy are ignored in this section for the sake of simplicity:

$$P_m = \frac{\Sigma_{t,m}}{\Sigma_t}. \tag{A-22}$$

This is followed by determining the type of interaction the neutron has with isotope m. Usually, the type of interactions are lumped into three categories: capture, fission or scattering. The interaction type will be denoted by using the letter i:

$$P_i = \frac{\Sigma_{i,m}}{\Sigma_{t,m}}. \tag{A-23}$$

Using both of these definitions, the probability that a neutron undergoes interaction i in material m is found from the following relation:

$$P_{i,m} = \frac{\Sigma_{i,m}}{\Sigma_t}. \tag{A-24}$$

When a neutron interaction is found, a separate sample is performed to determine the type of interaction using the above relation. If the neutron interaction is absorption or fission, the neutron history in the current cycle is stopped. For fission, the number of emitted neutrons and their energies are sampled based on the target nucleus and the energy of the incident neutron using the information from the ACE data files for $\bar{\nu}$ and the appropriate spectrum (Maxwell, Watt or evaporation spectra). The direction of flight is sampled isotropically, and the produced fission neutrons are used as new source neutrons for the next cycle. For k-eigenvalue calculations, the number of neutrons sampled is modified by a factor of $1/k$ where k is the eigenvalue estimate from the previous cycle.

Scattering reactions are much more complicated in Monte Carlo. Unlike absorption and fission, the neutron history is not terminated, and a new energy and direction of flight must be sampled for the incident neutron. This involves solving the kinetic collision equations for the LAB and center-of-mass (CM) frames. A detailed discussion for converting between the LAB and CM frames is discussed in several of the references [7] [17]. The discussion here will focus on the steps for sampling an elastic scattering collision. This involves three steps: 1) sampling the initial target velocity in the LAB-frame (free-gas treatment), 2) sampling the scattering angles in the CM-frame, and 3) the azimuthal rotation of the direction vectors. The first step is necessary to calculate the target velocity to convert to the CM-frame. Once in the CM-frame, the outgoing

velocities for both the neutron and target nuclide are determined by sampling the polar and azimuthal angles. The polar angle is sampled based on the reaction type and can be either isotropic or anisotropic, whereas the azimuthal angle is always isotropically sampled in the CM-frame. Using the calculated polar and azimuthal angles, the velocities are then converted back into the LAB-frame and the next neutron collision is sampled based on a new position, angle and energy.

A.5 Criticality Source Calculations

The k-eigenvalue problem is typically solved using the power iteration method. For criticality source calculations, the simulation is broken up into cycles or neutron generations. Each cycle follows a set number of neutron histories and the results between each cycle are treated independently. The initial fission source is either specified by the user or sampled randomly throughout the geometry. Fission neutrons that are produced in one cycle are used as the fission source for the next. Since the initial fission source can often be quite different from the true distribution, a number of cycles must be discarded (inactive cycles) at the beginning to establish an appropriate fission source distribution. This prevents the calculated results from being biased by a poor initial guess.

Typically, the number of source neutrons produced at the end of one cycle does not match the number produced at the beginning of the cycle. The ratio of these two quantities defines the criticality eigenvalue:

$$k_i = \frac{Number\ of\ generated\ source\ neutrons\ for\ generation\ i+1}{Number\ of\ neutron\ histories\ for\ generation\ i}, \tag{A-25}$$

where k_i is the criticality eigenvalue for cycle i. However, this estimate is not typically used in production codes and instead the criticality eigenvalue is found from a combination of fission neutron production estimators. The final eigenvalue of the simulation is an average based upon all active cycles (cycles simulated after the inactive cycles) calculation of the criticality eigenvalue. In Serpent, the number of neutron histories at the beginning of each cycle is a fixed number based on user input, therefore the number of neutrons based on the source

distribution must be artificially raised or decreased to match this. This is done through the k-eigenvalue method.

The artificial increase and decrease of neutron histories from cycle-to-cycle is similar to the process of dividing the fission term of the transport equation by the criticality eigenvalue. When the generated fission source is larger than the number of histories ($k_i > 1$), the fission source distribution for the next cycle is randomly selected from the generated fission source. If the generated fission source is smaller than the number of histories ($k_i < 1$), then several of the source neutrons are randomly duplicated to meet the number of neutron histories.

A.6 Scoring Results in Monte Carlo

The above sections have discussed how the neutron histories are treated during a Monte Carlo simulation. However, the purpose of these calculations is to determine various estimates based on the neutron interactions. This process involves *scoring* various quantities of interest, usually cross section and flux estimates, to assess the behavior of neutrons within the system. The purpose of this section is to outline the process for: estimating the neutron flux based on a collision estimator, and calculation of reaction rates and detector tallies.

A.6.1 Collision Estimator

Due to the use of delta-tracking in the Serpent Monte Carlo code, the neutron flux is estimated based on a collision estimator. Other Monte Carlo codes such as MCNP use the track-length estimator, which involves calculating the neutron path length traveled within a given cell to determine the flux distribution [71]. Within this work, however, the flux and reaction rate estimates for the simulations were based on the collision estimator. The concept is similar to the track-length estimator method but takes advantage of the fact that the total interaction rate can be determined by counting the number of collisions. This is done by *scoring*, recording information based on simulated events, the weighted inverse of the total cross section in which the interaction takes place:

$$\phi^i = \frac{W^i}{\Sigma_t^i}, \tag{A-26}$$

where W^i is the neutron weight, Σ_t^i is the total cross section of the material where the interaction is taking place based on the neutrons incoming energy and ϕ^i is the average path length covered by the neutrons. This method effectively involves summing the neutron mean free paths within the region of interest. The calculated quantity in Eq. A-26 continues to increase as the number of simulated particles interact within the given region. Therefore, the flux distribution can be normalized within Serpent based upon one of three user-defined methods: 1) fission rate, 2) neutron emission rate, or 3) total power.

It is important to note that this method works relatively well except for geometries that are optically thin. Since this method requires interactions to take place within the region of interest, if the region is small such that very few interactions takes place, this reduces the number of interactions and increases the relative uncertainty associated with the collision flux estimator. This problem does not exist with the surface tracking method because the total track length is calculated for every region that the neutron passes through, not just the regions it interacts in. However, the collision estimator is suitable for delta-tracking, since it does not require the neutron to track the nearest surfaces. This can reduce the simulation time if the geometry of interest is extremely complicated and consists of many surfaces or inhomogeneous material regions.

The collision flux estimator can also be split up to tally the flux distribution for various user-specified energy bins. This involves storing the quantity in Eq. A-26 based on the neutron energy within a certain range. Therefore flux estimates can be produced over several spatial and energy ranges.

A.6.2 Reaction Rates

In addition to the neutron flux, reactions rates are useful for determining the type of behavior occurring within the system. One simple method involves scoring each of the individual physical interactions (fission, scattering, absorption, etc.) when they take place. However, that method is not very efficient and can produce larger relative uncertainties for reaction types that do not take place very often in certain regions at specific energies. Instead, an implicit estimator is used with Serpent that scores the various reaction rates each time a

collision takes place. This is an extension of the collision estimator that calculates the reaction rate based on:

$$R = \sum_{i=1}^{I} f^i \phi^i, \tag{A-27}$$

where R is the reaction rate, ϕ^i is the collision estimate of the flux for neutron i of total neutrons I, and f^i is the response function based on the material of interest and incident neutron energy. To find the total reaction rate, f^i is replaced by the macroscopic total cross section.

One advantage to this method is that the reaction rates for various interactions (scattering, fission, absorption, etc.) can all be tallied independent of the actual physical interaction taking place. In essence, this reaction rate tallies the probability that the incident neutron will have a certain interaction. This method increases the total number of scores for each reaction type, which reduces the statistical uncertainty.

Average group-wise macroscopic cross sections can be calculated based on the reaction rate in Eq. A-27 and the flux estimator in Eq. A-26 by taking the ratio of the reaction rate to the flux. This does not require any normalization, since whatever normalization method the user decides should cancel out, due to the presence of the flux in both the numerator and denominator. The macroscopic cross sections are calculated based on the reaction rates and flux for each cycle independently. The final cross section is produced by averaging over all cycles. Therefore it is important to include enough neutron histories per cycle so that the cycle macroscopic cross section is an accurate representation of the true value. In addition, it is important to include a sufficient number of cycles so that the average macroscopic cross section is based on a large number of samples. By performing this process, the central limit theorem states that as the number of simulated particles (histories and cycles) increases, the simulated quantity will approach the true mean and the sample mean will be randomly distributed about the true mean with a normal distribution.

The statistical uncertainty of a calculated quantity is based on its standard deviation of the mean. For the central limit theorem this is:

$$\sigma_{\bar{X}} = \sqrt{\frac{\sigma^2}{n}}, \tag{A-28}$$

where $\sigma_{\bar{X}}$ is the standard deviation of mean $\bar{X}$, n is the number of samples and σ^2 is the variance of the true mean μ. From this expression, the standard deviation of the mean reduces as a function of the inverse square root of the number of samples. As n increases, the standard deviation decreases at a rate of $\sqrt{1/n}$. This rate of convergence is based on the Berry-Esseen theorem [72]. This is important because Monte Carlo results are based on the central limit theorem; therefore, if one wishes to reduce the statistical uncertainty on a given parameter, they must increase the number of sampled particles by a factor of four to reduce the uncertainty by a factor of two.

A.6.3 Surface Tallies

The topic of Monte Carlo tallies, specifically additional tallies implemented into the Serpent code, was touched upon in Section 2.1.1.4. The goal of this section is to provide a general overview of how tallies (detectors in Serpent) are performed. In general Monte Carlo tallies are calculated similarly to the method outlined in Section A.6.2. However, in that section, only tallies based on volume quantities were discussed. Tallies based on surface quantities are useful to determine the neutron current and surface flux.

The neutron current is calculated by scoring the number of particles that cross a surface. This is simple to tally since it only requires the neutron's energy to score the crossing in the appropriate energy bin. For the surface flux, the neutrons incoming angle with respect to the surface normal is required:

$$\phi_s^i = \frac{W^i}{|\mu^i|}, \tag{A-29}$$

where ϕ_s^i is the surface flux based on neutron i with weight W^i and μ^i is the cosine of the angle between the neutron direction and the surface normal. As μ^i approaches zero, the surface flux estimate of neutron i approaches infinity. This small scattering angle can lead to large uncertainties on the surface flux.

Instead, like most Monte Carlo codes, a grazing angle cut-off was implemented [73] [74]. For any angle μ<0.1, the zeroth moment is assumed to be 20 based on the following expression:

$$\frac{1}{|\mu|} = \frac{2}{\epsilon} = \frac{2}{0.1} = 20. \tag{A-30}$$

This representation of the cutoff is based on the same implementation used in MCNP [71].

A.7 Monte Carlo Depletion

Burnup in Serpent is carried out internally which reduces the complexity of coupling the code to an independent depletion solver [75]. This involves a two-step approach where the neutron-induced transmutation reactions are determined from the transport simulation which is then combined with the radioactive decay constants and fission product yields from nuclear data files. All of this information is combined to solve the Bateman equations which represent the time-dependent change in nuclide concentrations:

$$\frac{dN_j}{dt} = \sum_{i \neq j} \lambda_{ij} N_i - \lambda_j N_j, \;\; N_j(0) = N_0, \;\; j = 1, \dots, n, \tag{A-31}$$

where N_j is the concentration of nuclide j, λ_{ij} is the transmutation coefficient for isotope i to isotope j and n is the number of tracked nuclides. The $\lambda_{ij} N_i$ term represents the production of isotope N_j from other isotopes while the $\lambda_j N_j$ is the removal of isotope N_j over time.

The transmutation coefficient depends on the one-group flux, the fission yield for transmutation from isotope i to isotope j and the microscopic one-group total fission cross section:

$$\lambda_{ij} = \gamma_{ij} \phi \sigma_{f,i}. \tag{A-32}$$

The flux spectrum is calculated based on the normalization of a fine energy grid of usually over 100,000 points and the one-group microscopic total fission cross section is found using flux weighted integrals as described in Section A.6.2. The fission yields are linearly interpolated from tables based on the flux spectrum. Fine-group microscopic total fission cross sections and yields are found from ENDF data files.

The solution to the Bateman equations involves solving a series of coupled first-order differential equations. Typically this is solved by placing the expressions in matrix notation and solving for the nuclide concentrations based on the transmutation data found from the transport calculation and data tables. Serpent uses the CRAM to solve this system of equations [40]. Once the nuclide concentration is found using the Bateman equations, the materials within Serpent are updated and the next transport sweep is performed. Because the transmutation coefficients are assumed constant over the depletion step, it is preferred that the time step be relatively small so this approximation holds. However, in typical applications, this approach is insufficient since depletion can be carried out over months to years, which would require a large amount of depletion steps and thus slow down the calculation. Instead, a predictor-corrector method is utilized to correct the transmutation coefficients. This involves performing an initial transport and depletion solve (predictor) followed by a second transport solve with the “predicted” material compositions. The transmutation coefficients are then found from the average of the two transport calculations to perform a final depletion calculation to produce the “corrected” isotopic concentrations for the next depletion step.

Appendix B

Fuel vs. Non-Fuel Eddington Factors

B.1 Pin Cell Analysis

B.1.1 Pin Cell Geometry and Discretization

To illustrate the physical characteristics of Eddington factors, a simple one-dimensional pin cell composed of the various RBWR axial layers was created using Serpent. Figure B.1 depicts the axial and radial geometry and compositions. In the left figure, the outside varying colors represent the 34 different coolant densities used for each of the 34 different axial meshes. The 5 inside regions are represented by the blankets (green) and fissile (blue) zones. In the right figure is a top-down view of the pin cell. The axial discretization is described in Table B.1 and the fuel compositions can be found in Table B.2.

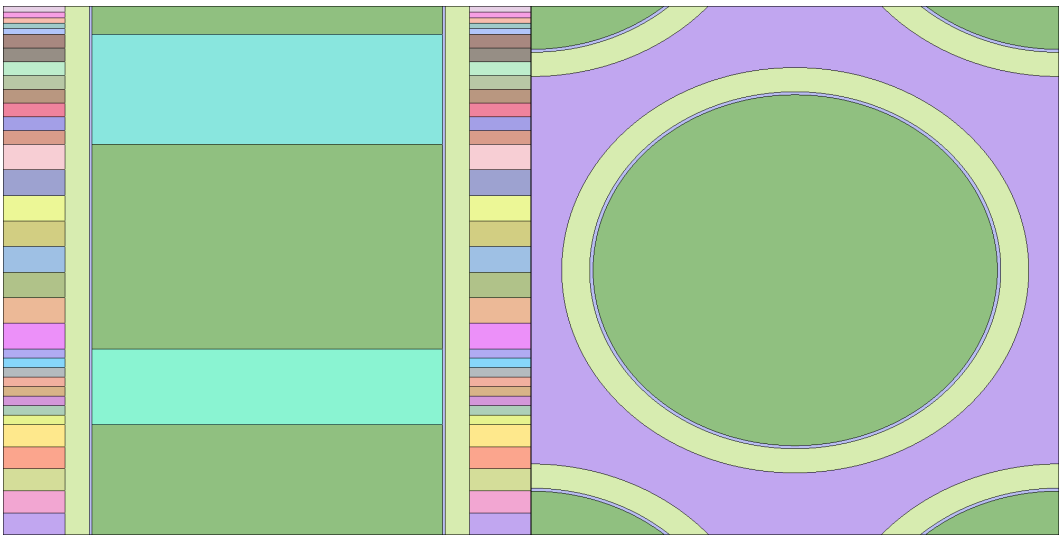

Figure B.1: RBWR pin cell model with axial view (left) and radial view (right)

Table B.1: Axial discretization

Axial Layer	Total Length (cm)	Number of Meshes	Mesh Length (cm)
Lower Blanket	28	5	5.6
Lower Fissile	19.3	8	2.4125
Internal Blanket	52	8	6.5
Upper Fissile	28	8	3.5
Upper Blanket	7	5	1.4
Total	134.3	34	-

Table B.2: Fuel compositions

Isotope	Blanket Number Density	Lower Fissile Number Density	Upper Fissile Number Density
U-235	4.738E-05	2.81000E-05	3.22857E-05
U-238	2.335E-02	1.38333E-02	1.59000E-02
O-16	4.679E-02	4.71667E-02	4.70000E-02
Np-237	-	4.30000E-05	3.34571E-05
Pu-238	-	2.83667E-04	2.20143E-04
Pu-239	-	4.23333E-03	3.29714E-03
Pu-240	-	3.52000E-03	2.74429E-03
Pu-241	-	4.94333E-04	3.84000E-04
Pu-242	-	4.85667E-04	3.77143E-04
Am-241	-	3.51000E-04	2.72571E-04
Am-242m	-	1.47333E-05	1.14571E-05
Am-243	-	1.27667E-04	9.96429E-05
Cm-243	-	2.22333E-06	1.72714E-06
Cm-244	-	1.02267E-04	7.93714E-05
Cm-245	-	3.19000E-05	2.47857E-05
Cm-246	-	9.86333E-06	7.69000E-06
Cm-247	-	1.74333E-06	1.35286E-06
Cm-248	-	1.23000E-06	9.57571E-07

The goal of this analysis was to illustrate the change in Eddington factors based on axial position as well as position in fuel and non-fuel regions. The pin cell model contained the same axial meshing as the RBWR assembly, as well as the same twelve group energy structure.

B.1.2 Pin Cell Eddington Factors

Tallies were placed within the fuel regions and all materials outside the fuel including the gap, cladding and coolant. For simplicity, only the fastest and most thermal groups (group 1 and group 12) are shown in this section. The variation in Eddington factor for the fast group in fuel and non-fuel regions is shown in Figure B.2 and Figure B.3.

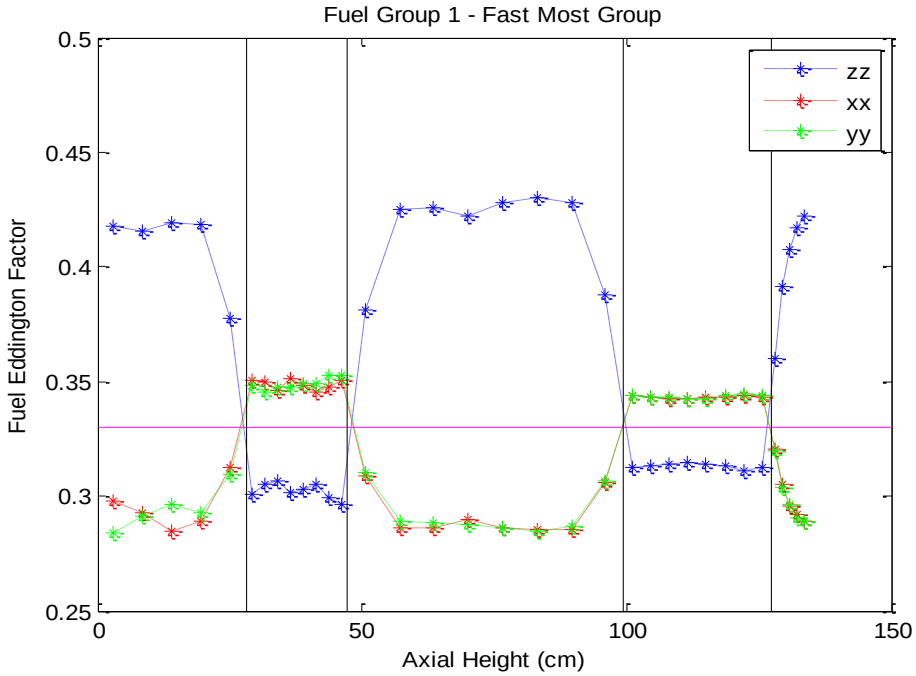

Figure B.2: Fast group Eddington factors in fuel regions as a function of axial height

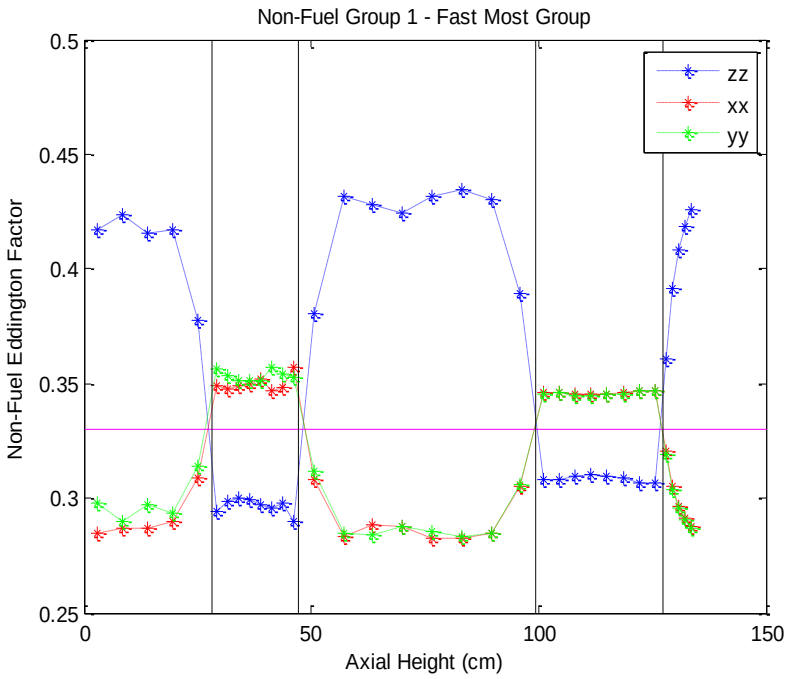

Figure B.3: Fast group Eddington factors in non-fuel regions as a function of axial height

There is little variation seen between the fuel and non-fuel regions for the fast group, and any fluctuations are likely due to statistical noise. This is because the high energy neutrons have a large mean free path and easily stream in and out of fuel and non-fuel regions. The xx and yy, or radial, Eddington factors are effectively the same, except for statistical noise due to the symmetry of the problem. It can also be noted that the radial components are smaller than one-third when the axial component is greater than one-third. And the same is true for the opposite scenario. This is because if an Eddington factor is greater than one-third, then the neutron favors traveling in that direction. For instance, between 50 cm and 100 cm neutrons prefer to travel axially as opposed to radially. Another observation is that fast neutrons tend to favor axial streaming in blanket regions and radial streaming in fissile regions. The reason for this is that a majority of the neutrons at high energies are born in fissile regions isotropically. This explains why the Eddington factors are closer to one-third for both the radial and axial components in fissile zones.

The same analysis was performed for the most-thermal group. Figure B.4 and Figure B.5 show the variation in Eddington factors as a function of height in both the fuel and non-fuel regions.

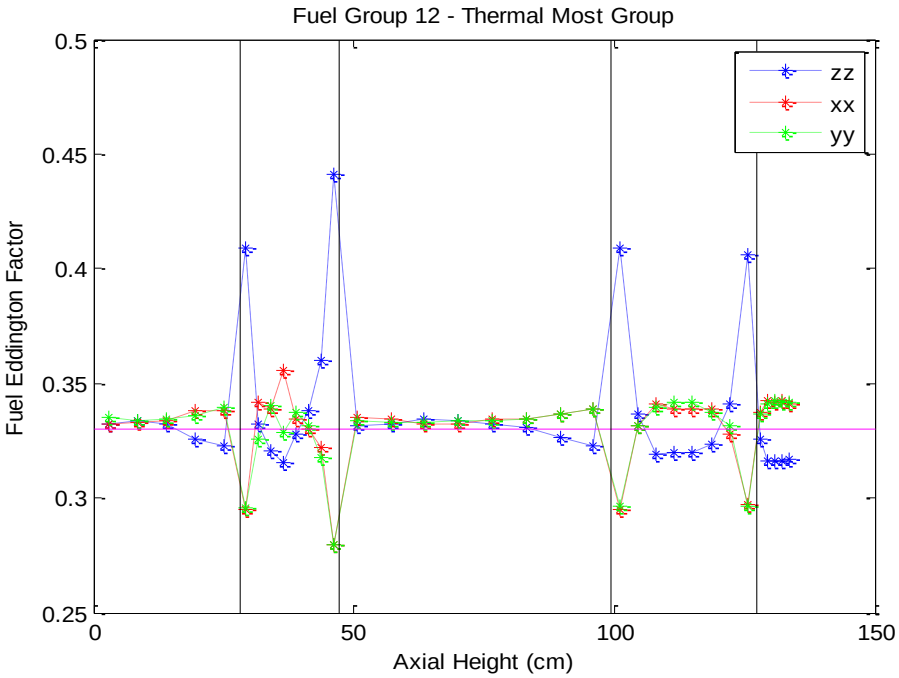

Figure B.4: Thermal group Eddington factors in fuel regions as a function of axial height

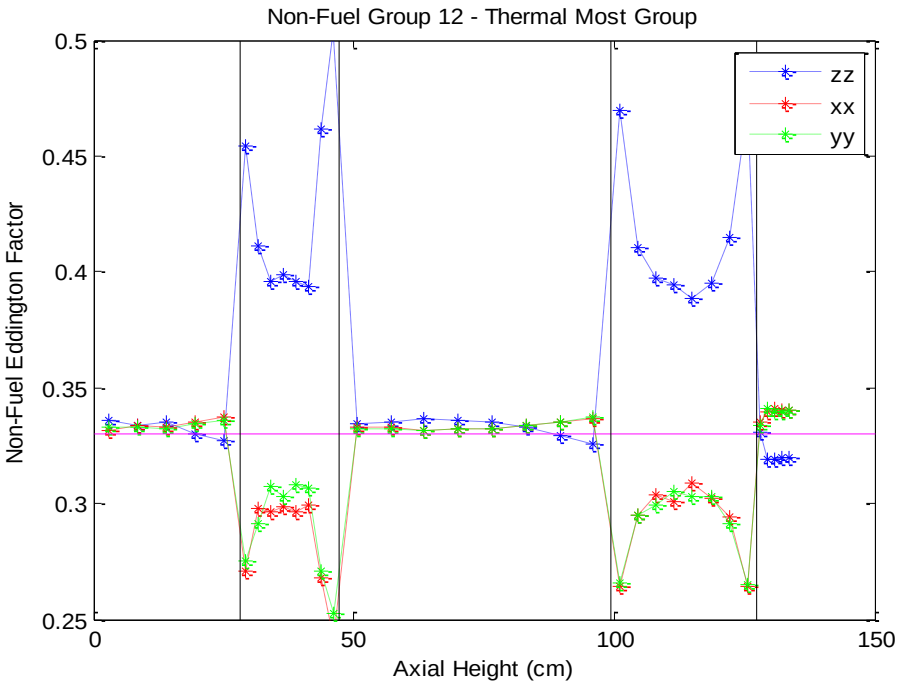

Figure B.5: Thermal group Eddington factors in non-fuel regions as a function of axial height

In this case there is a significant different between the fuel and non-fuel regions for the thermal group. Within the non-fuel regions adjacent to the fissile zones, there is a large preference for neutrons to stream in the axial direction. However, this is only true on the boundary of the fissile zones in the fuel regions. For both the fuel and non-fuel regions, streaming tends to be isotropic in the blanket regions except for the upper blanket, where radial streaming is slightly favored. The reason for this change in Eddington factors is due to the reduced mean free path of the neutrons, especially in the fissile regions. In the non-fuel regions, axial streaming is favored since most thermal neutrons that enter the fissile fuel will be absorbed eliminating the neutrons streaming in the radial direction. There is a peak in the Eddington factors on the edge of the fissile zones from the contribution of thermal neutrons leaking in from the blanket regions. This peaking also appears within the fuel regions due to the lower absorption cross section of the blanket zones. One other interesting note about the fuel region in the fissile zone is that the neutron streaming preference changes based on the axial height. On the edges of the fissile zone, neutrons prefer to stream axially. However, the interior

of the fissile zones prefer to stream radially. This can be explained by self-shielding of the fissile fuel. Neutrons closer to the interior of the fissile zone have a shorter distance to travel radially to exit the fuel and thus prefer radial streaming. While neutrons near the fissile-blanket interface have a shorter distance to travel axially to exit the fissile fuel and thus prefer axial streaming. A comparison of the fuel vs. non-fuel regions for the fast and thermal groups is shown in Figure B.6 and Figure B.7.

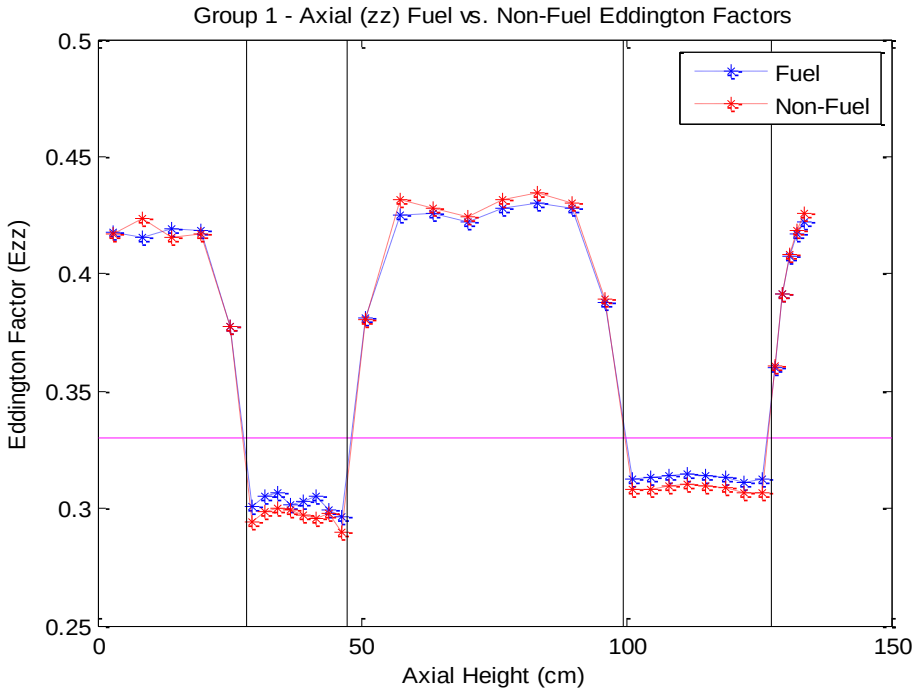

Figure B.6: Fuel vs. non-fuel Eddington factors for fast neutrons

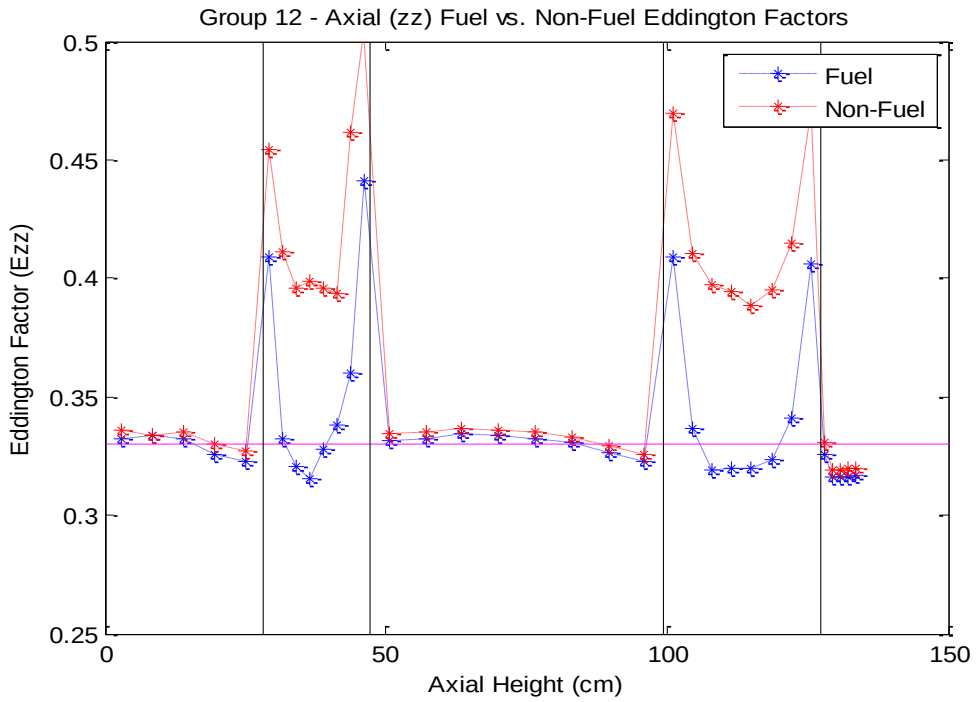

Figure B.7: Fuel vs. non-fuel Eddington factors for thermal neutrons

These figures directly compare the differences in streaming for fuel and non-fuel regions based on energy group. In the next section, the study will be extended to an RBWR-type assembly to see which sections have the largest impact on streaming within the core.

B.2 RBWR Assembly Analysis

B.2.1 RBWR Assembly Geometry and Discretization

An assembly model based on the RBWR assembly was generated within Serpent. Axial and radial images of the assembly are shown in Figure B.8. The same axial discretization as outlined in Table B.1 was used as well as the fuel composition in Table B.2. Reflective boundary conditions were applied in both the axial and radial directions. No reflectors were used in this model to simplify the geometry and improve statistics within the fuel regions.

Figure B.8: RBWR-type assembly axial view (left) and radial view (right)

B.2.2 RBWR Assembly Eddington Factors

A similar analysis to the pin cell was performed to generate Eddington factors for the various axial layers of the RBWR assembly. This time, all materials were tallied together instead of tallying the fuel and non-fuel regions separately. When generating cross sections, this is what is typically employed during the homogenization process. The Eddington factors for the fast and thermal groups as a function of axial height are shown in Figure B.9 and Figure B.10.

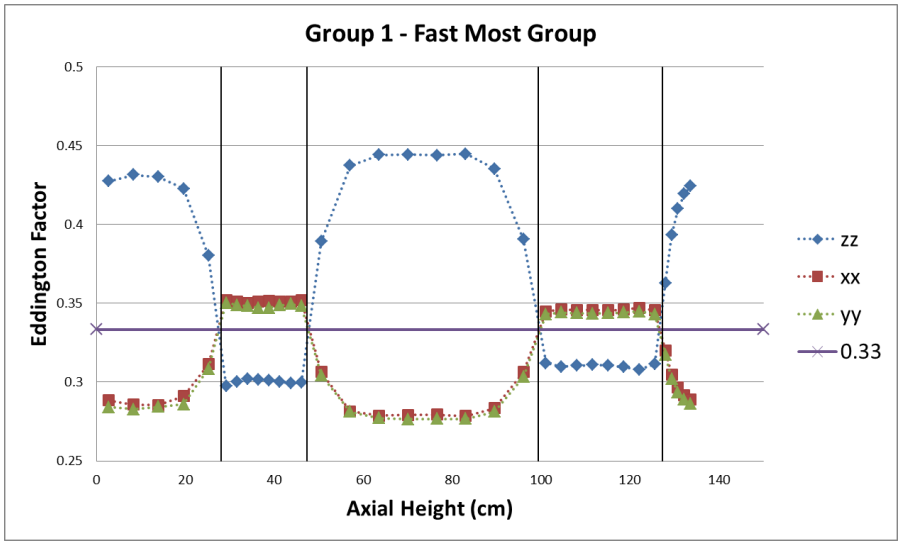

Figure B.9: RBWR assembly Eddington factors for fast group neutrons

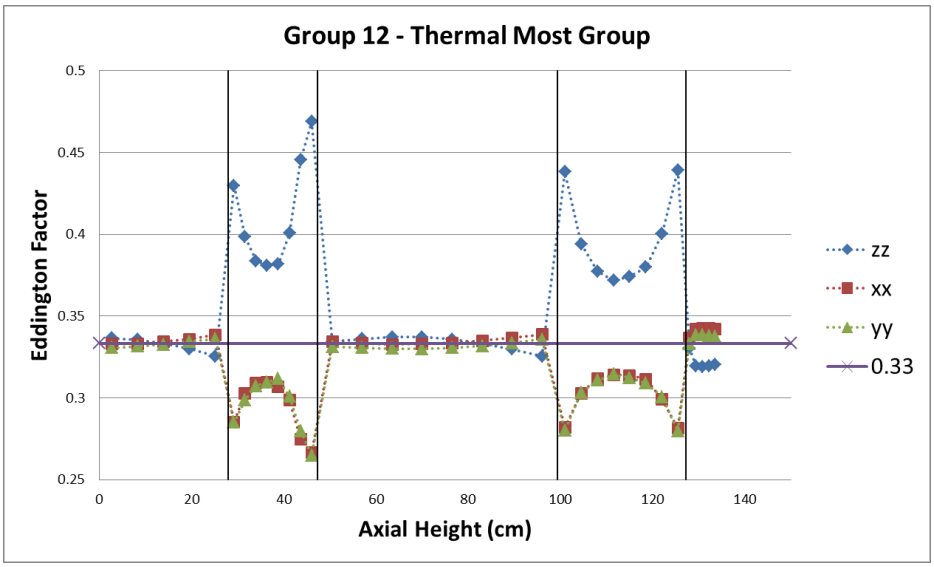

Figure B.10: RBWR assembly Eddington factors for thermal group neutrons

The fast group Eddington factors for the assembly are the same as the results seen in the pin cell analysis for both the fuel and non-fuel regions. For the assembly thermal group, the Eddington factors are similar to the non-fuel Eddington factors seen in the pin cell calculation. This indicates that a majority of the neutron streaming is taking place in the non-fuel regions for the thermal group. The reason for this is the small mean free path in the fuel regions (~1-2 cm) at thermal energies.

Appendix C

Scattering Cross Section in Pn

C.1 Scattering Cross Section Derivation

The differential scattering cross section can be written as an expansion of Legendre polynomials:

$$\Sigma_s(\vec{\Omega} \cdot \vec{\Omega}') = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \Sigma_{s,n} P_n(\vec{\Omega} \cdot \vec{\Omega}'), \tag{C-1}$$

where the $P_n(\vec{\Omega} \cdot \vec{\Omega}')$ term can be expanded using the Addition Theorem:

$$P_n(\vec{\Omega} \cdot \vec{\Omega}') = \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^m(\vec{\Omega}) Y_n^{m*}(\vec{\Omega}'). \tag{C-2}$$

In Eq. C-2 we have introduced what are commonly referred to as the Spherical Harmonics Functions:

$$Y_n^m(\vec{\Omega}) = a_{n,m} P_n^{|m|}(\mu) e^{im\omega}, \tag{C-3}$$

where μ is the polar angle, ω is the azimuthal angle, and the terms $a_{n,m}$ and the Associated Legendre Functions ($P_n^{|m|}$) are defined by:

$$a_{n,m} = (-1)^{(m+|m|)/2} \left[\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!} \right]^{\frac{1}{2}}, \tag{C-4}$$

$$P_n^m(\mu) = (1-\mu^2)^{m/2}\left(\frac{d}{d\mu}\right)^m P_n(\mu). \tag{C-5}$$

In 1-D, the angular flux is just a function of the polar angle and the spatial variable x:

$$\psi\left(\vec{r},\vec{\Omega}\right) = \psi(x,\mu). \tag{C-6}$$

This definition of the angular flux can then be substituted into the scattering integral:

$$\begin{aligned}\int_{4\pi}\Sigma_s\left(\vec{\Omega}'\cdot\vec{\Omega}\right)\psi\left(\vec{r},\vec{\Omega}'\right)d\vec{\Omega}' &= \int_0^{2\pi}\int_{-1}^{1}\Sigma_s\left(\vec{\Omega}'\cdot\vec{\Omega}\right)\psi(x,\mu')d\mu' d\omega' \\ &= \int_{-1}^{1}\left[\int_0^{2\pi}\Sigma_s\left(\vec{\Omega}'\cdot\vec{\Omega}\right)d\omega'\right]\psi(x,\mu')\,d\mu'.\end{aligned} \tag{C-7}$$

Next, we can use Eqs. C-1 through C-3 and substitute within the inner integral of Eq. C-7:

$$\begin{aligned}\int_0^{2\pi}\Sigma_s\left(\vec{\Omega}'\cdot\vec{\Omega}\right)d\omega' &= \sum_{n=0}^{\infty}\frac{2n+1}{4\pi}\Sigma_{s,n}\int_0^{2\pi}P_n\left(\vec{\Omega}'\cdot\vec{\Omega}\right)d\omega' \\ &= \sum_{n=0}^{\infty}\frac{2n+1}{4\pi}\Sigma_{s,n}\int_0^{2\pi}\frac{4\pi}{2n+1}\sum_{m=-n}^{n}Y_n^m\left(\vec{\Omega}\right)Y_n^{m*}\left(\vec{\Omega}'\right)d\omega' \\ &= \sum_{n=0}^{\infty}\Sigma_{s,n}\sum_{m=-n}^{n}\int_0^{2\pi}\left[a_{n,m}P_n^{|m|}(\mu)e^{im\omega}\right]\left[a_{n,m}P_n^{|m|}(\mu')e^{-im\omega'}\right]d\omega' \\ &= \sum_{n=0}^{\infty}\Sigma_{s,n}\sum_{m=-n}^{n}a_{n,m}^2P_n^{|m|}(\mu)P_n^{|m|}(\mu')e^{im\omega}\int_0^{2\pi}e^{im\omega'}d\omega'.\end{aligned} \tag{C-8}$$

The integral in Eq. C-8 is non-zero only when $m = 0$, therefore we evaluate Eq. C-8 at $m = 0$:

$$\int_0^{2\pi}\Sigma_s\left(\vec{\Omega}'\cdot\vec{\Omega}\right)d\omega' = \sum_{n=0}^{\infty}\Sigma_{s,n}a_{n,0}^2P_n^0(\mu)P_n^0(\mu')2\pi. \tag{C-9}$$

Equation C-9 can be further simplified by noting that the Associated Legendre Functions are equal to the Legendre Polynomials when $m = 0$, and $a_{n,0}^2$ is evaluated at $m = 0$:

$$\Sigma_s(x,\mu,\mu') = \int_0^{2\pi} \Sigma_s\left(\vec{\Omega}' \cdot \vec{\Omega}\right) d\omega' = \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{s,n} P_n(\mu) P_n(\mu'). \quad \text{(C-10)}$$

This is the final form of the scattering cross section used in the Pn derivation and is a function of two separate Legendre polynomials based on μ and μ'.

Bibliography

[1] A. Hébert, Applied Reactor Physics, Québec: Presses internationales Polytechnique, 2009.

[2] Studsvik, "HELIOS Methods. HELIOS User's Manual," Studsvik Scandpower, Inc., 2008.

[3] J. R. Askew, "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries," U.K. Atomic Energy Authority, 1972.

[4] M. J. Halsall, "CACTUS, A Characteristics Solutions to the Neutron Transport Equations in Complicated Geometries," U.K. Atomic Energy Authority, 1980.

[5] M. L. Adams and E. W. Larsen, "Fast Iterative Methods for Discrete Ordinates Particle Transport Calculations," *Progress in Nuclear Energy,* vol. 40, pp. 3-159, 2002.

[6] R. Takeda, J. Miwa and K. Moriya, "BWRs for long-term energy supply and for fissioning almost all transuranium," in *Proceedings of the 2007 Advanced Nuclear Fuel Cycles and Systems (Global) Conference*, 2007.

[7] J. J. Duderstadt and L. J. Hamilton, Nuclear Reactor Analysis, Ann Arbor: John Wiley & Sons, Inc., 1976.

[8] Studsvik, "CASMO-4E: Extended Capability CASMO-4 User's Manual," Studsvik Scandpower, Inc., 2009.

[9] N. Z. Cho, "Fundamentals and Recent Developments of Reactor Physics Methods," *Nuclear Engineering and Technology,* vol. 37, no. 1, 2005.

[10] B. A. Worley and A. F. Henry, "Spatial Homogenization of Diffusion Theory Parameters," Massachusetts Institute of Technology, 1977.

[11] A. F. Henry, B. A. Worley and A. A. Morshed, "Spatial Homogenization of Diffusion Theory Parameters," in *Proceedings of a Specialists' Meeting on Homogenization Methods in*

Reactor Physics, Lugano, 1978.

[12] K. S. Smith, "Spatial Homogenization Methods for Light Water Reactor Analysis," Cambridge, 1980.

[13] A. F. Bielajew, Fundamentals of the Monte Carlo method for neutral and charged particle transport, Ann Arbor: University of Michigan, 2001.

[14] B. Petrovic, "Monte Carlo Methods for Efficient Reactor Analysis," 6 April 2009. [Online]. Available: http://web.ornl.gov/sci/nsed/outreach/presentation/2009/Petrovic_seminar.pdf. [Accessed 2015].

[15] W. Navidi, Statistics for Engineers and Scientists, 2nd ed., New York: McGraw-Hill, 2008.

[16] G. F. Knoll, Radiation Detection and Measurements, 3rd ed., John Wiley & Sons, Inc., 2000.

[17] J. Leppänen, "Development of a New Monte Carlo Reactor Physics Code," 2007.

[18] E. Fridman, J. Leppänen and C. Wemple, "Comparison of Serpent and HELIOS-2 as applied for the PWR few-group cross section generation," in *M&C 2013*, Sun Valley, 2013.

[19] G. L. Olson, R. K. McCardell and D. B. Illum, "Fuel Summary Report: Shippingport Light Water Breeder Reactor," Idaho National Lab, Idaho Falls, 2002.

[20] M. B. Chadwick, M. Herman, P. Obložinský, M. E. Dunn, Y. Danon, A. C. Kahler, D. L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D. A. Brown, R. Capote, A. D. Carlson, Y. S. Cho, H. Derrien, K. Guber, G. M. Hale, S. Hoblit, S. Holloway, T. D. Johnson, T. Kawano, B. C. Kiedrowski, H. Kim, S. Kunieda, N. M. Larson, L. Leal, J. P. Lestone, R. C. Little, E. A. McCutchan, R. E. MacFarlane, M. MacInnes, C. M. Mattoon, R. D. McKnight, S. F. Mughabghab, G. P. Nobre, G. Palmiotti, A. Palumbo, M. T. Pigni, V. G. Pronyaev, R. O. Sayer, A. A. Sonzogni, N. C. Summers, P. Talou, I. J. Thompson, A. Trkov, R. L. Vogt, S. C. van der Marck, A. Wallner, M. C. White, D. Wiarda and P. G. Young, "ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data," *Nuclear Data Sheets,* vol. 112, no. 12, pp. 2887-2996, 2011.

[21] International Atomic Energy Agency, "Liquid Metal Cooled Reactors: Experience in Design

and Operation," IAEA, Vienna, 2007.

[22] C. Forsberg, "The Advanced High-Temperature Reactor: High-Temperature Fuel, Liquid Salt Coolant, Liquid-Metal-Reactor Plant," *Progress in Nuclear Energy,* vol. 47, no. 1-4, pp. 32-43, 2005.

[23] W.-P. Chang, Y.-M. Kwon, Y.-B. Lee and D. Hahn, "Model development for analysis of the Korea advanced liquid metal reactor," *Nuclear Engineering and Design,* vol. 217, no. 1-2, pp. 63-80, 2002.

[24] B. A. Lindley, C. Fiorina, F. Franceschini, E. J. Lahoda and G. T. Parks, "Thorium breeder and burner fuel cycles in reduced-moderation LWRs compared to fast reactors," *Progress in Nuclear Energy,* vol. 77, pp. 107-123, 2014.

[25] A. Shelley, S. Shimada, T. Kugo, T. Okubo and T. Iwamura, "Optimization of seed-blanket type fuel assembly for reduced-moderation water reactor," *Nuclear Engineering and Design,* vol. 224, no. 3, pp. 265-278, 2003.

[26] GE Nuclear Energy, "ABWR Plant General Description," GE, 2006.

[27] IAEA, "Status report 97 - Advanced Boiling Water Reactor (ABWR)," 21 July 2011. [Online]. Available: https://aris.iaea.org/PDF/ABWR.pdf. [Accessed 2015].

[28] HITACHI, "Resource-Renewable BWR for Long-term Energy Supply and Effective Use of Transuranic Elements," 2010.

[29] EPRI, "Technical Evaluation of the HITACHI Resource-Renewable BWR (RBWR) Design Concept," EPRI, Palo Alto, 2011.

[30] R. Hill, Interviewee, *Personal Communication.* [Interview]. December 2010.

[31] T. J. Downar and M. Kazimi, "Transient Safety Analysis of Fast Spectrum TRU Burning LWRs with Internal Blankets," NEUP, 2015.

[32] T. J. Downar, D. Lee and Y. Xu, "PARCS v3.0: U.S. NRC Core Neutronics Simulator," 2009.

[33] Y. Xu and T. J. Downar, "GenPMAXS-V5: Code for Generating the PARCS Cross Section Interface File PMAXS," Ann Arbor, 2009.

[34] A. Wysocki, Y. Xu, B. Collins, A. Manera and T. Downar, "PATHS: PARCS Advanced Thermal Hydraulic Solver," 2012.

[35] A. Talamo and W. Gudowski, "Comparative Studies of JENDL-3.3, JENDL-3.2, JEF-2.2 and ENDF," *Nuclear Science and Technology,* vol. 42, no. 12, pp. 1040-1053, 2005.

[36] J. Cetnar, W. Gudowski and A. Talamo, "Comparative Studies of ENDF/B-6.8, JEF-2.2 and JENDL-3.2 Data Libraries by Monte Carlo Modeling of High Temperature Reactors on Plutonium Based Fuel Cycles," *Nuclear Science and Technology,* vol. 41, no. 12, pp. 1228-1236, 2004.

[37] B. Pritychenko, S. F. Mughaghab and A. A. Sonzogni, "Calculations of Maxwellian-averaged cross sections and astrophysical reaction rates using the ENDF/B-VII.0, JEFF-3.1, JENDL-3.3, and ENDF/B-VI.8 evaluated nuclear reaction data libraries," *Atomic Data and Nuclear Data Tables,* vol. 96, no. 6, pp. 645-748, 2009.

[38] A. C. Kahler, "Monte Carlo Eigenvalue Calculations with ENDF/B-VI.8, JEFF-3.0, and JENDL-3.3 Cross Sections for a Selection of International Criticality Safety Benchmark Evaluation Project Handbook Benchmarks," *Nuclear Science and Engineering,* vol. 145, no. 2, pp. 213-224, 2003.

[39] S. C. van der Marck, "Benchmarking ENDF/B-VII.1, JENDL-4.0 and JEFF-3.1.1 with MCNP6," *Nuclear Data Sheets,* vol. 113, no. 12, pp. 2935-3005, 2012.

[40] M. Pusa and J. Leppänen, "An efficient implementation of the Chebyshev rational approximation method (CRAM) for solving the burnup equations," in *PHYSOR 2012*, Knoxville, 2012.

[41] M. Stalek and C. Demaziere, "Development and validation of a cross-section interface for PARCS," *Annals of Nuclear Energy,* vol. 35, no. 12, pp. 2397-2409, December 2008.

[42] N. Brown, *3D Branching,* Brookhaven National Lab, 2012.

[43] B. Herman, E. Shwageraus, B. Forget and M. Kazimi, "Cross Section Generation Strategy for High Conversion Light Water Reactors," Cambridge, 2011.

[44] A. Hall, Y. Xu, A. Ward, T. Downar, K. Shirvan and M. Kazimi, "Advanced Neutronics

Methods for Analysis of the RBWR-AC," in *ANS Transactions*, Atlanta, 2013.

[45] A. Hall, T. Downar, A. Ward, M. Jarret, A. Wysocki, Y. Xu and K. Shirvan, "Advanced Methods Development for Equilibrium Cycle Calculations of the RBWR," in *ICAPP 2014*, Charlotte, 2014.

[46] E. L. Redmond, "Multigroup Cross Section Generation Via Monte Carlo Methods," Cambridge, 1997.

[47] US NRC, "TRACE v5.0: Theory Manual," Washington DC, 2012.

[48] T. M. Sutton, "Wielandt Iteration as Applied to the Nodal Expansion Method," *Nuclear Science and Engineering,* vol. 98, pp. 169-173, 1988.

[49] L. N. Trefethen and D. Bau III, Numerical Linear Algebra, Philadelphia: Society for Industrial and Applied Mathematics, 1997.

[50] J. Y. Cho, H. G. Joo, B.-o. Cho and S. Q. Zee, "Hexagonal CMFD Formulation Employing Triangle-Based Polynomial Expansion Nodal Kernel," in *M&C 2001*, Salt Lake City, 2001.

[51] M. Ishii and T. Hibiki, Thermo-Fluid Dynamics of Two-Phase Flow, New York: Springer, 2011.

[52] K. Shirvan, N. Andrews and M. Kazimi, "Best Estimate Void Fraction and Critical Power Correlations for Tight Lattice BWR Bundles," in *ICAPP*, 2013.

[53] J. K. Fletcher, "A solution of the neutron transport equation using spherical harmonics," *Journal of Physics A: Mathematical and General,* vol. 16, pp. 2827-2835, 1983.

[54] M. Clark and K. F. Hansen, Numerical Methods of Reactor Analysis, New York: Academic Press, 1964.

[55] A. M. Weinberg and E. P. Wigner, The Physical Theory of Neutron Chain Reactors, Chicago: The University of Chicago Press, 1958.

[56] E. Larsen, *NERS 543: Nuclear Reactor Theory II,* Ann Arbor, 2012.

[57] V. Y. Gol'din, "A quasi-diffusion method for solving the kinetic equation," *USSR Computational Mathematics and Mathematical Physics,* vol. 4, no. 6, pp. 136-149,

December 1964.

[58] D. Y. Anistratov and V. Y. Gol'din, "Multilevel Quasidiffusion Methods for Solving Multigroup Neutron Transport k-Eigenvalue Problems in One-Dimensional Slab Geometry," *Nuclear Science and Engineering,* vol. 169, pp. 111-132, 2011.

[59] D. Y. Anistratov and V. Y. Gol'din, "Solution of the Multigroup Transport Equation by the Quasi-Diffusion Method," *USSR Academy of Sciences,* vol. 128, 1986.

[60] D. Y. Anistratov, "Application of the Quasi-Diffusion Approach for Solving Neutron Transport Problems," Moscow, 1992.

[61] A. Iserles, A First Course in the Numerical Analysis of Differential Equations, 2nd ed., Cambridge University Press, 2009.

[62] J.-Y. Cho, K.-S. Kim, C.-C. Lee, S.-Q. Zee and H.-G. Joo, "Axial SPn and Radial MOC Coupled Whole Core Transport Calculation," *Nuclear Science and Technology,* vol. 44, no. 9, pp. 1156-1171, 2007.

[63] M. Hursin, B. Collins, Y. Xu and T. Downar, "The Development and Implementation of a One-Dimensional Sn Method in the 2D-1D Integral Transport Solution," *Nuclear Science and Engineering,* vol. 176, pp. 186-200, 2014.

[64] J. Leppänen and A. Isotalo, "Burnup calculation methodology in the Serpent 2 Monte Carlo code," in *PHYSOR 2012*, Knoxville, 2012.

[65] J. Leppänen, T. Viitanen and V. Valtavirta, "Multi-physics coupling scheme in the Serpent 2 Monte Carlo code," in *Transactions of the American Nuclear Society*, 2012.

[66] J. Leppänen, "Modeling of nonuniform density distributions in the Serpent 2 Monte Carlo code," *Nuclear Science and Engineering,* vol. 174, pp. 318-325, 2013.

[67] J. Leppänen, V. Valtavirta, T. Viitanen and M. Aufiero, "Unstructured Mesh Based Multi-Physics Interface for CFD Code Coupling in the Serpent 2 Monte Carlo Code," in *PHYSOR*, Kyoto, 2014.

[68] J. Leppänen, V. Hovi, T. Ikonen, J. Kurki, M. Pusa, V. Valtavirta and T. Viitanen, "The Numerical Multi-Physics project (NUMPS) at VTT Technical Research Centre of Finland,"

Annals of Nuclear Energy, vol. 84, pp. 55-62, 2015.

[69] E. R. Woodcock, T. Murphy, P. J. Hemmings and T. C. Longworth, "Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry," 1965.

[70] J. Leppänen, "Performance of Woodcock Delta-Tracking in Lattice Physics Applications Using the Serpent Monte Carlo Reactor Physics Burnup Calculation Code," *Annals of Nuclear Energy,* vol. 37, no. 5, pp. 715-722, 2010.

[71] X-5 Monte Carlo Team, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5," Los Alamos National Laboratory, 2008.

[72] A. C. Berry, "The Accuracy of the Gaussian Approximation to the Sum of Independent Variates," *Transactions of the American Mathematical Society,* vol. 49, no. 1, pp. 122-136, 1941.

[73] J. A. Favorite, "On the Accuracy of a Common Monte Carlo Surface Flux Grazing Approximation," *Nuclear Science and Engineering,* vol. 168, pp. 115-127, 2011.

[74] F. H. Clark, "Variance of Certain Flux Estimators Used in Monte Carlo Calculations," *Nuclear Science and Engineering,* vol. 27, pp. 235-239, 1967.

[75] J. Leppänen and M. Pusa, "Burnup Calculation Capability in the PSG2/Serpent Monte Carlo Reactor Physics Code," in *International Conference on Mathematics, Computational Methods & Reactor Physics*, Saratoga Springs, 2009.