Copyright
by
Christopher Lee Dembia
2012

The Thesis Committee for Christopher Lee Dembia certifies that this is the approved version of the following thesis:

A Multi-Region Collision Probability Method For Determining Neutron Spectra and Reaction Rates

APPROVED BY

SUPERVISING COMMITTEE:

Mark Deinert, Supervisor

Erich Schneider

A Multi-Region Collision Probability Method For Determining Neutron Spectra and Reaction Rates

by

Christopher Lee Dembia, B.S.

THESIS
Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN
August 2012

Dedicated to Geoff Tanner and Tasha Hawthorne.

Acknowledgments

Over the last two years, Geoff Recktenwald and Mark Deinert have been my academic parents. They have been exceedingly patient, kind, understanding, optimistic, and friendly. Most importantly, they have always made time to give me advice. It is for this that I am most grateful. They certainly have had a lasting impact on my life, and have taught me most of what I know about academia.

Thank you to the denizens of French House for providing me with a friendly home in a familiar cooperative setting. Thank you to Luo Yun for spending nights with me in the Perry Castañeda Library.

Thank you to Andy Tang for always trying to talk to me.

A Multi-Region Collision Probability Method For Determining Neutron Spectra and Reaction Rates

Christopher Lee Dembia, M.S.E.
The University of Texas at Austin, 2012

Supervisor: Mark Deinert

The collision probability approach to neutron transport can be used to obtain the energy-dependent neutron spectrum in nuclear reactor systems as well as other quantities of interest. This method makes the approximation that the neutron distribution is constant within homogeneous regions, or cells, in the system. This assumption restricts geometries that can be modeled by the collision probability approach. The geometry modeled is typically an infinite lattice of two homogeneous cells: a fuel pin cylinder and the coolant that surrounds it. The transport of neutrons between the homogeneous cells is done using probabilities describing the chance that a neutron having a collision in one cell has its next collision in another cell. These collision probabilities can be cast in terms of escape and transmission probabilities for each cell.

Some methods exist that extend the collision probability approach to systems composed of more than two homogeneous cells. In this work, we present a novel collision probability method, based on previous work by Schneider et al.

(2006a), for an arbitrary number of cells. The method operates by averaging the transmission probabilities across cells of the same shape, and thus assumes a certain level of homogeneity across all cells.

When using multigroup cross sections, which the collision probability approach requires, it is necessary to consider the effect that a system's geometry and composition has on those multigroup cross sections. The cross sections must be computed in a way that accounts for the resonance self-shielding that may reduce the reaction rates in the resonance region. The process of developing self-shielded cross sections in a heterogeneous system utilizes an escape cross section. We compute this escape cross section using the same collision probabilities used to obtain the energy spectrum.

Results are presented for simple two-cell systems, and preliminary results for four-cell simulations are also given. An extension to the method is provided that accounts for the fact that in thermal systems the assumption of homogeneity is not always valid.

Table of Contents

Acknowledgments **v**

Abstract **vi**

List of Tables **xi**

List of Figures **xii**

Chapter 1. Introduction **1**

1.1 Popular Codes 2
1.2 Background 2
1.3 Overview 4

Chapter 2. Neutron Transport with Collision Probabilities **5**

2.1 The Transport Equation 6
2.2 Direction 9
  2.2.1 The Transport Correction 14
2.3 Energy: Multigroup Transport 18
  2.3.1 The Multigroup Transport Correction 22
2.4 Space: The Integral Transport Equation 24
  2.4.1 Matrix Representation 27
2.5 Collision Probabilities 29
  2.5.1 Infinite Lattice Geometry 30
  2.5.2 Escape and Transmission 31
  2.5.3 Self-Collision 34
  2.5.4 Remaining Elements of the Transport Matrix 36
  2.5.5 Rational Approximations 38
  2.5.6 Reciprocity and the Transmission Probability 42
2.6 The Homogeneity Correction 43

Chapter 3. Self-Shielded Multigroup Cross Sections 48
3.1 Resonance Self-Shielding 49
3.2 Flux Approximation 50
3.3 The Background Cross Section 52
3.4 Heterogeneous Systems and the Equivalence Relation 54
3.5 The Escape Cross Section 55
3.6 The Self-Shielding Factor 57
3.7 Implementation 59

Chapter 4. Results 63
4.1 Two-Cell Reactors 63
  4.1.1 A Light Water Thermal Reactor 64
  4.1.2 A Sodium Cooled Fast Reactor 66
  4.1.3 Light Water Reactor Cross Sections 66
  4.1.4 Fast Reactor Cross Sections 70
  4.1.5 Thermal Spectrum and Reaction Rates 70
  4.1.6 Fast Reactor Spectrum and Reaction Rates 75
  4.1.7 Criticality 76
4.2 A Four-Cell Reactor 77
  4.2.1 Fast Reactor 77
  4.2.2 Thermal Reactor 77

Appendices 86

Appendix A. Relevant Cross Sections 87

Appendix B. Code Manual 93
B.1 Modules 93
  B.1.1 Dataprocessing 96
  B.1.2 Selfshielding 97
  B.1.3 Multicell 97
B.2 Data Structures 98
  B.2.1 Parameter Input 100

        B.2.2 Geometry Input 101
        B.2.3 Results 104
        B.2.4 Data Library 108
    B.3 Utility Functions 110
    B.4 Development Options 111
    B.5 Further Work 116
    B.6 Development Notes 119

Index **122**

Bibliography **124**

Vita **128**

List of Tables

4.1 Light water reactor dimensions and composition 65
4.2 Sodium fast reactor dimensions and composition 66
4.3 Light water reactor reaction rates 74
4.4 Light water reactor reaction rates with MCNPX 74
4.5 Sodium fast reactor reaction rates 76
4.6 Multiplication factor comparison 77
4.7 Four-cell thermal reactor dimensions and composition 81

List of Figures

2.1 Illustration of the transport correction 18
2.2 Escape probability approximations 41
2.3 Geometry for homogeneity correction 43
2.4 Homogeneity 46

3.1 Self-shielding factor for U-238 escape 58
3.2 Point-wise cross section near 6.67 eV for U-238 capture 59
3.3 Self-shielding procedure flowchart 60

4.1 Two-cell system 64
4.2 Self-shielded group cross section for U-238 capture 68
4.3 Self-shielded group cross section for U-235 capture 69
4.4 Self-shielded group cross section for U-238 capture 71
4.5 Self-shielded group cross section for U-235 capture 72
4.6 Light water reactor spectral flux 73
4.7 Sodium fast reactor spectral flux 75
4.8 Four-cell fast reactor dimensions and composition 78
4.9 Four-cell fast reactor spectral flux in uranium 79
4.10 Four-cell fast reactor spectral flux in plutonium 80
4.11 Four-cell thermal reactor spectral flux in the UOX fuel 82
4.12 Four-cell thermal reactor spectral flux in the IMF fuel 83
4.13 Four-cell thermal reactor spectral flux in the UOX fuel, homogeneity corrected 84
4.14 Four-cell thermal reactor spectral flux in the IMF fuel, homogeneity corrected 85

A.1 Total cross section for H-1 bound in water 88
A.2 Total cross section for O-16. 88
A.3 Total cross section for Na-23. 89

A.4 Total cross section for Cr-52. 89
A.5 Total cross section for Fe-56. 90
A.6 Total cross section for Ni-58. 90
A.7 Total cross section for Zr-90. 91
A.8 Total cross section for U-235. 91
A.9 Total cross section for U-238. 92

B.1 VBUDSII modules 94
B.2 VBUDSII data structures 99

Chapter 1

Introduction

Nuclear reactor physics is concerned with obtaining the spatial and spectral neutron distribution in a reactor. This information is critical for reactor design and analysis, and can be used by reactor operators on a daily basis, but may also be used in the attempt to license the use of new fuels through the National Regulatory Commission or another regulatory agency. As an example, one may use this information to discover how a reactor's composition evolves in time.

This work focuses on only one portion of a reactor analysis code: the spectral lattice calculation. Here one seeks a fine description of the neutron distribution as a function of energy, but assumes that the variation of the distribution in space is minimal. So-called collision probability methods are well-suited to such calculations. The method accounts for the spatial variable using probabilities that describe the chance that a neutron moves from region or cell of the system to another. Traditionally, such codes assume that a lattice is composed of two cells: a fuel pin and the surrounding coolant. We present a novel collision probability method that allows for the system to be modeled with more than two cells.

1.1 Popular Codes

The texts by Bell and Glasstone (1970) and Stamm'ler and Abbate (1983) both describe the common modules of reactor analysis codes. The use of the collision probability method, while it makes large assumptions about a system's configuration, is commonly a part of most large codes. Even for geometries that are heterogeneous on the assembly level, a collision probability method is often used before doing a whole core calculation. The review (Hébert, 2007) provides a history of popular lattice codes, listing codes such as WIMS and the open-source DRAGON code with which Hébert is associated.

Of particular interest is the collision probability formulation given by Stamm'ler and Abbate (1983) that extends the collision probability approach to multiple annular regions so that a coarse flux distribution in the fuel pin is obtained. This formulation was recently implemented by Ozgener and Ozgener (2005).

1.2 Background

This work is a continuation of an ongoing project to develop a reactor physics code for rapid fuel cycle scoping calculations. The original code that came from this work, VBUDS, could be used to predict the evolution of a reactor's material composition, or perform what is known as a burnup calculation. Such calculations require a detailed knowledge of the energy distribution of neutrons in the system. This was obtained for a two-cell (fuel pin and moderator) geometry with just over one thousand energy groups.

The work resulted in a number of publications on the method itself and on its application to nuclear fuel cycle studies (Schneider et al., 2005, 2006a,b). Since then, a number of students have extended this work in different directions. Barratt (2003), who was at Cornell University toward the end of Dr. Schneider's doctoral work, studied the calculation of the Dancoff factor, a necessary component of a general collision probability formulation. Matavosian (2007) performed his doctorate research at the University of Texas at Austin on a collision probability method that incorporates a number of annular sub-regions. Most recently, Saller (2009) and Dembia et al. (2012) have performed work on resonance self-shielding calculations for collision probability models.

In the present work, a code has been developed that can produce a neutron spectrum and reaction rates for general simple lattice geometries. It is intended that the code performs burnup calclulations, as does (Schneider, 2002), but due to time limitations such functionality is not incorporated into the work.

The task of obtaining the neutron spectrum requires that the multigroup cross section data takes into account the nature of the system being studied. This is necessitated by the well-known resonance self-shielding effect, by which cross sections of one nuclide affect the flux in the cell and thereby alter the reaction rates for the other nuclides in the same cell. In a heterogeneous system, the fact that neutrons can move between cells further affects the computation of effective multigroup cross sections. This is treated by introducing an escape cross section that describes how likely a neutron is to escape a cell. A number of different methods can be used to compute the escape cross section; we use the same collision probabilties from the spectral solution for this purpose.

1.3 Overview

This work is divided into a chapter on the novel multi-region (or multi-cell) collision probability method, and a chapter on the processing of multigroup cross sections for this method. It is assumed the reader is comfortable with the fundamental nuclear engineering concepts of flux and cross section. Results for a few different reactor types are presented. For simple geometries, the method yields excellent results in comparisons to MCNPX.

Chapter 2

Neutron Transport with Collision Probabilities

The objective of this chapter is to formulate a method for obtaining the neutron flux distribution in energy and in all discrete cells of a system. We start at the beginning, with the complete steady-state differential neutron transport equation for the angular neutron flux. This equation is a function of space, energy, and direction. The form of the equation that we will use is substantially simplified from this initial form. We address the treatment of the transport equation for each of the three variables in turn.

In deriving the form of the transport equation that we use in our collision probability the method, we introduce a number of unknowns that describe the probability that neutrons move between discrete cells in the system. The second half of this chapter focuses on deriving expressions for these unknowns. It is this portion of the work that includes novel contributions to the theory of collision probabilities.

Much of this section was informed by the texts by Bell and Glasstone (1970) and Stamm'ler and Abbate (1983), which are both very popular in the field. In particular, the Stamm'ler text provides very clear and concise explanations of some of the most important concepts. Unfortunately, the Stamm'ler text is out of print.

2.1 The Transport Equation

The steady state neutron transport equation, in its most general form, casts the neutron flux as a function of space, direction, and neutron energy (Schneider, 2002). It is derived through the conservation of neutrons about **r** {cm} with energy about E {eV} traveling in a direction about $\mathbf{\Omega}$ {ster}:

$$\begin{aligned}\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},E,\mathbf{\Omega})+\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\mathbf{\Omega}) = &\\ \int_{4\pi} d\mathbf{\Omega}' \int_0^{\infty} dE'\,\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}')\psi(\mathbf{r},E',\mathbf{\Omega}') &\\ +\chi(E)\int_{4\pi} d\mathbf{\Omega}' \int_0^{\infty} dE'\,\nu\frac{\Sigma_f(\mathbf{r},E')}{4\pi}\psi(\mathbf{r},E',\mathbf{\Omega}') &\end{aligned} \tag{2.1}$$

where ψ {#/cm^2/s/eV/ster} is the angular neutron flux. The left hand side terms represent neutron sinks, and the right hand side terms represent neutron sources. More accurately, the left hand side provides the total interaction rate; the term includes self-scattering which is not a sink. We look at each term in Equation 2.1 in turn.

Streaming $\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},E,\mathbf{\Omega})$

This accounts for the net flow of neutrons out of a differential volume element dV at **r** in the direction given by the unit vector $\mathbf{\Omega}$.

Removal $\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\mathbf{\Omega})$

This represents the chance that a neutron in the volume element dV about $\mathbf{r}$ with energy about E traveling in a direction about $\mathbf{\Omega}$ undergoes a collision that removes it from the system, changes its energy, or changes its direction of travel. The total cross section $\Sigma_t(\mathbf{r},E)$ $\{\mathrm{cm}^{-1}\}$ is the sum of all other cross sections (i.e. scattering and absorption).

Scattering $\int_{4\pi} d\mathbf{\Omega}' \int_0^\infty dE'\, \Sigma_s(\mathbf{r},E \leftarrow E',\mathbf{\Omega} \leftarrow \mathbf{\Omega}')\psi(\mathbf{r},E',\mathbf{\Omega}')$

This accounts for the chance that a neutron in the volume element dV about $\mathbf{r}$ with any energy E' traveling in any direction $\mathbf{\Omega}'$ may collide with a nucleus that causes it to have an energy about E and a direction of travel about $\mathbf{\Omega}$. The double differential scattering cross section $\Sigma_s(\mathbf{r},E \leftarrow E',\mathbf{\Omega} \leftarrow \mathbf{\Omega}')$ $\{\mathrm{cm}^{-1}/\mathrm{eV}/\mathrm{ster}\}$ encapsulates the sometimes complex nature scattering. It is called double differential because it describes the chance of a scatter into (or from) both a differential energy range and into (or from) a differential direction of travel. This term represents a source into E and $\mathbf{\Omega}$, and accounts for neutrons from energy or direction of travel before it scattered. For this reason, this quantity is integrated over all possible incident energies and solid angles. Note that we have used left-pointing arrows rather than the conventional right-pointing arrows in the arguments of the scattering cross section. This notation allows for a cleaner transition to a computational implementation of the transport equation.

Fission $\chi(E)/(4\pi)\int_{4\pi} d\mathbf{\Omega}' \int_0^\infty dE'\, \nu/\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\mathbf{\Omega}')$

This accounts for the chance that a neutron in the volume element dV

about $\mathbf{r}$ with any energy E' may collide with a nucleus and induce a fission that produces $\chi(E)\nu(E')$ neutrons at $\mathbf{r}$ with energy E in direction $\mathbf{\Omega}$. The fission spectrum $\chi(E)$ is a (unitless) probability distribution describing the energies of fission neutrons. The quantity ν (unitless) is the average number of fission neutrons released per fission and is a weak function of E'. $\Sigma_f(\mathbf{r}, E')$ $\{\text{cm}^{-1}\}$ describes the probability of fission occurring. Often, ν and Σ_f are grouped together. For this reason, ν is not shown as an independent function of energy. The factor of 4π, which has units of steradians, comes from the fact that the fission cross section is independent of direction. This can be understood by expressing $\Sigma_f(\mathbf{r}, E', \mathbf{\Omega}') = \Sigma_f(\mathbf{r}, E')/4\pi$.

The only unknown in Equation 2.1 is the angular flux ψ, as all other quantities are properties of the materials in the system and are obtained from a mix of theory and experiment. With the flux, one can model the power distribution in a reactor, or one can find reaction rates for particular interactions (e.g. absorption, fission) to model the evolution of a system's composition over time.

Equation 2.1 is not amenable to analytical solutions for systems of practical interest. To make practical use of the relation, it is often discretized in one or more of its variables and solved using computational methods. There are many ways to discretize the relation, and the method chosen depends on our objective. As for the direction variable, both diffusion theory and the spherical harmonics expansion treat the angular dependence of the flux in an approximate sense. Alternatively, one can use the method of discrete ordinates to discretize the equation in direction and therefore obtain a detailed angular dependence. With all of these methods, it is common to discretize the equation in the energy variable to form a set of multigroup equations (Bell and Glasstone, 1970). In this work, we are ultimately concerned with obtaining a flux that can be used to model the evolution of the reactor's composition. This requires a detailed knowledge of the energy dependence of the flux, but only very coarse knowledge of its spatial dependence.

In the remainder of this chapter, we develop a form of the transport equation that provides a detailed energy description of the flux but that approximates its spatial distribution through the use of collision probabilities. In order to do this, we address each of the variables $\mathbf{r}$, E, and $\mathbf{\Omega}$ separately. First, we remove the angular dependence of the transport equation. Then we discretize the resulting equations in energy to form general multigroup equations. Then, we account for the spatial dependence with probabilities that describe how neutrons move between discrete uniform cells in the system. This last step will provide what is known as an integral transport equation.

2.2 Direction

We are interested in obtaining the energy and spatial dependence of the flux, but not its angular dependence. Therefore, we seek a way to remove this variable from the transport equation. We replace the angular flux $\psi(\mathbf{r}, E, \mathbf{\Omega})$ with a flux $\phi(\mathbf{r}, E)$ integrated over all directions $\mathbf{\Omega}$.

$$\phi(\mathbf{r}, E) = \int_{4\pi} d\boldsymbol{\Omega}\, \psi(\mathbf{r}, E, \boldsymbol{\Omega}) \tag{2.2}$$

Accordingly, $\phi(\mathbf{r}, E')$ has units of $\{\#/\mathrm{cm}^{-2}/\mathrm{s}/\mathrm{eV}\}$. The 4π below the integral symbol indicates the integral is performed over all solid angles. We attempt to remove the transport equation's dependence on direction by integrating the entire equation over all directions $\boldsymbol{\Omega}$. We examine each term in Equation 2.1 in turn.

Streaming

Beginning with:

$$\int_{4\pi} d\boldsymbol{\Omega}\, \boldsymbol{\Omega} \cdot \nabla\psi(\mathbf{r}, E, \boldsymbol{\Omega})$$

we observe that we can rearrange this expression as follows:

$$\nabla \cdot \int_{4\pi} d\boldsymbol{\Omega}\, \boldsymbol{\Omega}\psi(\mathbf{r}, E, \boldsymbol{\Omega})$$

and define the net current $\mathbf{J}(r, E)$ to be:

$$\mathbf{J} = \int_{4\pi} d\boldsymbol{\Omega}\, \boldsymbol{\Omega}\psi(\mathbf{r}, E, \boldsymbol{\Omega})$$

so that the streaming term can be written independently of direction as:

$$\nabla \cdot \mathbf{J}(\mathbf{r}, E)$$

Removal

Beginning with:

$$\int_{4\pi} d\boldsymbol{\Omega}\, \Sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, E, \boldsymbol{\Omega})$$

we observe that the total cross section is independent of direction:

$$\Sigma_t(\mathbf{r},E)\int_{4\pi} d\mathbf{\Omega}\,\psi(\mathbf{r},E,\mathbf{\Omega})$$

and use the definition of the $\phi(\mathbf{r},E)$ to obtain:

$$\Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E)$$

Scattering

Beginning with:

$$\int_{4\pi} d\mathbf{\Omega}\int_{4\pi} d\mathbf{\Omega}'\int_0^\infty dE'\,\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}')\psi(\mathbf{r},E',\mathbf{\Omega}')$$

we move around the energy integral and the angular flux:

$$\int_0^\infty dE'\int_{4\pi} d\mathbf{\Omega}'\,\psi(\mathbf{r},E',\mathbf{\Omega}')\int_{4\pi} d\mathbf{\Omega}\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}')$$

and express the right two integrals in the simpler notation:

$$\Sigma_s(\mathbf{r},E\leftarrow E')\phi(\mathbf{r},E')$$

thereby defining the single differential scattering cross section as:

$$\Sigma_s(\mathbf{r},E\leftarrow E')=\frac{1}{\phi(\mathbf{r},E')}\int_{4\pi} d\mathbf{\Omega}'\psi(\mathbf{r},E',\mathbf{\Omega}')\int_{4\pi} d\mathbf{\Omega}\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}')$$

so that the scattering term can be written independently of direction as:

$$\int_0^\infty dE'\,\Sigma_s(\mathbf{r},E\leftarrow E')\phi(\mathbf{r},E')$$

Fission

Beginning with:

$$\frac{\chi(E)}{4\pi}\int_{4\pi} d\mathbf{\Omega}\int_{4\pi} d\mathbf{\Omega}'\int_0^\infty dE'\nu\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\mathbf{\Omega}')$$

we observe that the $\nu\Sigma_f$ cross section is independent of direction:

$$\frac{\chi(E)}{4\pi}\int_0^\infty dE'\,\nu\Sigma_f(\mathbf{r},E')\int_{4\pi} d\mathbf{\Omega}\int_{4\pi} d\mathbf{\Omega}'\psi(\mathbf{r},E',\mathbf{\Omega}')$$

and use the definition of the flux $\phi(\mathbf{r},E)$ to obtain:

$$\frac{\chi(E)}{4\pi}\int_0^\infty dE'\,\nu\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')\int_{4\pi} d\mathbf{\Omega}$$

and we observe that no terms depend on the outgoing direction $\mathbf{\Omega}$:

$$\chi(E)\int_0^\infty dE'\,\nu\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')$$

Combining the results above yields the following for the neutron transport equation independent of $\mathbf{\Omega}$:

$$\begin{aligned}\nabla\cdot\mathbf{J}(\mathbf{r},E)+\Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E) &= \int_0^\infty dE'\,\Sigma_s(\mathbf{r},E\leftarrow E')\phi(\mathbf{r},E') \\ &\quad + \chi(E)\int_0^\infty dE'\,\nu\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')\end{aligned} \tag{2.3}$$

This introduces an additional unknown, the neutron current $\mathbf{J}$. This vector quantity can be understood as a direction-weighted flux that points in the direction of the net flow of neutrons. One way to manage this additional unknown is to introduce the diffusion approximation, which removes the current $\mathbf{J}$ as an unknown by relating it to the flux ϕ. Given the method we use to discretize space in Section 2.4, we will be able to ignore the streaming term, and so it is not discussed further here.

Fission is adequately assumed to emit neutrons isotropically, and so this term does not have an angular dependence and the fission cross section was removed from the integral over direction above.

Scattering, on the other hand, cannot always be assumed to be isotropic. In particular, scattering is quite forward-biased with any target at high energies or with hydrogen at epithermal and higher energies. Accordingly, the differential scattering cross section in Equation 2.3, with units of {cm^{-1}/eV}, depends on direction and even the angular flux. Though it is not always valid, let us briefly consider the case of isotropic scattering. In this case the single differential scattering cross section reduces to:

$$\Sigma_s(\mathbf{r},E\leftarrow E')=\frac{\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}')}{\phi(\mathbf{r},E')}\int_{4\pi}d\mathbf{\Omega}'\psi(\mathbf{r},E',\mathbf{\Omega}')\int_{4\pi}d\mathbf{\Omega} \tag{2.4}$$

The first integral is simply $\phi(\mathbf{r},E')$, and the second integral evaluates to 4π. Thus the single differential and the double differential cross sections differ simply by the total solid angle over all directions.

$$\Sigma_s(\mathbf{r},E\leftarrow E')=4\pi\Sigma_s(\mathbf{r},E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}') \tag{2.5}$$

It is clear from this expression that the single differential scattering cross section $\Sigma_s(\mathbf{r}, E \leftarrow E')$ has units of {cm^{-1}/eV}. This situation is clearly preferable, as it allows us to remove the dependence on the angular flux of the scattering term. It is possible to assume linear anisotropy and still use this convenient form of the scattering cross section. The procedure for doing this is called the transport correction (Stamm'ler and Abbate, 1983).

2.2.1 The Transport Correction

In most cases, it can be safely assumed that the angular dependence of the scattering cross section is only through one variable, the cosine of the scattering angle $\mu = \mathbf{\Omega} \cdot \mathbf{\Omega}'$. As such, the scattering cross section function can be represented by a Legendre polynomial expansion in μ (Bell and Glasstone, 1970).

$$\Sigma(E \leftarrow E', \mu) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \Sigma_n(E \leftarrow E') P_n(\mu) \tag{2.6}$$

The subscript s and the position argument $\mathbf{r}$ have been removed for clarity. The double differential scattering cross section $\Sigma(E \leftarrow E', \mu)$ has units of {cm^{-1}/eV/ster}. The quantity P_n represents the Legendre polynomial of order n. The quantity Σ_n represents the coefficient for the n-th order term with units of {cm^{-1}/eV}, and is independent of μ. The first Legendre polynomial is defined as simply $P_0 = 1$ and the second polynomial is $P_1(\mu) = \mu$. In the case of isotropic scattering, only the $n = 0$ coefficient, Σ_0, is non-zero and Equation 2.6 reduces to Equation 2.5. If only the first two terms are kept, then the expansion yields:

$$\Sigma(E \leftarrow E', \mu) = \frac{1}{4\pi}\Sigma_0(E \leftarrow E') + \mu\frac{3}{4\pi}\Sigma_1(E \leftarrow E') \tag{2.7}$$

This provides an approximate dependence of the scattering from E' to E as a function of the scattering cosine. Since this equation is linear in μ, this case is called linear anisotropic scattering. The magnitude of Σ_1 is typically very small, but it cannot be neglected for scattering with any material at high energies, or for scattering with hydrogen at most energies.

In the method of spherical harmonics, and sometimes in the method of discrete ordinates, a number of terms from the Equation 2.6 expansion are inserted into the transport equation, Equation 2.1. It is convenient in such cases to have a polynomial representation of the scattering cross section rather than some other complicated function, as it allows a simple explicit treatment of the angular dependence of scattering. However since we are avoiding the use of the angular flux and Equation 2.7 is a function of direction, we cannot use the expansion explicitly. We must either assume isotropic scattering so that Σ_s is independent of μ, or we must employ what is known as the transport correction.

We seek to replace Σ_1 in Equation 2.7 with the average value of the scattering angle for scattering occurring at energy E', $\bar{\mu}(E')$. This quantity is obtained by applying the average value theorem:

$$\bar{\mu}(E') = \frac{\int_0^\infty dE \int_{-1}^1 d\mu\, \mu(E')\Sigma(E \leftarrow E')}{\int_0^\infty dE \int_{-1}^1 d\mu \Sigma(E \leftarrow E')} \tag{2.8}$$

If $\Sigma(E \leftarrow E')$ is given by the Legendre expansion in Equation 2.6, then Equation 2.8 conveniently reduces to:

$$\bar{\mu}(E') = \frac{\Sigma_1(E')}{\Sigma_0(E')} \tag{2.9}$$

This is a result of the orthogonality of the Legendre polynomials. Here, $\Sigma_0(E')$ and $\Sigma_1(E')$ are obtained by integrating over all outgoing energies. Inserting Equation 2.9 into the linear anisotropy approximation of Equation 2.7 yields:

$$\begin{aligned}\Sigma(\mu) =& \frac{1}{4\pi}\left(\Sigma_0 + 3\mu\bar{\mu}\Sigma_0\right) \\ =& \frac{1}{4\pi}\Sigma_0(1 + 3\mu\bar{\mu})\end{aligned} \tag{2.10}$$

where all quantities (except the μ input) are functions of E' and E, but the notation has been omitted for clarity. At energies where scattering can be assumed to be isotropic in the center-of-mass frame of the collision, $\bar{\mu}$ is given by a constant in energy:

$$\bar{\mu} = \frac{2}{3A} \tag{2.11}$$

where A is the atomic number of the target. However, at high neutron energies, center-of-mass isotropy cannot be assumed and doing so leads to substantial errors. In such cases the value of $\bar{\mu}$ can be obtained from a nuclear data processing code such as NJOY (MacFarlane and Muir, 1994). This new form of the linear anisotropy approximation is more convenient since Σ_1 has been removed, Σ_0 is simply the total scattering cross section, and $\bar{\mu}$ is given by Equation 2.11 or is tabulated.

In the transport correction we make the crude approximation that the total scattering cross section Σ_s is the sum of a term that represents the isotropic portion of the scattering, Σ_s^{tr}, and a term that accounts for solely forward scattering (i.e. no interaction), $\bar{\mu}\Sigma_s$. This allows us to express the quantity Σ_s^{tr} in terms of known quantities:

$$\Sigma_s^{tr}(E) = \Sigma_s(E) - \bar{\mu}(E)\Sigma_s(E) \tag{2.12}$$

The quantity Σ_s^{tr} is called the transport corrected total scattering cross section, and it can be used as an effective isotropic scattering cross section. It is clear that as scattering becomes isotropic, Σ_{tr} approaches Σ_s.

The transport correction is illustrated in Figure 2.1, which shows that the transport correction yields a cross section in which the anisotropy of the scattering cross section has been removed. This is how the transport correction is discussed by Stamm'ler and Abbate (1983).

The transport correction can be obtained by a different method, as is presented by Lamarsh (1965). In this method, one derives a transport *cross section* $\Sigma_{tr}(E)$ from a mean free path that describes the distance a neutron travels before it "forgets" its original direction of travel. This new cross section can be

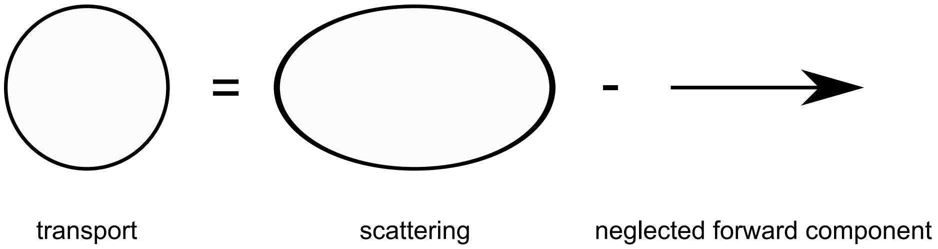

Figure 2.1: An illustration of the transport correction. The transport corrected scattering cross section is obtained by subtracting the forward-scattering component from the anisotropic scattering cross section. Alternatively, a transport cross section can be defined using the average distance that a neutron moves in a particular direction.

used in place of the total cross section when necessary:

$$\Sigma_{tr}(E) = \Sigma_t(E) - \bar{\mu}(E)\Sigma_s(E) \tag{2.13}$$

or equivalently

$$\Sigma_{tr}(E) = \Sigma_a(E) + \Sigma_s(E)(1 - \bar{\mu}(E)) \tag{2.14}$$

The author finds that for a computer implementation of the transport correction the formulation by Stamm'ler is clearer. Both yield the same total cross section, but the derivation by Lamarsh does not indicate that the scattering events themselves should also be treated differently. In Section 2.3.1, the transport correction is adapted to multigroup cross sections.

2.3 Energy: Multigroup Transport

With a simpler direction-independent form of the transport equation, we now turn to the task of obtaining the flux's energy dependence. Beyond this step, we are left only with the task of discretizing the equation in space.

We begin by discretizing all quantities into G energy groups that span a range of energies from E_G to E_0. The g-th energy group spans an energy range from E_g to E_{g-1}. As is typical, the groups are ordered in descending energy so that the highest energy is E_0 and the lowest is E_G, and in general $E_g < E_{g-1}$. The width of an energy group is $\Delta E_g = E_{g-1} - E_g$ and is not the same across groups.

The flux ϕ_g is now defined discretely for each energy group through the continuous flux:

$$\phi_g(\mathbf{r}) = \int_g dE\, \phi(\mathbf{r}, E) \tag{2.15}$$

An integral over g indicates an integral from E_g to E_{g-1}. The group flux $\phi_g(\mathbf{r})$ has units of $\{\#/\text{cm}^2/\text{s}\}$ as it has been integrated over energy. The position argument $\mathbf{r}$ is temporarily removed for clarity in the proceeding equations, though all quantities may still be functions of position. Logically, the group flux is larger for a larger ΔE_g with the same number of neutrons per energy. We must also obtain cross sections that depend on energy group rather than on a continuous energy domain to form so-called group constants. We use σ_g and Σ_g to represent, respectively, the microscopic and macroscopic group cross section in group g for any interaction. In this case, it is not appropriate to simply integrate the value of the cross section over the energy group. The group cross sections must be defined in such a way that they preserve the reaction rates $R(E) = N\sigma(E)\phi(E)$ of the interactions that occur in the system. It is the reactions in the system that define its criticality, its power distribution, and the evolution of its composition. We can attempt to preserve the reaction rates by enforcing that the product of the group cross section and group flux is equal to the integral of the continuously-defined reaction rate over the group.

$$\sigma_g \phi_g = \int_g dE\, \sigma(E)\phi(E) \tag{2.16}$$

The constraint of Equation 2.16 defines the group cross section σ_g (and consequently Σ_g) for any interaction as follows:

$$\sigma_g = \frac{\int_g dE\, \sigma(E)\phi(E)}{\phi_g} = \frac{\int_g dE\, \sigma(E)\phi(E)}{\int_g dE\, \phi(E)} \tag{2.17}$$

The group cross section σ_g has units of barns, and does not increase with increasing group width for constant $\sigma(E)$. The group to group scattering cross section is defined analogously, and may be called a group transfer cross section.

$$\sigma_{s,g\leftarrow g'} = \frac{\int_g dE \int_{g'} dE'\, \sigma_s(E \leftarrow E')\phi(E)}{\phi_g} \tag{2.18}$$

This group transfer cross section has units of {b/eV}. Equipped with group quantities, we are prepared to express the transport equation as a set of multigroup equations. We integrate each term in Equation 2.3 over group g so that all quantities within a are independent of energy. We reintroduce the position argument **r** for completeness.

Streaming

$$\int_g dE\, \nabla \cdot \mathbf{J}(\mathbf{r}, E) = \nabla \cdot \mathbf{J}_g(\mathbf{r})$$

Removal

$$\int_g dE\, \Sigma_t(\mathbf{r}, E)\phi(\mathbf{r}, E) = \Sigma_{tg}(\mathbf{r})\phi_g(\mathbf{r})$$

Scattering

We find the following with the help of Equation 2.18:

$$\int_g dE \int_0^\infty dE'\, \Sigma_s(\mathbf{r}, E \leftarrow E')\phi(\mathbf{r}, E') = \sum_{g'}^{G} \Sigma_{s,g\leftarrow g'}(\mathbf{r})\phi_{g'}(\mathbf{r})$$

Fission

$$\int_g dE\, \chi(E) \int_0^\infty dE'\, \nu\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') = \chi_g \sum_{g'}^{G} \nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r})$$

The quantity $\nu\Sigma_f(E)$ is group-averaged as a single quantity to form $\nu\Sigma_{fg}$.

As mentioned before, the method by which we discretize the spatial variable does not rely on the neutron current **J**. Therefore, it is not necessary to define $\mathbf{J}_g(\mathbf{r})$. The terms above are combined to provide G coupled differential equations for the G unknowns ϕ_0 through ϕ_G. These equations are coupled through the scattering and fission sources. The position argument **r** is removed for clarity.

$$\nabla \cdot \mathbf{J}_g + \Sigma_{tg}\phi_g = \sum_{g'}^{G} \Sigma_{s,g\leftarrow g'}\phi_{g'} + \chi_g \sum_{g'}^{G} \nu\Sigma_{fg'}\phi_{g'} \tag{2.19}$$

$$= \sum_{g'}^{G} \left(\Sigma_{s,g\leftarrow g'} + \chi_g \nu\Sigma_{fg'}\right)\phi_{g'} \tag{2.20}$$

For brevity, we define a source term S_g to be the source of neutrons into group g from any other group g' (including $g' = g$).

$$S_g = \sum_{g'}^{G} \left(\Sigma_{s,g\leftarrow g'} + \chi_g \nu \Sigma_{f g'} \right) \phi_{g'} \tag{2.21}$$

Now, Equation 2.19 can be expressed in a simpler form.

$$\nabla \cdot \mathbf{J}_g + \Sigma_{tg} \phi_g = S_g \tag{2.22}$$

All terms have units of $\{\#/\mathrm{s}/\mathrm{cm}^3\}$. This can easily be seen as a conservation of neutrons: all neutrons that enter group g must eventually collide in group g or stream away. This includes neutrons that undergo a scattering event that does not change its energy enough to change its group (in-group scattering).

A subtlety in this derivation is that the quantities Σ_{tg}, $\Sigma_{s,g\leftarrow g'}$, and $\nu\Sigma_{fg}$ actually depend on $\phi(E)$, and so they are not known quantities in Equation 2.19. To obtain these group constants, we must initially approximate $\phi(E)$. The procedure for this is discussed in Chapter 3.

2.3.1 The Multigroup Transport Correction

We adapt the transport correction of Section 2.2.1 to the multigroup formulation. Previously, the transport correction was formulated only for total scattering. Here, we apply the correction to the group transfer cross section only.

We subtract the forward-scattering component from only the in-group scattering terms, $\Sigma_{s,g\leftarrow g'}$:

$$\Sigma^{tr}_{s,g\leftarrow g'} = \Sigma_{s,g\leftarrow g'} - \bar{\mu}_g \Sigma_{s,g} \tag{2.23}$$

where $\Sigma_{s,g}$ is the total scattering cross section for group g:

$$\Sigma_{s,g} = \sum_{g'}^{G} \Sigma_{s,g\leftarrow g'} \tag{2.24}$$

and the tr superscript again denotes that the cross section is transport corrected. Accordingly, the transport corrected total scattering cross section for scattering out to any other group is

$$\Sigma^{tr}_{s,g} = \Sigma_{s,g} - \bar{\mu}_g \Sigma_{s,g} \tag{2.25}$$

Lastly, the transport corrected total cross section is:

$$\Sigma^{tr}_{t,g} = \Sigma_a + \Sigma^{tr}_s \tag{2.26}$$

What is often omitted from the literature is that in a multigroup formulation the scattering kernel $\Sigma_{s,g\leftarrow g'}$ must be modified for the transport correction. If only the total scattering cross section $\Sigma_{s,g}$ and the total cross section $\Sigma_{t,g}$ (both being vectors) are transport corrected, then the cross sections are inconsistent. In the present work, we have only modified the kernel and have let that correction propagate through to the other cross sections.

In the remaining sections, the tr superscript is removed from the notation, but it is to be assumed that the total cross section is always transport corrected. In particular, it is used in the computation of the mean free path of neutrons in Section 2.5.5, and in the self-shielding of cross sections in (Equation 3.12).

The derivation of the transport correction in this work relies heavily on (Stamm'ler and Abbate, 1983), which is out of print.

2.4 Space: The Integral Transport Equation

In discretizing Equation 2.1 into macroscopic regions in space, we develop a so-called integral transport equation because the continuous functions in Equation 2.1 are integrated over finite cells in space. This allows us to work with quantities that are no longer functions of space. This will allow us to solve for the flux at different energies and in different cells in space. This, of course, is our goal in this work.

The present formulation is for any number of uniform cells in space. The only assumption we make is that the flux is constant within each of these cells. This is the so-called flat flux approximation (FFA) (Bell and Glasstone, 1970; Stamm'ler and Abbate, 1983). It is valid at energies away from resonances. Additionally, we only consider infinite geometries, so that neutrons cannot leak from the system. We currently do not need to know any more about the geometry than this. In Section 2.5, our implementation of the integral transport equation will require us to further restrict the geometry that can be considered.

We consider a system of C uniform cells. Cell i has volume V^i and surface area S^i. In a two-dimensional problem, the volume V^i reduces to the cross sectional area of the cell. The cross sections in cell i are constant with space so that $\Sigma_g^i(\mathbf{r}) = \Sigma_g^i$. By the flat flux approximation, the flux in cell i is given by Equation 2.27.

$$\phi_g^i = \frac{1}{V^i}\int_{V^i} dV \phi_g(\mathbf{r}) = \text{const.} \tag{2.27}$$

Similarly to how we have approached the transport equation previously, we integrate the multigroup transport equation over the volume V^i of a cell.

Streaming

This vanishes because the flux is constant within a cell.

Removal

$$\int_{V^i} dV \Sigma_{tg}(\mathbf{r})\phi_g(\mathbf{r}) = V^i \Sigma_{tg}^j \phi_g^i$$

Scattering

$$\int_{V^i} dV \sum_{g'}^{G} \Sigma_{s,g\leftarrow g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) = V^i \Sigma_{s,g\leftarrow g'}^i \phi_{g'}^i$$

Fission

$$\int_{V^i} dV \chi_g \sum_{g'}^{G} \nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) = V^i \chi_g^i \nu\Sigma_{fg'}^i \phi_g^i$$

We cannot simply write the transport equation by setting the removal equal to the scattering and fission sources. This is because the neutrons that are born into cell i and group g do not necessarily undergo their next collision (removal) in cell i. It is possible that neutrons that are removed from cell i had their last collision in some other cell j. This behavior provides the spatial dependence of the flux. We may express the conservation of neutrons in the system through the following:

$$\begin{bmatrix}\text{removal collisions}\\ \text{in cell } i\end{bmatrix} = \sum_{j}^{C} \begin{bmatrix}\text{probability of being}\\ \text{born in cell } j\\ \text{and having the next}\\ \text{collision in cell } i\end{bmatrix} \begin{bmatrix}\text{sources in}\\ \text{cell } j\end{bmatrix} \tag{2.28}$$

This states that all of the collisions that occur in cell i are executed by neutrons that are born in any other cell, including i. Importantly, this equation implies the system is of infinite extent. This is because there is no sink term on the left hand side that accounts for the leakage of neutrons from the system.

In order to express Equation 2.28 as an equation that we can solve, we define the term $\Pi^{i\leftarrow j}$ to be the probability that a neutron born in cell j has its next collision in cell i. These probabilities are dependent on cross sections, and accordingly the probabilities may be different for each energy group. Using $\Pi^{i\leftarrow j}$ we obtain the following for the integral transport equation:

$$V^i \Sigma^i_{tg} \phi^i_g = \sum_{j}^{C} \Pi^{i\leftarrow j}_g V^j S^j_g \tag{2.29}$$

where V^i is the volume of cell i, and both sides of the equation have units of {#/s}. The source term S^j_g is now defined for each cell j and represents the source of neutrons starting in cell j and having a collision in group g' that causes the introduction of neutrons into group g.

$$S_g^j = \sum_{g'}^{G} \left(\Sigma^j_{s,g\leftarrow g'} \phi^j_g + \chi_g \nu \Sigma^j_{f\, g'} \right) \phi^j_{g'} \tag{2.30}$$

It is this term that couples the equations in energy. Some cells do not contain fissionable isotopes, such as moderator or control rod cells. For these cells, the fission cross section is zero and the source term consists only of scattering from other groups.

Equations 2.29 and 2.30 constitute the integral transport equation using collision probabilities. It consists only of discrete quantities. The task in Section 2.5 is to find expressions for those collision probabilities in terms of known quantities.

We have $C \times G$ algebraic equations that are coupled through their energy dependence and now their spatial dependence. These equations will be amenable to solution through iteration or through an eigenvalue solution. It is useful to visualize these $C \times G$ equations as a matrix.

2.4.1 Matrix Representation

Equation 2.31 presents the integral transport equation, Equation 2.29, as a matrix for a single group g. Thus, there are G many matrices like this that must be solved.

$$\begin{bmatrix} V^1\Sigma^1_{tg}\phi^1_g \\ V^2\Sigma^2_{tg}\phi^2_g \\ \vdots \\ V^i\Sigma^i_{tg}\phi^i_g \\ \vdots \\ V^C\Sigma^C_{tg}\phi^C_g \end{bmatrix} = \begin{bmatrix} & & \\ & \Pi^{i\leftarrow j} & \\ & & \end{bmatrix} \begin{bmatrix} V^1\left(\sum_{g'}\Sigma^1_{s,g\leftarrow g'}\phi^1_{g'} + \frac{\chi_g}{k_\infty}\sum_{g'}\nu\Sigma_{fg'}\phi^1_{g'}\right) \\ V^2\left(\sum_{g'}\Sigma^2_{s,g\leftarrow g'}\phi^2_{g'} + \frac{\chi_g}{k_\infty}\sum_{g'}\nu\Sigma_{fg'}\phi^2_{g'}\right) \\ \vdots \\ V^j\left(\sum_{g'}\Sigma^j_{s,g\leftarrow g'}\phi^j_{g'} + \frac{\chi_g}{k_\infty}\sum_{g'}\nu\Sigma_{fg'}\phi^j_{g'}\right) \\ \vdots \\ V^C\left(\sum_{g'}\Sigma^C_{s,g\leftarrow g'}\phi^C_{g'} + \frac{\chi_g}{k_\infty}\sum_{g'}\nu\Sigma_{fg'}\phi^C_{g'}\right) \end{bmatrix} \tag{2.31}$$

The left hand side represents sinks in cell i, and the right hand side provides sources in cell j. The element $\Pi(i,j)$ in the $\mathbf{\Pi}$ matrix maps the neutron sources in cell j to the sink in cell i. It is this analog to matrix notation that has led us to use left-pointing arrows in our function arguments, superscripts and subscripts. As already mentioned, this equation is amenable to an eigenvalue solution if the quantities are rearranged into the following form:

$$\mathbf{R}\phi = \mathbf{S}\phi + \frac{1}{k_\infty}\mathbf{F}\phi \tag{2.32}$$

Here, **R**, **S**, and **F** are square matrices with side $C \times G$ and they respectively represent removal, scattering, and fission. Accordingly, the unknown vector ϕ has length $C \times G$. This matrix equation is further rearranged:

$$(\mathbf{R}-\mathbf{S})\phi = \frac{1}{k_\infty}\mathbf{F}\phi \tag{2.33}$$

In this form, it is evident that $1/k_\infty$ is the eigenvalue.

To solve for the flux, three types of cross sections are needed: the scattering kernel, the total cross section, and the $\nu\Sigma_f$ cross section. The scattering kernel includes inelastic scattering, and the total cross section is the sum of the elastic, inelastic, capture, and fission cross sections. The consideration of the (n,2n) and (n,3n) interactions can be implemented easily by the method used in WIMS (Leszczynski et al., 2007) (see Chapter B for its implementation in this work). These are assumed to be known, and are processed by the method discussed in Chapter 3. However, the $\mathbf{\Pi}$ matrix is as of yet unknown. Obtaining $\mathbf{\Pi}$ is the task in the next section.

2.5 Collision Probabilities

Logically, the derivation of collision probabilities, that is, $\mathbf{\Pi}$, requires us to specify something about the system's geometry. We present the geometry we intend to model, and then proceed to derive the collision probabilities for this geometry.

A number of methods exist for deriving these collision probabilities. For certain cylindrical systems, Carlvik provided an exact solution for multiple annular regions that employs Bickley functions (Stamm'ler and Abbate, 1983). Our method relies on approximations of escape and transmission probabilities.

This method was developed in recent years by Dr. Bingham Cady, Dr. Geoff Recktenwald, and Dr. Mark Deinert. The work is informed by the method of collision probabilities presented by Stacey (2001) and Stamm'ler and Abbate

(1983). The method extends the collision probability method used in the VBUDS mentioned in the introduction to a multi-region reactor core.

2.5.1 Infinite Lattice Geometry

The geometry is a regular infinite lattice of pins, denoted P, and "antipin" regions surrounding the pins, denoted A. The antipin regions are square with a circular pin cut out of the middle. The pin usually contains fuel and the antipin contains coolant or moderator. For the sake of generality, we do not restrict ourselves to this mapping of materials to cell shapes. Note that the use of the term "antipin" is not conventional; it is typical to simply refer to the pin as the fuel cell and the antipin as the coolant or moderator cell. If cell i is a pin cell, its diameter is d_i. If cell i is an antipin region, it has pitch p_i and must be associated with with a cell ip whose diameter is d_{ip}.

The cells are numbered 1 through C. By convention, the antipin cells come first. This ordering is important, given the arrangement of terms in the $\mathbf{\Pi}$ matrix. As an example, the system may consist of both UO_2 and IMF (inert matrix fuel) pins of different diameters in a water moderator. Each fuel cell must be surrounded by its own moderator. Cell 1 may be the moderator surrounding UO_2, cell 2 the moderator surrounding IMF, cell 3 the UO_2 pin, and cell 4 the IMF pin.

It is necessary to quantify the relative amounts of the two types of cells. This is done with the fractions α_i for the antipin cells and β_i for the pin cells. These two quantities must have the property

$$\sum_{i \in A} \alpha_i = 1 \qquad \sum_{i \in P} \beta_i = 1 \tag{2.34}$$

The fractions may be given by the volume fraction of a cell among cells of the same shape, or simply by the number of cells of a given shape divided by the total number of cells of the same shape. These fractions will be used shortly in the calculation of transmission probabilities.

In the present work, we only consider infinite multi-pin-cell geometries as described here. However, the only aspect of the formulation that depends on the specific geometry of the cells is the rational approximation of transmission probabilities (Section 2.5.5).

2.5.2 Escape and Transmission

We introduce two new quantities, the escape and transmission probabilities, that we will use to define the transport probabilities $\Pi^{i \leftarrow j}$. Through these two probabilities, the transport probabilities are ultimately a function of both the geometry of the system and of the total (transport corrected) cross section of the cell.

In this procedure, we consider only one energy group. As a result, in-scattering and fission are identical from the point of view of the group. Thus, we use the term “source” to mean that a neutron has been born in a given cell by either scattering or fission.

The escape probability P^i is the probability that a neutron that had its last interaction in cell i leaves the cell before interacting in it through scattering or absorption.

The transmission probability T^i is the probability that a neutron that enters cell i leaves the cell before interacting in it through scattering or absorption.

The two probabilities differ solely by how the neutron was introduced to cell i. The escape probability is used at neutron birth, while the transmission probability is used to describe the neutron's behavior after its birth. Since neutrons only have two options in a given cell, to interact or to not interact, we can mention two related probabilities. The probability that a neutron that enters cell i interacts next in the same cell i is:

$$1 - T^i \tag{2.35}$$

and is called the cell's blackness by Stamm'ler and Abbate (1983). The non-escape probability is the chance that a neutron born in cell i has its next interaction in the same cell i before escaping:

$$1 - P^i \tag{2.36}$$

In general, T^i and P^i are functions of neutron energy.

We develop explicit expressions for the escape probability P^i and the transmission probability T^i in Sections 2.5.5 and 2.5.6, respectively. For now, we only seek to employ these quantities to define the transport probabilities, $\Pi^{i \leftarrow j}$.

We define an average antipin transmission probability T^A and an average pin transmission probability T^P through the following:

$$T^A = \sum_{i \in A} \alpha_i T^i \qquad\qquad T^P = \sum_{i \in P} \beta_i T^i \tag{2.37}$$

Given that we have divided the cells into antipin cells and pin cells, it is helpful to represent $\mathbf{\Pi}$ as a block matrix.

$$\mathbf{\Pi} = \begin{bmatrix} \mathbf{\Pi}^{A \leftarrow A} & \mathbf{\Pi}^{A \leftarrow P} \\ \mathbf{\Pi}^{P \leftarrow A} & \mathbf{\Pi}^{P \leftarrow P} \end{bmatrix} \tag{2.38}$$

The upper left block represents transport between antipin cells, and the lower right cell represents transport between pin cells. The upper right block represents transport from the pin cells to antipin cells, and the lower left represents transport from antipin cells to pin cells. The columns of the matrix must sum to one, since each source particle must be present somewhere in the system.

The form of a term $\Pi^{i \leftarrow j}$ depends on its location in the matrix. We address the diagonal elements first.

2.5.3 Self-Collision

Let us first consider the probability that a neutron born in an antipin cell $i \in A$ has its next collision in the same cell i, $\Pi^{i \leftarrow i}$. Such neutrons can meet this fate in cell i in a number of ways. The simplest case is that after being born, the neutron collides in the cell before leaving it. This is the non-escape probability $1 - P^i$.

Alternatively, after birth the neutron may leave cell i (P^i), pass through the average pin cell once (T^P), then enter cell i again to meet its demise ($(1 - T^A)\alpha_i$). This last expression is the probability that the neutron does not transmit through the average antipin cell, weighted by the chance that the antipin cell in which it collides is cell i. Combining all these terms yields:

$$\begin{bmatrix}\text{probability of being born in cell } i \text{ and} \\ \text{colliding in cell } i \text{ after 1 transmission}\end{bmatrix} = P^i T^P (1 - T^A)\alpha_i \qquad (2.39)$$

This expression is for antipin cells only. The neutron can in fact pass through other cells any number of times before ultimately colliding in cell i where it was born. To account for this, we can add to the previous probability the probability of transmitting through the average antipin cell T^P and then the average pin cell again T^A before finally interacting in cell i. We consider n such pairs of transmissions.

$$\begin{bmatrix}\text{probability of being born in cell } i \text{ and} \\ \text{colliding in cell } i \text{ after } n \text{ transmission}\end{bmatrix} = P^i T^P (T^A T^P)^n (1 - T^A)\alpha_i \qquad (2.40)$$

The quantities are displayed in order of occurrence from left to right. The quantity $\Pi^{i \leftarrow i}$ is given by the sum of all of these probabilities.

$$\Pi^{i \leftarrow i} = (1 - P^i) + \sum_n^{\infty} P^i T^P (T^A T^P)^n (1 - T^A) \alpha_i \qquad i \in A \qquad (2.41)$$

Since $T^A T^P$ is necessarily less than 1, the sum in this equation describes a converging geometric series and we can express $\Pi^{i \leftarrow j}$ in its final form for antipin cells:

$$\Pi^{i \leftarrow i} = (1 - P^i) + P^i T^P \tau^A \alpha_i \qquad i \in A \qquad (2.42)$$

where we define τ^A to be the probability that a neutron's path ends in an antipin.

$$\tau^A = \frac{1 - T^A}{1 - T^A T^P} \qquad \tau^P = \frac{1 - T^P}{1 - T^A T^P} \qquad (2.43)$$

Analogously, τ^P is the probability that a neutron's path ends in a pin. The derivation for pin cells is similar, and yields the following:

$$\Pi^{i \leftarrow i} = (1 - P^i) + P^i T^A \tau^P \beta_i \qquad i \in P \qquad (2.44)$$

This infinite sum in the transport probabilities, resulting from the infinite lattice geometry, is described in Bell and Glasstone (1970). The novelty of the method presented here is the inclusion of the fractions α_i and β_i, and the average transmission probabilities $\bar{T}^A$ and $\bar{T}^P$ to extend the two-cell formulation into a multi-cell formulation. Note that in order for the average transmission probabilities to be accurate the cells must be distributed in a homogeneous manner.

The infinite lattice can alternatively be realized through the invocation of a reflective or white boundary condition at the cell surface, as is done in the multi-region method (for annular rings) by Stamm'ler and Abbate (1983) and Ozgener and Ozgener (2005).

2.5.4 Remaining Elements of the Transport Matrix

The remaining elements of the **Π** matrix are easier to obtain now that we have obtained the diagonal elements. The probability of transporting from antipin cell j to a different antipin cell i is the probability of escaping cell i after birth (P^i), transmitting through the average pin cell (T^P), and transmitting through other cells any number of times before not transmitting through cell i, ($\tau^A \alpha_i$).

$$\Pi^{i \leftarrow j} = P^j T^P \tau^A \alpha^i \qquad\qquad i, j \in A \tag{2.45}$$

The probability of transporting from pin cell j to a different pin cell i is obtained similarly.

$$\Pi^{i \leftarrow j} = P^j T^A \tau^P \beta^i \qquad\qquad i, j \in P \tag{2.46}$$

The probability of transporting from antipin cell j to pin cell i does not include T^P, since the neutron no longer needs to transmit through the first pin it enters; it may simply collide there. Logically, this probability depends on the fraction β_i for the cell it is colliding in, as well as on τ^P.

$$\Pi^{i \leftarrow j} = P^j \tau^P \beta_i \qquad\qquad i \in P, j \in A \tag{2.47}$$

The probability of transporting from pin cell j to antipin cell i is obtained similarly.

$$\Pi^{i \leftarrow j} = P^j \tau^A \alpha_i \qquad\qquad i \in A, j \in P \tag{2.48}$$

We have addressed each type of element in the transport probability matrix **Π**. The four blocks of this matrix for 2 antipin cells and 2 pin cells is provided.

$$\mathbf{\Pi}^{A\leftarrow A} = \begin{bmatrix} (1-P^1)+ \; P^1 T^P \tau^A \alpha_1 & P^2 T^P \tau^A \alpha_1 \\ P^1 T^P \tau^A \alpha_2 & (1-P^2)+ \; P^2 T^P \tau^A \alpha_2 \end{bmatrix} \tag{2.49}$$

$$\mathbf{\Pi}^{P\leftarrow P} = \begin{bmatrix} (1-P^3)+ \; P^3 T^A \tau^P \beta_3 & P^4 T^A \tau^P \beta_3 \\ P^3 T^A \tau^P \beta_4 & (1-P^4)+ \; P^4 T^A \tau^P \beta_4 \end{bmatrix} \tag{2.50}$$

$$\mathbf{\Pi}^{P\leftarrow A} = \begin{bmatrix} P^1 \tau^P \beta_3 & P^2 \tau^P \beta_3 \\ P^1 \tau^P \beta_4 & P^2 \tau^P \beta_4 \end{bmatrix} \tag{2.51}$$

$$\mathbf{\Pi}^{A\leftarrow P} = \begin{bmatrix} P^3 \tau^A \alpha_1 & P^4 \tau^A \alpha_1 \\ P^3 \tau^A \alpha_2 & P^4 \tau^A \alpha_2 \end{bmatrix} \tag{2.52}$$

Note that all elements in column j contain the quantity P^j and all elements in row i contain the quantity α^i or β^i, whichever is appropriate. The former represents escape from j and the latter represents blackness in i.

2.5.5 Rational Approximations

Having obtained all the transport probabilities in terms of the escape and transmission probabilities, the task becomes to obtain explicit expressions for these two probabilities. Many expressions are available for these quantities. The expressions that one uses may only work for a certain cell geometry. We are concerned only with circular (cylindrical) pin and antipin cells. The derivation of transmission and escape probabilities for other geometries, though possible, is not considered here. All quantities in this section are properties of cell i, but this notation has been removed for clarity.

We consider the escape probability P for pin cells first. The escape probability is a function of the geometry and the composition (specifically the mean free path λ of the neutrons) of the cell. An exact solution for any geometry can be obtained through integration of an expression describing exponential attenuation of the source in the cell. This integration is always performed assuming an isotropic and uniform source, and so it is important that the transport correction to the scattering cross section is used so that scattering can be treated as isotropic (Section 2.2.1).

It is possible to avoid the integration and instead use rational approximations to the escape probability for potentially arbitrary geometries. In such cases the dependence on geometry is primarily through the mean chord length $\bar{l}$,

$$\bar{l} = \frac{4V}{S} \tag{2.53}$$

The simplest rational approximations are for the two limiting cases: for a very thin cell, the escape probability is 1, and for a cell whose dimensions are much larger than the neutron mean free path, it is given by

$$\lim_{r\to\infty} P = \frac{A}{4V}\lambda = \frac{\lambda}{\bar{l}} \tag{2.54}$$

where r represents some characteristic dimension of the cell. It is through Equation 2.54 that the mean chord length enters into this topic. The most ubiquitous of the rational approximations is Wigner's rational approximation:

$$P = \frac{1}{1+\bar{l}/\lambda} \tag{2.55}$$

This approximation is a balance between the two limiting cases just described (Bell and Glasstone, 1970). Monte Carlo simulations show that the unitless parameter $R = \bar{l}/\lambda$ can be treated as a lumped parameter so that these approximations are a function of only R (Stacey, 2001). Bell introduced a correction factor a to Wigner's approximation:

$$P = \frac{1}{1 + R/a} \tag{2.56}$$

Carlvik proposed a two-term approximation(Stamm'ler and Abbate, 1983):

$$P = 1 - R\left(\frac{b_1}{R + a_1} + \frac{b_2}{R + a_2}\right) \tag{2.57}$$

For cylinders, $a_1 = 2$, $a_2 = 3$, $b_1 = 2$, $b_2 = -1$. Others have used similar approximations with more than just two terms (Hébert and Marleau, 1991). Lastly we present an approximation given in Stacey (2001) that we refer to as the Sauer approximation.

$$P = \frac{1}{R}\left(1 - \frac{1}{(1 + R/c)^c}\right) \tag{2.58}$$

The parameter c depends on the geometry. It is this approximation that we use in the present work, with $c = 5.00$ for pin cells. Figure 2.2 compares some of these approximations to each other. A log scale is used for the x axis because the parameter R varies with cross section, which we always consider on a log scale.

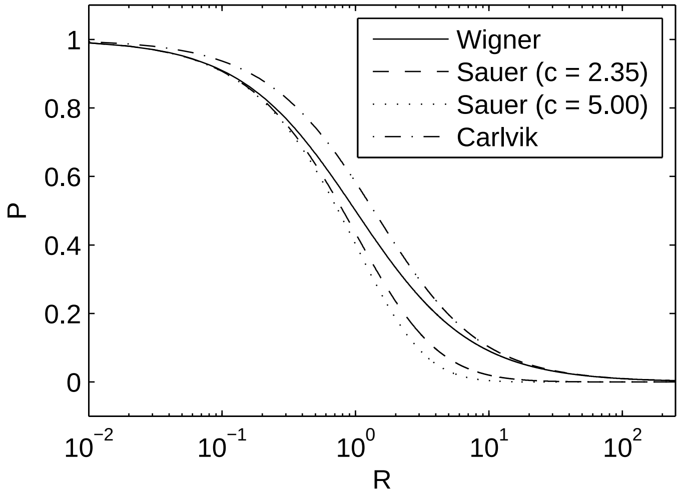

Figure 2.2: Comparison of the rational approximations to the escape probability as a function of R. In this work, Sauer's approximation is used.

For the antipin cells, the transmission probability is also called the Dancoff factor. An exact solution is not quite as easy to obtain, and so it is often tabulated through Monte Carlo methods. Schneider (2002) and Barratt (2003) discuss such methods in detail. In our work, we simply employ the Sauer approximation but with a smaller value for c, $c = 2.35$. The mean chord length for pins and antipin cells is provided:

$$\bar{l} = \begin{cases} d & \text{pin} \\ d\left(\dfrac{4p^2}{\pi d^2} - 1\right) & \text{antipin} \end{cases}$$

where d is the diameter and p is the pitch associated with the cell. The mean free path used in this section is computed from the total cross section, $\lambda = 1/\Sigma_t$, after the transport correction has been applied to the scattering cross section (Section 2.2.1). As such, the escape probability and thus the transport probabilities depend on energy group.

2.5.6 Reciprocity and the Transmission Probability

All that remains for calculating the transport probabilities is obtaining an expression for the transmission probability. This is obtained via the reciprocity relationship, which provides (for any cell):

$$1 - T = RP \tag{2.59}$$

This relation arises from equations that relate the flux at some point B due to a source at some point D to the flux at D due to a source at B. The relation is valid for arbitrary geometries, and a thorough derivation is given in Bell and Glasstone (1970). This specific form of the reciprocity relation assumes that neutrons are born isotropically. It is derived using one-group transport, and so both in-scattering and fission are treated equivalently as sources. This means that scattering must be treated isotropic. This again necessitates the use of the transport correction to the scattering cross section when computing the mean free path in R (Section 2.2.1).

As a result of reciprocity, the transmission probability is given by the following when Sauer's approximation is used for the escape probability:

$$T = \frac{1}{(1 + R/c)^c} \tag{2.60}$$

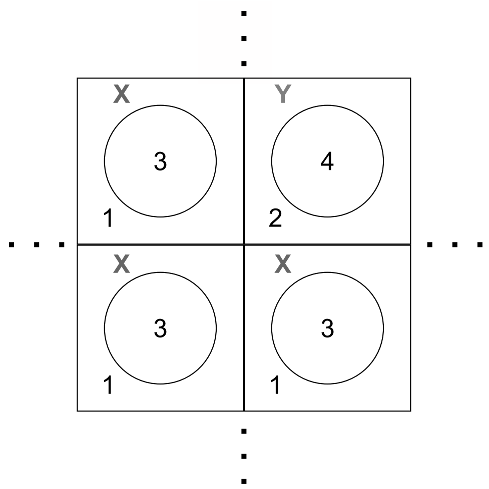

Figure 2.3: The geometry for which the homogeneity correction is derived. It consists of two antipin-pin cell pairs X and Y that extend infinitely. The cell pairs can be present in a 1:1 or a 3:1 ratio.

2.6 The Homogeneity Correction

As will be shown in Chapter 4, the method outlined in this chapter yields excellent results for two-cell systems. Chapter 4 also shows that the model is not quite as successful for four-cell systems. In this section we propose an extension to our method to account for this discrepancy. Part of the error in the four-cell results arise from the definition of the average transmission probabilities.

In the remainder of this section we restrict ourselves to a system with two antipin cells numbered 1 and 2 and two pin cells numbered 3 and 4. The cells are paired into pairs X and Y, so that pair X consists of cells 1 and 3 and pair Y consists of cells 2 and 4. This geometry is given in Figure 2.3. While it is possible to extend this method to more general geometries, we do not do so here.

The average pin transmission probability, defined in Equation 2.37 is:

$$T^P = \frac{\beta_3}{\left(1+\bar{l}\Sigma_t^3/c\right)^c} + \frac{\beta_4}{\left(1+\bar{l}\Sigma_t^4/c\right)^c} \tag{2.61}$$

where $\Sigma_t^i = 1/\lambda^i$. The term T^P appears in $\mathbf{\Pi}$ for the antipin terms in *both* pairs, X and Y. This means that the transport in one pair is affected by the cross sections in the other pair. Thus, a resonance in Σ_t^4 appears to remove neutrons from cell 1, even though it is from the other pair. While it is realistic that a resonance in one discrete region causes a flux depression in other regions, the use of Equation 2.61 overestimates this effect. Simply, the expression assumes a level of homogeneity that may not be accurate.

The current method can be extended by considering more carefully when the system appears homogeneous to neutrons. The system can be well-approximated as homogeneous when the flux is the same in different cells, or when the mean free path of neutrons is greater than the largest dimension of any uniform cell. This assumption is often valid at high energies, but is not valid in water-moderated systems at thermal energies where the high scattering cross section of water provides a very short mean free path for neutrons. Thus, the formulation of $\mathbf{\Pi}$ provided in Section 2.5 is only valid at energies and in cells where the mean free path of neutrons is larger than the cell dimensions or some other characteristic dimension. In the case that the mean free path is less than cell dimensions, we expect that neutrons will not leave their cell pair. Thus, any terms that transport neutrons from one cell pair to the other, such as $\Pi^{2\leftarrow 1}$ or $\Pi^{3\leftarrow 4}$ can be neglected.

We now refer to the $\mathbf{\Pi}$ we originally formulated in Equations 2.49 through 2.52 as $\mathbf{\Pi}_O$, as it is only valid for seemingly homogeneous systems. We define a new matrix $\mathbf{\Pi}_E$ that is expected to be valid only for seemingly heterogeneous systems. Then, the $\mathbf{\Pi}$ used in the transport of neutrons is given by

$$\mathbf{\Pi} = h\mathbf{\Pi}_O + (1-h)\mathbf{\Pi}_E \tag{2.62}$$

where $h = h(\lambda)$ is a function of mean free path called the *homogeneity*. Its key property is that its value is close to 1 when the system appears mostly homogeneous and is close to 0 when the system appears mostly heterogeneous. The intent with Equation 2.62 is that at energies and in cells where the system is seemingly homogeneous, $\mathbf{\Pi}_O$ dominates the transport, and vice versa. This new definition of $\mathbf{\Pi}$ will still preserve the property that the columns sum to 1.

It is important to note that since h is dependent on mean free path, it varies strongly with neutron energy, and a different value of h is given to each group. We choose to model h with the error function, as this provides a smooth transition between the homogeneous and heterogeneous cases. Figure 2.4 shows the homogeneity function as it varies with the mean free path. The value of $\lambda = \lambda^h$ is chosen to be a characteristic dimension of the system above which the system is seemingly homogeneous and below which the system is seemingly heterogeneous. Additionally, $h(\lambda)$ can be parameterized by the slope at λ^h to change how gradually the homogeneous effects are replaced with heterogeneous effects. A possible form of the homogeneity function is:

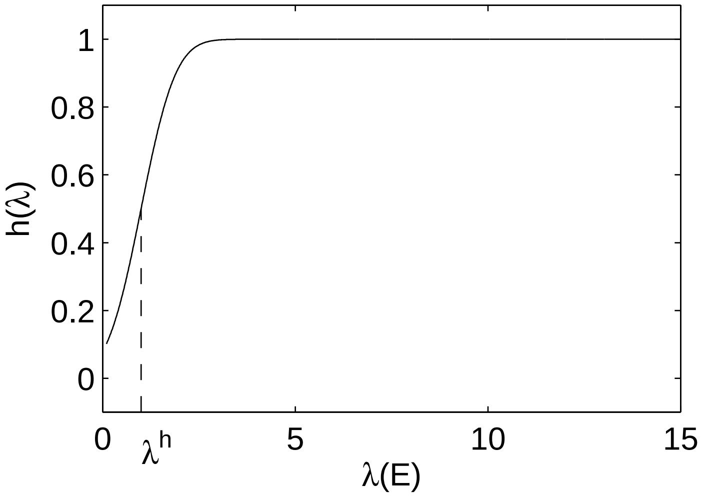

Figure 2.4: The homogeneity function $h(E)$ is defined using the error function shifted so that its inflection point occurs at λ^h, a threshold mean free path. This threshold is set to a characteristic dimension of the system.

$$h(\lambda) = \frac{1}{2}\left(1 + \mathrm{erf}[k(\lambda - \lambda^h)]\right) \tag{2.63}$$

where k is the parameter that controls the slope at $\lambda = \lambda^h$.

The value of λ^h for a given group is obtained by a volume-weighted average of the mean free path in the antipin of each cell pair.

$$\lambda_g = \alpha_1 \lambda_g^1 + \alpha_2 \lambda_g^2 \tag{2.64}$$

$$= \frac{\alpha_1}{\Sigma_g^1} + \frac{\alpha_2}{\Sigma_g^2} \tag{2.65}$$

The cross sections appearing in Equation 2.65 are known. Now that h is completely defined, we turn turn to defining Π_E. All terms that represent transport between cell pairs vanish. Furthermore, the average transmission probabilities across cells of the same type, T^A and T^P, reduce to the transmission probabilities T^i given directly by Equation 2.60. The terms τ^A and τ^P, which depended on T^A and T^P, are correspondingly redefined in terms of T^i.

For the four-cell geometry this yields:

$$\mathbf{\Pi}_E = \begin{bmatrix} Q^1 + P^1 T^3 \tau^1 & 0 & P^3 \tau^1 & \\ 0 & Q^2 + P^2 T^4 \tau^2 & 0 & P^4 \tau^2 \\ P^1 \tau^3 & 0 & Q^3 + P^3 T^1 \tau^3 & 0 \\ 0 & P^2 \tau^4 & 0 & Q^4 + P^4 T^2 \tau^4 \end{bmatrix} \tag{2.66}$$

where $Q^i = 1 - P^i$ has been used for brevity. The first and third rows and columns of Equation 2.66 describe the heterogeneous transport within cell pair X, and the second and fourth do the same for cell pair Y. Since there are no cross-pair terms, the cell pairs are uncoupled in space at energies where $h = 0$. However, the cell pairs are still coupled to each other through energies at which the homogeneous assumption and $\mathbf{\Pi}_O$ are valid. To approach an understanding of Equation 2.66, note that the T^3 appearing in the first row and column in the homogeneous case are replaced by $T^P = \beta^3 T^3 + \beta^4 T^4$ in the heterogeneous case presented here.

Chapter 3

Self-Shielded Multigroup Cross Sections

In this chapter we describe the method for obtaining multigroup cross sections that are corrected for the resonance self-shielding effect. This effect must be considered because the presence of resonances affects the integrals in Equation 2.17. This subject is an area of active research, particularly as researchers attempt to to improve simulation fidelity. We are interested in obtaining self-shielded cross sections for a heterogeneous geometry.

The traditional method for treating heterogeneity involves applying an equivalence relation to the background cross section of the Bondarenko (1964) method (Kidman et al., 1972; Stamm'ler and Abbate, 1983; Gopalakrishnan and Ganesan, 1998; Schneider et al., 2006a; Yamamoto, 2008; Joo et al., 2009). The desire to improve simulation fidelity, which can be particularly important for modeling next generation reactors with more complex geometries, has led to the use subgroup methods (Cullen, 1974; Hebert, 1997; Chiba, 2003; Huang et al., 2011). These methods, for example, are more capable of managing configurations that exhibit double heterogeneity (Powney and Newton, 2004). Self-shielding methods primarily differ in the accuracy with which they attempt to approximate the neutron flux within a group. The subgroup method is beyond the scope of this work. An overview of the different self-shielding methods is provided in the review papers by Hwang (1982) and Hébert (2007).

3.1 Resonance Self-Shielding

In this section we consider the process for obtaining correct multigroup cross sections, which were introduced in Section 2.3. It was mentioned then that the calculation of multigroup cross sections requires knowledge of the unknown flux $\phi(E)$. If the flux $\phi(E)$ is roughly constant through a group then the flux drops out of Equation 2.17, and the flux is not needed to obtain group cross sections. This assumption is often valid, but cannot be used when the flux varies rapidly within a group as is the case in groups where the cross section exhibits resonances. To obtain group cross sections in these cases, the flux $\phi(E)$ must be known or approximated and the integration in Equation 2.17 must be performed for the relevant interactions. However, it is not possible to work in a spectral code with the point-wise data this integration requires, as the continuous flux is never known. It is necessary to precompute this integration with a separate data processing code. In the next section we describe the self-shielding effect that affects the shape of the continuous flux $\phi(E)$. In Sections 3.2 and 3.3 we introduce the background cross section method that allows this integration to be performed in a problem-independent manner.

Consider a homogeneous mixture of two nuclides: a moderator such as oxygen and a resonant absorber such as uranium-238 with a resonance at energy E^*. If the uranium-238 is present with a high concentration, then many neutrons will be absorbed in its resonance and the flux exhibits a sharp dip at E^*. Since the reaction rates with which group cross sections are computed depend on this flux, all group cross sections are decreased from the value they would have if the flux did not dip.

However, the presence of a resonant isotope does not ensure a dip in the flux. If other nuclides in the system are present with a concentration much greater than that of the resonant absorber, then the resonance has little effect on the flux and there is no self-shielding. We expect then that the resonance self-shielding effect depends largely on the composition of the system under consideration. We will identify a parameter σ_0 that characterizes this problem dependence.

3.2 Flux Approximation

We initially consider a flux approximation for homogeneous systems, and will extend our result for homogeneous systems to heterogeneous systems in Section 3.4 with an equivalence relation. For a homogeneous system containing M nuclides, the macroscopic total cross section is given by:

$$\Sigma_t(E) = \sum_m^M N^m \sigma^m(E) \tag{3.1}$$

where N^m is the number density of the m-th nuclide in the mixture. In this homogeneous medium, the neutron balance is given by the following (Gopalakrishnan and Ganesan, 1998):

$$\Sigma_t(E)\phi(E) = S(E) \tag{3.2}$$

where $S(E)$ is a smooth function that represents the flux profile within a group if the total cross section were constant (i.e. neglecting resonance effects). $S(E)$ can be treated as constant for small energy groups, but depends on energy if the group covers an energy range that is wide enough for the flux to rise or fall appreciably. For example, in a thermal spectrum reactor, $S(E)$ would be given by a Maxwell-Boltzmann relation plus a fission spectrum. We can use Equation 3.2 to express the flux $\phi(E)$ in terms of the macroscopic total cross section:

$$\phi(E) = \frac{S(E)}{\Sigma_t(E)} \tag{3.3}$$

This equation shows the effect mentioned earlier, that the presence of a resonance peak in $\Sigma_t(E)$ causes $\phi(E)$ to dip. Equation 3.3 can be used to approximate the behavior of the in-group flux under the narrow resonance (NR) approximation (Bell and Glasstone, 1970). We employ this flux approximation in the definition of the group cross section, Equation 2.17, to arrive at Equation 3.4:

$$\sigma_g^m = \frac{\displaystyle\int_g dE\, \sigma^m(E)\frac{S(E)}{\Sigma_t(E)}}{\displaystyle\int_g dE\, \frac{S(E)}{\Sigma_t(E)}} \tag{3.4}$$

Here, σ_g^m represents the microscopic group cross section for the m-th nuclide in the mixture. Equation 3.4 can be used to generate the appropriate group cross section for every individual reaction of interest. Note that it is the total cross section that always appears on the right hand side of Equation 3.4, as a resonance in any of the interactions that contribute to the total cross section cause a depression in the flux.

3.3 The Background Cross Section

A key contribution of Bondarenko (1964) was to separate the macroscopic total cross section into a term that depends solely on the point-wise cross section for the desired nuclide m and a term that encompasses all other isotopes in the mixture (i.e. the system of interest). We separate the macroscopic cross section (Equation 3.1) into two terms:

$$\Sigma_t = N^m \sigma_t^m + \sum_{n \neq m} N^n \sigma_t^n \tag{3.5}$$

The cross sections in Equation 3.5 are continuous functions of energy, but we have omitted the energy argument for brevity. We factor out the number density of nuclide m, for which we desire the group cross section, from both terms:

$$\begin{aligned} \Sigma_t &= N^m \left(\sigma_t^m + \frac{1}{N^m} \sum_{n \neq m} N^n \sigma_t^n \right) \\ &= N^m (\sigma_t^m + \sigma_0^m) \end{aligned} \tag{3.6}$$

where σ_0^m is called the background cross section:

$$\sigma_0^m = \frac{1}{N^m} \sum_{n \neq m} N^n \sigma_t^n \tag{3.7}$$

Combining Equations 3.4, 3.5, and 3.7, we obtain a new expression for the group cross section using the background cross section:

$$\sigma_g^m = \sigma_g^m(\sigma_0^m) = \frac{\displaystyle\int_g dE\, \sigma^m \frac{S}{\sigma_t^m + \sigma_0^m}}{\displaystyle\int_g dE\, \frac{S}{\sigma_t^m + \sigma_0^m}} \tag{3.8}$$

where σ_t^m, σ_0^m, and S are again functions of energy. Later, we will assume that the background cross section is constant within a given group. It is clear from Equation 3.8 that the dependence of the group cross section on a specific problem is contained in the background cross section. The background cross section given by Equation 3.7 sheds light on the nature of self-shielding. Importantly, the self-shielding effect vanishes as the background cross section increases.

- As the concentration of nuclide m decreases, we reach what is called infinite dilution: a dilute nuclide m does not cause a flux depression simply because there is very little of it and so few neutrons interact with it.
- As the concentration of nuclide m increases, the denominator of the flux approximation increases, causing a dip in the flux. This dip in the flux in turn decreases the value of the group cross section from its value at infinite dilution.

- As the concentration or cross section of any nuclide $n \neq m$ increases, we again approach the case of infinite dilution in which the cross section of nuclide m becomes less important.

3.4 Heterogeneous Systems and the Equivalence Relation

Equation 3.8 applies to a homogeneous mixture of nuclides. Nuclear reactors, by contrast, are typically heterogeneous systems with distinct regions for fuel, coolant and control materials. This is important because in a heterogeneous system it is not just the total cross section that affects the in-group flux. The probability that neutrons of a given energy will escape from a cell must also be taken into consideration. In order to capture this effect we modify the neutron balance in Equation 3.2 by adding an escape cross section Σ_e $\{\mathrm{cm}^{-1}\}$:

$$[\Sigma_t(E) + \Sigma_e(E)]\phi(E) = S(E) \tag{3.9}$$

Equation 3.9 constitutes a so-called ‘equivalence relation‘ because it allows us to treat the heterogeneous case identically to how we treated the homogeneous case by simply adding an effective cross section to the total cross section (Bell and Glasstone, 1970). We rearrange Equation 3.9 to obtain the flux $\phi(E)$:

$$\phi(E) = \frac{S(E)}{\Sigma_t(E) + \Sigma_e(E)} \tag{3.10}$$

We momentarily delay the task of obtaining an expression for the escape cross section Σ_e. Expanding the denominator of Equation 3.10 as we did before, we obtain:

$$\begin{aligned}
\Sigma_t + \Sigma_e &= N^m \left(\sigma_t^m + \frac{1}{N^m} \sum_{n \neq m} N^m \sigma_t^n \right) + \Sigma_e \\
&= N^m \left(\sigma_t^m + \frac{1}{N^m} \left(\sum_{n \neq m} N^m \sigma_t^n + \Sigma_e \right) \right) \qquad (3.11) \\
&= N^m (\sigma_t^m + \sigma_0^m)
\end{aligned}$$

where the background cross section is now given by:

$$\sigma_0^m = \frac{1}{N^m} \left(\sum_{n \neq m} N^m \sigma_t^n + \Sigma_e \right) \qquad (3.12)$$

This equation provides the background cross section for cell i. The escape cross section can be viewed as the cross section of an additional nuclide in a homogeneous system. Equations 3.8 and 3.12 can then be used to obtain the group cross section for any reaction, any nuclide, and in any cell of a reactor.

3.5 The Escape Cross Section

We now turn to the task of obtaining an expression for the escape cross section. To do this, we employ the use of collision probabilities (Section 2.5) that describe the flow of neutrons among an arbitrary number of cells. We observe that the following equality, which relates the collision probabilities to the total and escape cross sections, must hold true.

$$\Pi^{i\leftarrow i} = \frac{\Sigma_t}{\Sigma_t + \Sigma_e} \tag{3.13}$$

The cross sections here are defined for cell i and are functions of energy. This equation states that a neutron born in cell i has two options: to collide in cell i or to escape it. The probability of colliding in cell i by definition must be equal to $\Pi^{i\leftarrow i}$. This expression is equivalent to the Wigner rational approximation (MacFarlane and Muir, 1994), which is used commonly in the study of collision probabilities. The escape cross section of a cell is often simply approximated as the reciprocal of the average chord length $\bar{l}$ of the cell (Bell and Glasstone, 1970).

$$\Sigma_e = \frac{1}{\bar{l}} = \frac{S}{4V} \tag{3.14}$$

Here, S and V are respectively the surface area and volume of cell i. In their derivation of the more advanced neutron current method, Yamamoto (2008) provides a conventional equivalent background cross section that employs Equation 3.14 as well as a Dancoff factor D to account for lattice geometry:

$$\sigma_0 = \frac{1}{N^m}\left(\sum_{n\neq m} N^n \sigma_t^n + D\frac{S}{4V}\right) \tag{3.15}$$

Methods more advanced than the Bondarenko method relate the collision probabilities to the escape cross section by employing Carlvik's two-term rational approximation (Stamm'ler and Abbate, 1983) or other multi-term approximations (Hébert and Marleau, 1991).

Alternatively, one can use Equation 3.13 as an implicit definition of the escape cross section (MacFarlane and Muir, 1994; Nasr and Roushdy, 1991; Schneider et al., 2006a).

$$\Sigma_e = \Sigma_t \left(\frac{1 - \Pi^{i \leftarrow i}}{\Pi^{i \leftarrow i}} \right) \tag{3.16}$$

For a homogeneous system, $\Pi^{i \leftarrow i}$ is equal to 1 and the escape cross section vanishes as expected. Equation 3.16 can be used if the collision probabilities are known and a value for the escape cross section is sought. Indeed, this is our situation. Thus, we use the collision probabilities we computed in Section 2.5 to obtain the escape and background cross section. It is noted then that our procedure for obtaining correct group cross sections is tied to our method for obtaining a spectral solution.

3.6 The Self-Shielding Factor

The limit of the group cross section as the background cross section approaches infinity yields what is called the infinite dilution (or unshielded) cross section. It is common to define a self-shielding factor f as the ratio of the group cross section to the infinite dilution cross section:

$$f(\sigma_0) = \frac{\sigma_g(\sigma_0)}{\sigma_g(\sigma_0 \to \infty)} \tag{3.17}$$

Accordingly, the quantity $\sigma_g(\sigma_0) = f(\sigma_0)\sigma_g(\sigma_0 \to \infty)$ is called the self-shielded cross section. Both $\sigma_g(\sigma_0)$ and $\sigma_g(\sigma_0 \to \infty)$ are obtained through

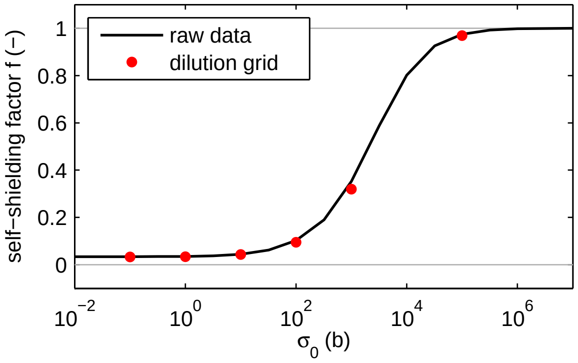

Figure 3.1: The self-shielding factor for the U-238 capture cross section at its 6.67 {eV} resonance as a function of background cross section. The red dots denote the dilution grid (see Section 3.7) used in this work for interpolating the cross sections based on background cross section.

Equation 3.8, but the latter is obtained assuming that σ_g is unaffected by the other nuclides in the mixture. Figure 3.1 shows the self-shielding factor as a function of background cross section for the uranium-238 capture cross section at 6.67 eV. The figure shows that the self-shielding factor generally falls between zero and unity, and approaches unity as the background cross section increases. The function is commonly fit to a tanh curve (Kidman et al., 1972):

$$f(\sigma_0) = A \tanh\left[B(\ln \sigma_0 + C)\right] + D \tag{3.18}$$

where the constants A, B, C, and D are fitting parameters. A method for obtaining these parameters for given self-shielding data is described by Kidman et al. (1972).

Figure 3.2 compares the point-wise U-238 capture cross section to the

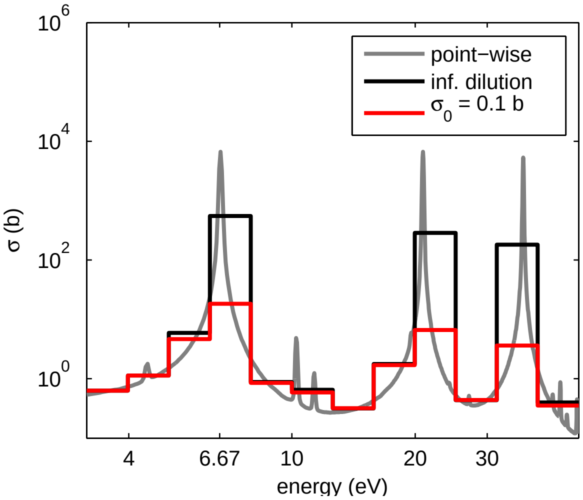

Figure 3.2: The three lowest energy resonances in the U-238 capture cross section, given as point-wise, infinite dilution, and at a background cross section of $\sigma_0 = 0.1\,\{\mathrm{b}\}$.

group cross section at infinite dilution as well as at a background cross section of $\sigma_0 = 0.1\,\{\mathrm{b}\}$. It is clear from this figure that self-shielding only has an effect at resonances, as in most bins the infinite dilution and $\sigma_0 = 0.1\,\{\mathrm{b}\}$ group cross sections have the same value.

3.7 Implementation

In this section we describe how the Bondarenko approach is implemented in a spectral code. This procedure is also shown in the flowchart of Figure 3.3. Remember that a key benefit of the Bondarenko method is that the integration over point-wise data is done with the problem dependence contained a single parameter σ_0. The generation of group cross sections is performed by a data processing code such as NJOY (MacFarlane and Muir, 1994) for select values of

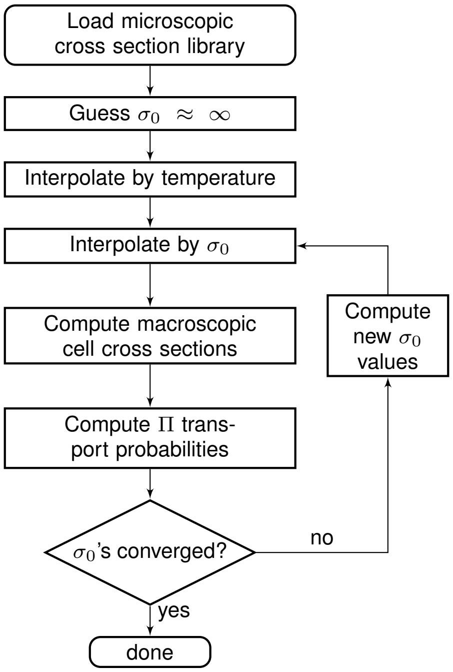
Load microscopic cross section library
Guess $\sigma_0 \approx \infty$
Interpolate by temperature
Interpolate by σ_0
Compute macroscopic cell cross sections
Compute Π transport probabilities
σ_0's converged?
no
yes
done
Compute new σ_0 values

Figure 3.3: Flowchart describing self-shielding calculations in which the escape cross section is computed directly from collision probabilities.

σ_0, and this data is stored for use in the spectral code.

Recall that the calculation of the group constants depends on a source function $S(E)$. Typically, a code such as NJOY provides a number of options for the function $S(E)$ to be used in computing the integrals. Thus, this function is managed by the data processing code, and need not be considered when computing the background cross section of a system. However, the form of $S(E)$ depends on the type of reactor being studied (fast, thermal, etc.) In our work, we have set NJOY to use $S(E) = 1/E$.

The values of σ_0 at which cross sections are obtained form what is called a dilution grid. The background cross section can vary from 0.1 to 10^{10} barns (infinite dilution), and so the dilution grid must span a large range of values. As a result, some care must be taken in choosing the proper values of σ_0 to use in the grid. We have found that the values 0.1, 1, 10.0 100.0, 10^3, 10^5, and 10^{10} (infinite dilution) capture the behavior of the self-shielding factor (Schneider, 2002). This grid is shown in Figure 3.1 as red dots. This provides us with a table, for each nuclide and each energy group, of self-shielded cross sections (or equivalently, self-shielding factors) at different values of σ_0. We can interpolate on this table using the values of σ_0 that describe our specific system, which we obtain through Equations 3.7 and 3.12. The group cross sections are a function of temperature as well, and so this interpolation table is typically two-dimensional.

Although the background cross section was derived using continuous quantities, we must use grouped cross sections instead because a spectral code only has access to group cross sections. Therefore, it is necessary to ignore any possible energy dependence of $\sigma_0(E)$ within an energy group. Fortunately, under the narrow resonance approximation this practice is expected to introduce minimal error (Gopalakrishnan and Ganesan, 1998). Note that the collision probabilities are also used as group-averaged quantities rather than as continuous functions even though they were initially presented as such.

The use of group cross sections instead of continuous energy quantities for obtaining σ_0 introduces yet another issue. The background cross section is now computed using the very same quantities that the background cross section is supposed to provide. As a result, the self-shielding procedure is iterative. It is common to initially assume that all group cross sections have their infinite dilution value. A value of σ_0 is computed from these infinite dilution group cross sections. Then, this newly-computed σ_0 is used to obtain self-shielded cross sections using the interpolation table.

Chapter 4

Results

We present results for both two-cell and four-cell systems. The method has been shown to provide excellent results in the two-cell cases. However, the four-cell configuration poses a challenge for the method.

4.1 Two-Cell Reactors

The two-cell reactors that we present are a light water reactor and a sodium-cooled fast reactor. This infinite geometry is described in Figure 4.1. First, we show the effectiveness of the self-shielding method by presenting self-shielded cross section for the uranium in both of these reactors. Subsequently, we show spectrum, reaction rate, and criticality results for each of the two two-cell configurations.

The results for both of the simulations are benchmarked against MCNPX 2.7.0, and the results of the light water simulation are additionally compared to the well-known Rowlands benchmark (Rowlands, 1999).

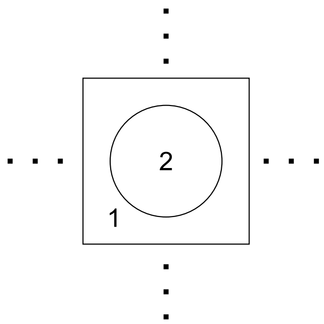

Figure 4.1: Diagram of a two-cell system. The system consists of a pin (cell 2) and a surrounding antipin cell (cell 1) that repeat as an infinite lattice.

4.1.1 A Light Water Thermal Reactor

The Rowlands benchmark is part of an OECD effort to obtain the discrepancy in results (multiplication factor, reaction rates, etc.) that arises from the processing of the same set of nuclear data by different methods across the industry (Rowlands, 1999). To this end, the benchmark specifies a number of cases of simplified reactor configurations (UOX-fueled and MOX-fueled LWRs). This benchmark is commonly used in the literature to evaluate resonance self-shielding methods (Gopalakrishnan and Ganesan, 1998; Hébert, 2005). Thus, we have chosen to model our light water reactor after case 1 of this benchmark.

Case 1 of the benchmark is provided for both a square moderator cell and a circularized moderator cell (using the Wigner-Seitz method). We implement the square cell version of case 1. This case incorporates a 0.05 {cm} thick

Table 4.1: Dimensions and composition of the LWR from case 1 of the Rowlands benchmark (Rowlands, 1999). The fuel is enriched uranium dioxide, and is surrounded by a water moderator. A zirconium fuel rod is modeled by smearing the appropriate amount of zirconium across the water.

Pin	
diameter	0.8000 cm
temperature	294.0 K
nuclide	density (#/b/cm^2)
U-235	0.0007080
U-238	0.0226040
O-16	0.0466240

Annulus	
pitch	1.2000 cm
temperature	294.0 K
nuclide	density (#/b/cm^2)
H-1	0.0574461
O-16	0.0286544
Zr-90	0.0061594

zirconium fuel rod. In our simulation we do not model the fuel rod exactly and instead we smear the same amount of zirconium across the moderator as would be contained in the fuel rod. Since our system, described in Table 4.1, is not exactly that specified by the benchmark, we also compare our results to an MCNPX simulation with the same input as with our method. We compare results for the multiplication constant and reaction rates between all three simulations, but compare the neutron spectrum only to MCNPX (a neutron spectrum is not provided in the Rowlands benchmark). We model the system using 100 energy groups from 1 {meV} to 10 {MeV}.

Table 4.2: Dimensions and composition of a sodium fast reactor. The fuel is enriched uranium dioxide and is surrounded by a sodium moderator. A steel fuel rod is modeled by smearing chromium, iron, and nickel across the sodium.

Pin	
diameter	1.3727 cm
temperature	900.0 K
nuclide	density (#/b/cm^2)
U-235	0.0062999
U-238	0.0183091
O-16	0.0492181

Annulus	
pitch	2.0000 cm
temperature	600.0 K
nuclide	density (#/b/cm^2)
Na-23	0.0210618
Cr-52	0.0016310
Fe-56	0.0053902
Ni-58	0.0007107

4.1.2 A Sodium Cooled Fast Reactor

We model an infinite lattice of a sodium-cooled fast reactor fueled with fresh uranium dioxide. A fuel pin is modeled by smearing a 0.5 {cm} thick steel rod (using only chromium, iron, and nickel) across the moderator. The dimensions and composition of this system are provided in Table 4.2. For this system, we use 43 energy groups that are equally spaced in lethargy from 400 {eV} to 10 {MeV}; the flux at energies below this range is zero.

4.1.3 Light Water Reactor Cross Sections

The cross sections are compared to their values at infinite dilution, as well as to the cross sections that MCNPX reports for a simulation of this system.

The MCNPX cross sections are obtained from the reaction rates provided by a cell tally in conjunction with the appropriate tally multiplier. The MCNPX cross sections are assumed to be self-shielded.

Figure 4.2 shows the capture cross section for uranium-238 in the energy range 1 eV to 100 keV. The top graph shows that there is substantial self-shielding of this cross section. This is expected, because uranium-238 is present in such a great concentration and many neutrons are absorbed in its resonances. The middle graph shows the error between MCNPX and our method. The bottom graph shows the background cross section for uranium-238, as well as the escape portion of this background cross section. Over the entire energy range, the background cross section has a relatively small value, as we expect in the case where self-shielding is substantial. Though the error between MCNPX and our method is large in a few energy groups, the method is mostly able to compute the escape probability that yields the correct self-shielded cross sections.

Figure 4.3 shows the capture cross section for uranium-235, which is present in a much smaller concentration than is uranium-238. As a result, the background cross section is large and the infinite dilution cross section is valid. By comparing the bottom graph of Figure 4.3 to the unshielded cross section in Figure 4.2, it is evident that the background cross section for uranium-235 is dominated by the uranium-238 capture cross section.

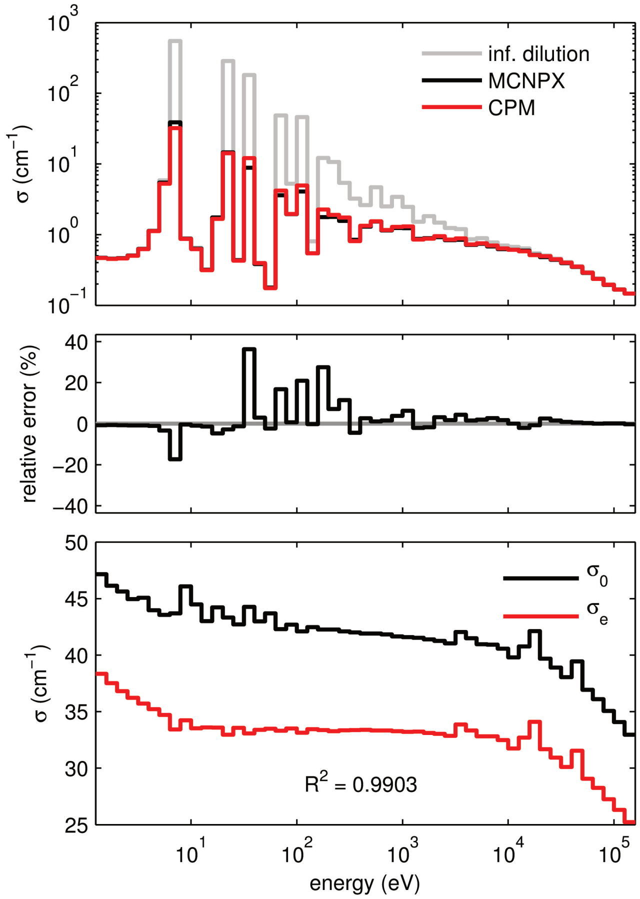

Figure 4.2: Group cross section for U-238 capture from 1 eV to 100 keV. The top graph provides the infinite dilution cross section and compares the cross section generated by our method ("CPM") to that obtained by MCNPX. The middle graph provides the error between our method and MCNPX. The bottom graph shows the background cross section that yields the self-shielded cross section shown at top, as well as the escape portion of this background cross section.

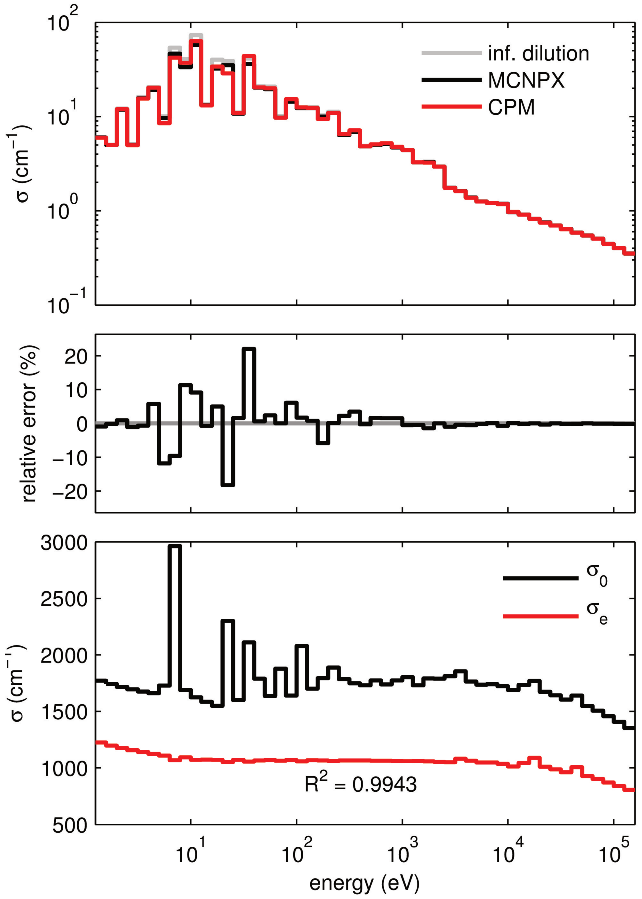

Figure 4.3: Group cross section for U-235 capture from 1 eV to 100 keV. The top graph provides the infinite dilution cross section and compares the cross section generated by our method ("CPM") to that obtained by MCNPX. The middle graph provides the error between our method and MCNPX. The bottom graph shows the background cross section that yields the self-shielded cross section shown at top, as well as the escape portion of this background cross section.

4.1.4 Fast Reactor Cross Sections

In Figure 4.4 and Figure 4.5 we again present the cross sections for capture by uranium-238 and uranium-235 but now in the fast reactor. The shape of the escape cross section is now dominated by the sodium cross section. Again, as expected the background cross section for uranium-238 is much smaller than it is for uranium-235 because it is present in a much higher concentration. As a result the uranium-238 cross section is substantially self-shielded.

4.1.5 Thermal Spectrum and Reaction Rates

The neutron spectrum obtained by our method, shown in Figure 4.6, correctly captures all essential features of the MCNPX flux, including the three resonance dips in the epithermal region. The coefficient of determination R^2 between the two results, at 0.9998, indicates a high accuracy of our method across the energy groups.

Table 4.3 compares 3-group reaction rates from our method to those given in the Rowlands benchmark. The results are normalized to 100,000 total absorptions in the system. On the other hand, Table 4.4 compares our results to those obtained by an MCNPX simulation in which the zirconium fuel rod has also been smeared. Table 4.4 indicates a high level of accuracy of our method in computing reaction rates, as all errors are below 5%. However, it is clear from Table 4.3 that smearing the zirconium throughout the moderator introduces substantial error in comparison to published Rowlands results.

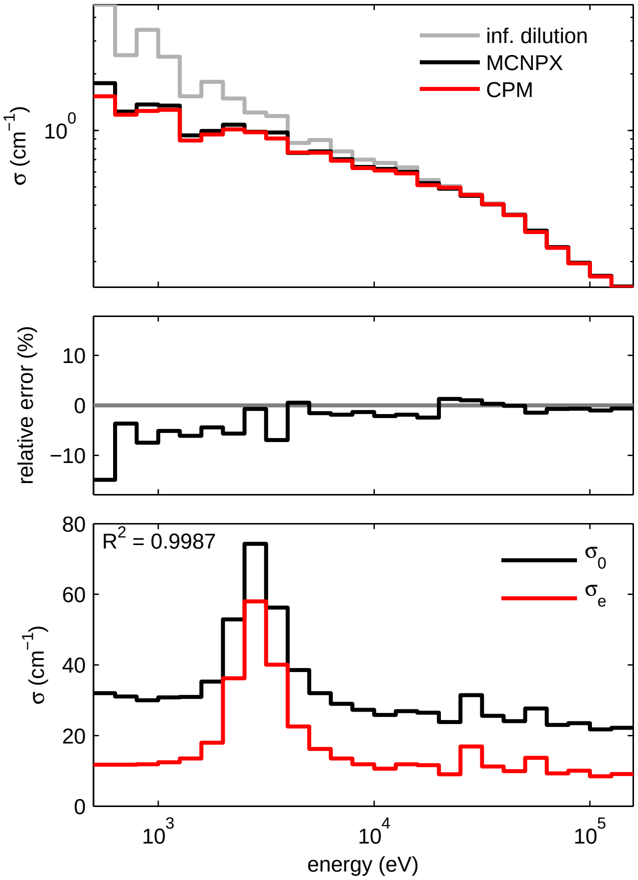

Figure 4.4: Self-shielded uranium-238 capture cross section in a sodium-cooled fast reactor. The top graph provides the infinite dilution cross section and compares the self-shielded cross section produced by our method ("CPM") to MCNPX. The middle graph presents the error between our cross section and MCNPX's. The bottom graph shows the background cross section generated by our code as well as the escape portion of this background cross section. The escape cross section is dominated by the sodium-23 resonance present in the coolant. The relatively small value for the background cross section gives rise to substantial self-shielding. The coefficient of determination between the two results is 0.9987.

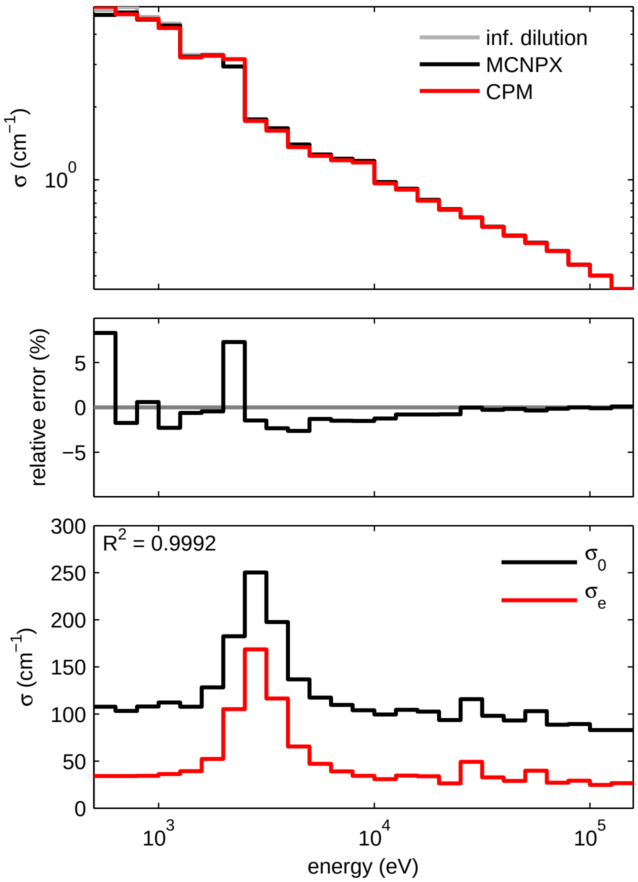

Figure 4.5: Self-shielded uranium-235 capture cross section in a sodium-cooled fast reactor. The top graph provides the infinite dilution cross section and compares the self-shielded cross section produced by our method ("CPM") to MCNPX. The middle graph presents the error between our cross section and MCNPX's. The bottom graph shows the background cross section generated by our code as well as the escape portion of this background cross section. The escape cross section is dominated by the sodium-23 resonance present in the coolant. However, uranium-235 is present in a smaller concentration than uranium-238 and so its background cross section is larger than for uranium-238. The coefficient of determination between the two results is 0.9992.

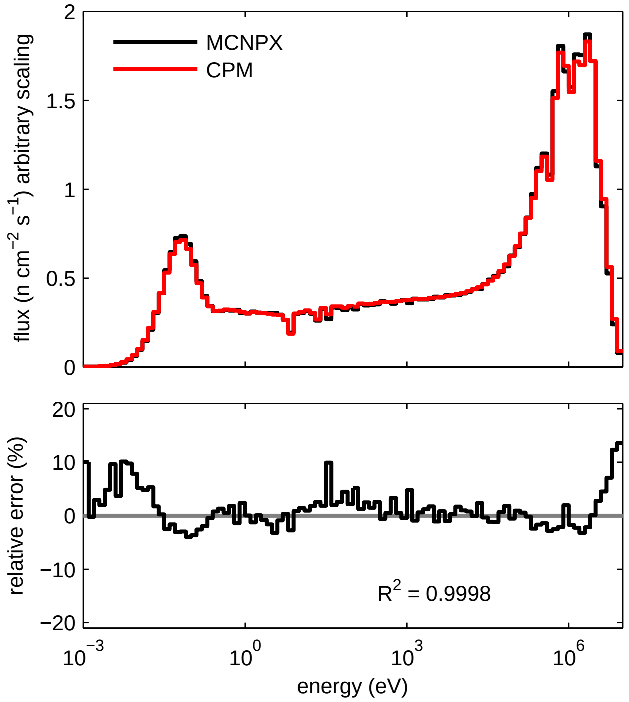

Figure 4.6: Comparison to MCNPX of spectral flux in the fuel of the LWR from case 1 of the Rowlands benchmark. The top graph compares the fuel flux from our method ("CPM") to MCNPX results. The bottom graph provides the error between our method and MCNPX. The coefficient of determination between the results is 0.9998.

Table 4.3: Comparison to Rowlands results of 3-group reaction rates in the LWR from case 1 of the Rowlands benchmark. The absorption rates for the uranium isotopes are provided. Our method is noted as "CPM". The numbers are normalized to 100,000 total absorptions in the system. The errors are slightly higher than in Table 4.4 as a result of how we have approximated the presence of the fuel rod.

nuclide	reaction	group	MCNPX	CPM	error (%)
U-235	fission	fast	728	674	-7.42
		res.	4363	4038	-7.45
		thermal	51926	49555	-4.57
	capture	fast	117	110	-5.98
		res.	2386	2365	-0.88
		thermal	9051	8795	-2.83
U-238	fission	fast	2963	2785	-6.01
		res.	1	1	0
	capture	fast	2262	2161	-4.47
		res.	16148	15303	-5.23
		thermal	8771	8420	-4.00

Table 4.4: Comparison to MCNPX of 3-group reaction rates in the LWR from case 1 of the Rowlands benchmark. The absorption rates for the uranium isotopes are provided. Our method is noted as "CPM". The numbers are normalized to 100,000 total absorptions in the system. All errors are below 5%, and are slightly smaller than in Table 4.3 because the fuel rod has been smeared in MCNPX just as it has been in our model.

nuclide	reaction	group	MCNPX	CPM	error (%)
U-235	fission	fast	675	674	-0.15
		res.	4093	4038	-1.34
		thermal	49424	49555	0.27
	capture	fast	110	110	0
		res.	2371	2365	-0.25
		thermal	8762	8795	0.38
U-238	fission	fast	2765	2785	0.72
		res.	1	1	0
	capture	fast	2171	2161	-0.46
		res.	15218	15303	0.56
		thermal	8395	8420	0.30

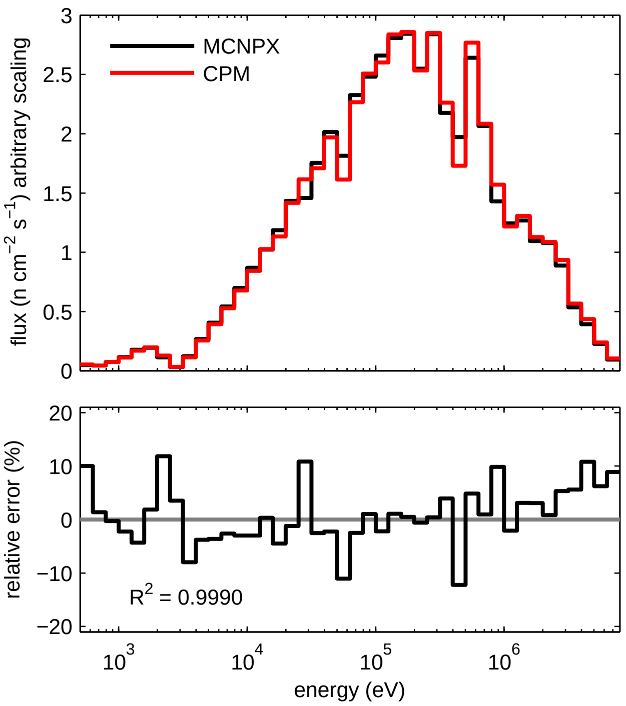

Figure 4.7: Comparison to MCNPX of spectral flux in the fuel of the sodium fast reactor in the results. The top graph compares the fuel flux from our method to MCNPX results. Our method is able to capture the flux dips corresponding to resonances. The bottom graph provides the error between our method and MCNPX. The coefficient of determination between the results is 0.9990.

4.1.6 Fast Reactor Spectrum and Reaction Rates

The neutron spectrum in the fuel is compared in Figure 4.7 to results obtained by MCNPX for the same system. The relative error between the two results is shown in the lower graph. The coefficient of determination R^2 between the two results is 0.9977, which indicates a good correlation of the results across the energy groups.

Table 4.5 compares one-group reaction rates computed by our method

Table 4.5: Comparison to MCNPX of one-group reaction rates in a sodium fast reactor. The absorption rates for the uranium isotopes are provided. Our method is noted as "CPM". These reaction rates encompass the entire energy range that is modeled, from 400 {eV} to 10 {MeV}. The numbers are normalized to 100,000 total absorptions in the system. All errors are below 5%.

nuclide	reaction	MCNPX	CPM	error (%)
U-235	fission	56438	56325	-0.20
	capture	14802	14694	-0.73
U-238	fission	5126	5340	4.17
	capture	22848	22870	0.10

("CPM") to those obtained from MCNPX. We focus on the absorption reactions with uranium. In most cases, our method presents minimal error. The numbers presented are normalized to a total of 100,000 absorptions in the entire system (in both the fuel and moderator).

4.1.7 Criticality

A summary of the multiplication factors for both the fast and thermal reactors is provided in Table 4.6. The error in the thermal results is very small, especially when compared to the same model in MCNPX. The discrepancy with the published Rowlands results is attributed to the smearing of the zirconium fuel rod in our simulation. An error of less than 10 mk is considered acceptable (Schneider et al., 2006a).

Table 4.6: Comparison to MCNPX and published Rowlands results of the multiplication factor for both the fast and thermal reactor. Our method is noted as “CPM”. Our model provides very accurate results in comparison to MCNPX simulations. There is greater error between our model and the results provided by the Rowlands benchmark because we have smeared the zirconium fuel rod in the benchmark across the moderator.

run	comparison	CPM	error (mk)
fast	1.536340	1.539315	2.975
Rowlands	1.390110	1.401140	11.030
Rowlands in MCNPX	1.397500	1.401140	3.640

4.2 A Four-Cell Reactor

We present three sets of results for four-cell geometries. The first set is for a fast reactor, and the second two are for a thermal reactor.

4.2.1 Fast Reactor

The reactor is fueled by highly enriched uranium dioxide and by plutonium dioxide. We present the neutron spectrum for the uranium dioxide cell in Figure 4.9, and for the plutonium dioxide cell in Figure 4.10. The results align fairly well with MCNPX results. This indicates that the current method is satisfactory for the analysis of multi-region fast reactors. The difference in the multiplication factor between MCNPX and our method is only 10 mk.

4.2.2 Thermal Reactor

We present a light water moderated reactor composed of low enriched uranium, using the same pin composition as in the Rowlands benchmark, as

Pin cells

HEU	
diam.	0.8000 cm
temp.	294.0 K
nuclide	N (#/b/cm^2)
U-235	5.6642×10^{-3}
U-238	2.2604×10^{-2}
O-16	4.6624×10^{-2}

PuO$_2$	
diam.	0.8000 cm
temp.	294.0 K
nuclide	N (#/b/cm^2)
Pu-239	2.5560×10^{-2}
O-16	5.1119×10^{-2}

Annular cells

HEU sodium	
pitch	1.2000 cm
temp.	294.0 K
nuclide	N (#/b/cm^2)
Na-23	2.3111×10^{-2}

PuO$_2$ sodium	
pitch	1.2000 cm
temp.	294.0 K
nuclide	N (#/b/cm^2)
Na-23	2.3111×10^{-2}

Figure 4.8: Dimensions and composition of a four-cell UOX and plutonium doixide fueled reactor. The ratio of UOX pins to plutonium pins is 3:1.

well IMF (inert matrix fuel). The geometry and composition of the system are described in Table 4.7. The four-cell thermal reactor results are not as promising as the fast reactor results. This is largely because the system does not appear homogeneous to neurons in the thermal range. The flux is qualitatively different from that given by MCNPX. At 42 mk, the error in the multiplication factor is not acceptable.

For this reason, we explored a method to account for the fact that the system appears heterogeneous at low energies, where the hydrogen cross section is high and the neutron mean free path is small. We modify the theory according to Section 2.6, and obtain the new results shown in Figure 4.13 and Figure 4.14.

This correction slightly improves our results, as the coefficients of determination have come closer to unity. Additionally, the error in the multiplication

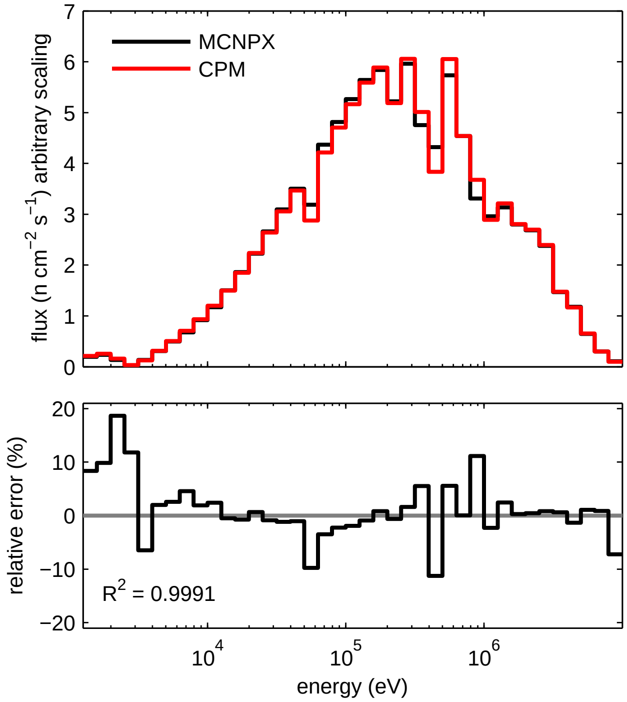

Figure 4.9: Four-cell fast reactor spectral flux in the uranium fuel pin. The correlation between MCNPX and our method ("CPM") is fairly high.

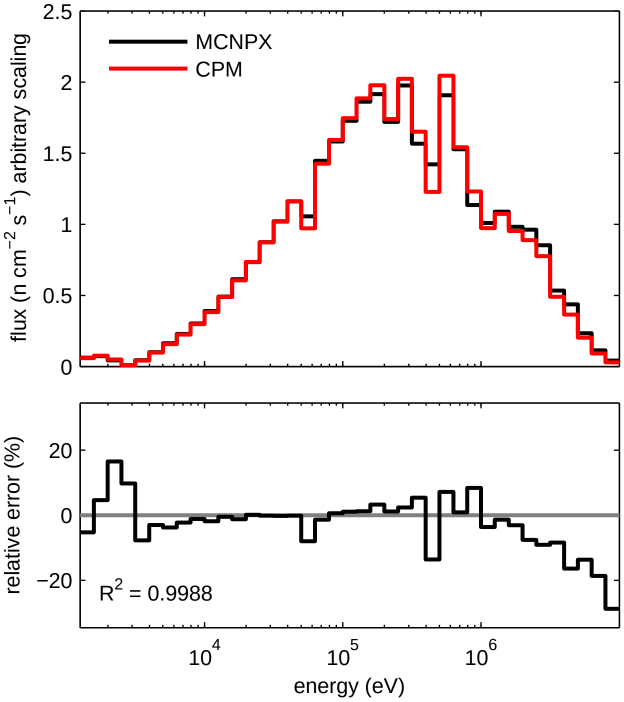

Figure 4.10: Four-cell fast reactor spectral flux in the plutonium fuel pin. The correlation between MCNPX and our method ("CPM") is fairly high.

Table 4.7: Dimensions and composition of a four-cell model moderated with water and fueled with enriched UOX and IMF (inert matrix fuel). The ratio of UOX pins to IMF pins is 3:1. The dimensions and coolant for both of these pins are the same.

Pin cells

UOX

diam.	0.8000 cm
temp.	294.0 K

nuclide	N (#/b/cm^2)
U-235	7.0800×10^{-4}
U-238	2.2604×10^{-2}
O-16	4.6624×10^{-2}

IMF

diam.	0.8000 cm
temp.	294.0 K

nuclide	N (#/b/cm^2)
Pu-239	1.4704×10^{-3}
Pu-240	4.3800×10^{-4}
Pu-241	3.0650×10^{-4}
Pu-242	8.6500×10^{-5}
Zr-90	1.8900×10^{-2}
Zr-94	6.3844×10^{-3}
Y-89	6.4825×10^{-3}

Annular cells

UOX water

pitch	1.2000 cm
temp.	294.0 K

nuclide	N (#/b/cm^2)
H-1	6.6988×10^{-2}
O-16	3.3414×10^{-2}

IMF water

pitch	1.2000 cm
temp.	294.0 K

nuclide	N (#/b/cm^2)
H-1	6.6988×10^{-2}
O-16	3.3414×10^{-2}

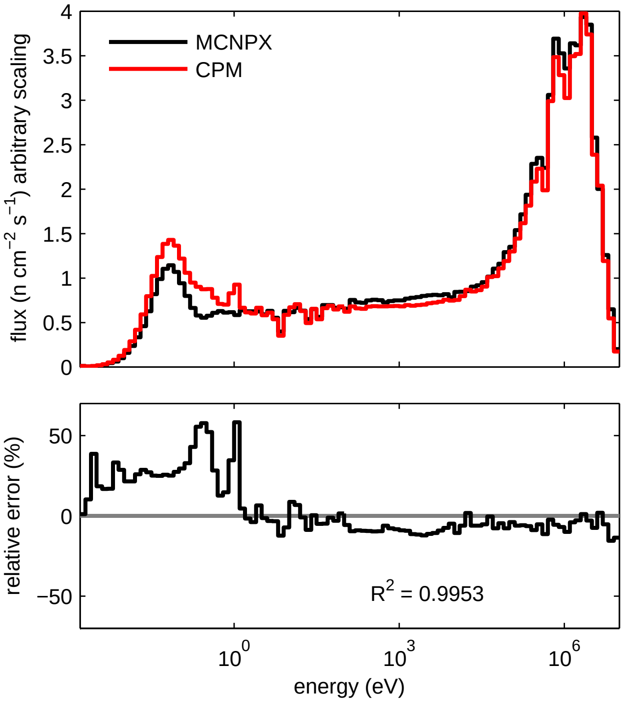

Figure 4.11: Four-cell thermal reactor spectral flux in the UOX fuel.

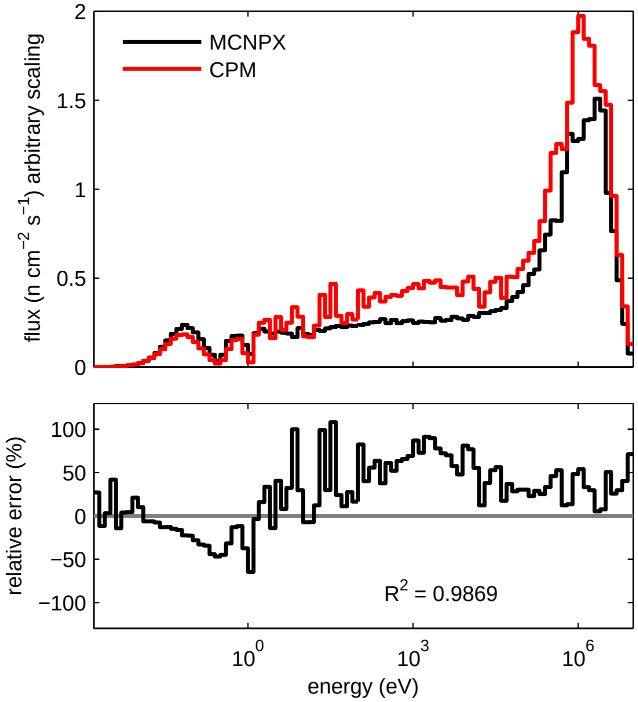

Figure 4.12: Four-cell thermal reactor spectral flux in the IMF fuel.

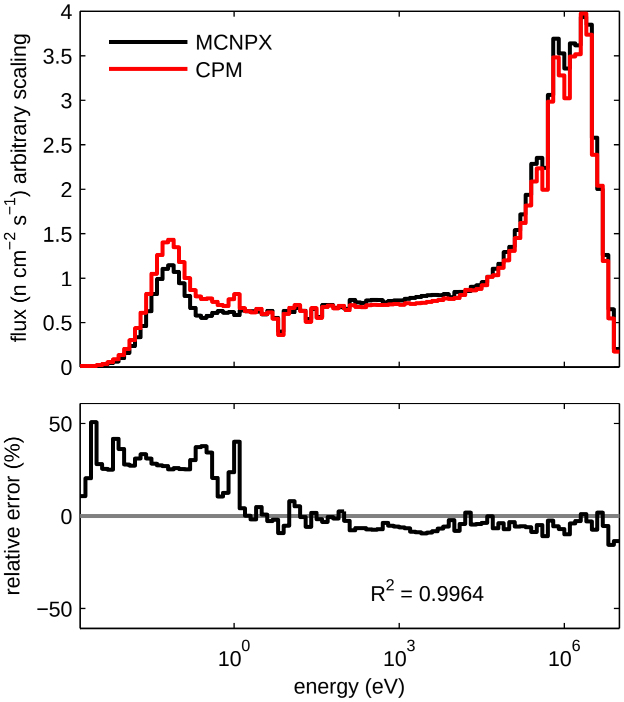

Figure 4.13: Four-cell thermal reactor spectral flux in the UOX fuel, homogeneity corrected.

factor drops from 42 mk to 31 mk.

Alternate implementations of the method provided in Section 2.6 are possible, and may lead to further improvement of the results.

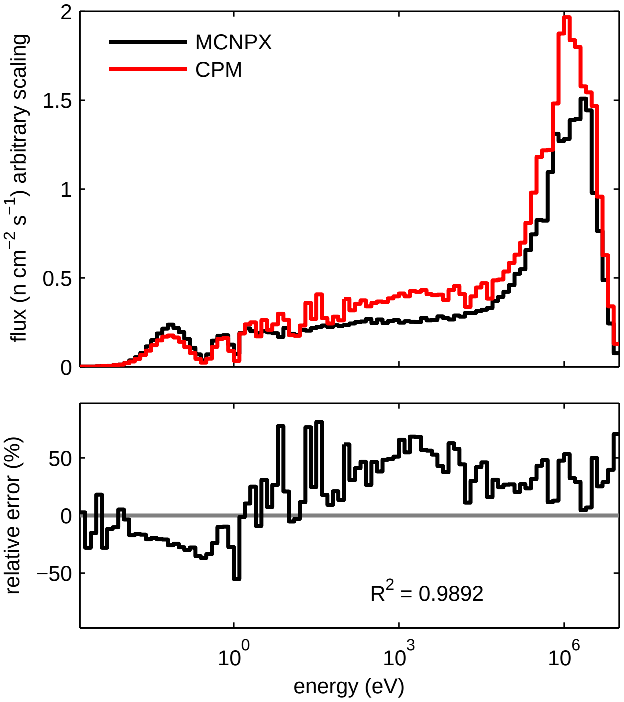

Figure 4.14: Four-cell thermal reactor spectral flux in the IMF fuel, homogeneity corrected.

Appendices

Appendix A

Relevant Cross Sections

Cross section plots are shown for all nuclides in the systems presented in Chapter 4. For each nuclide, the total cross section is compared to the transport corrected total cross section. It is evident that except for hydrogen, this correction only affects cross sections at high energies.

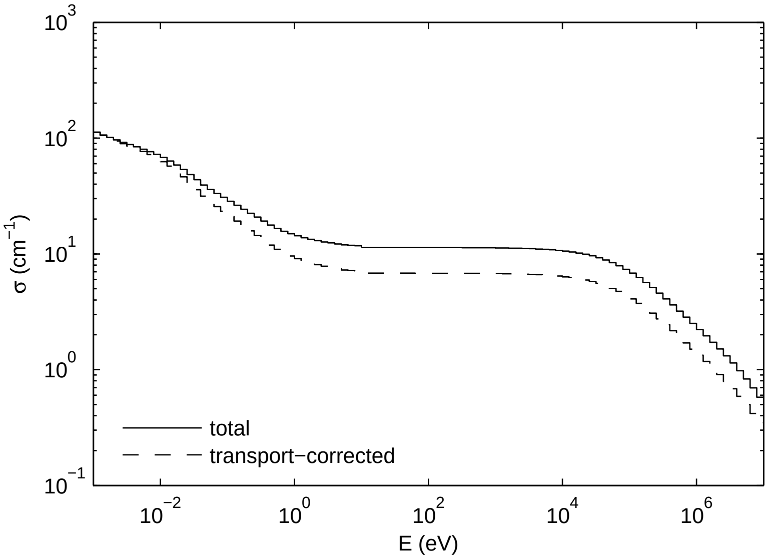

Figure A.1: Total microscopic cross section for H-1 bound in water.

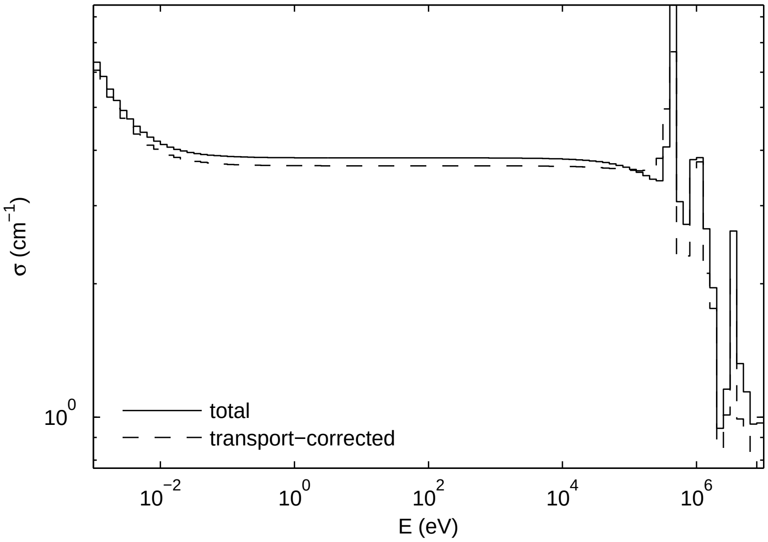

Figure A.2: Total cross section for O-16.

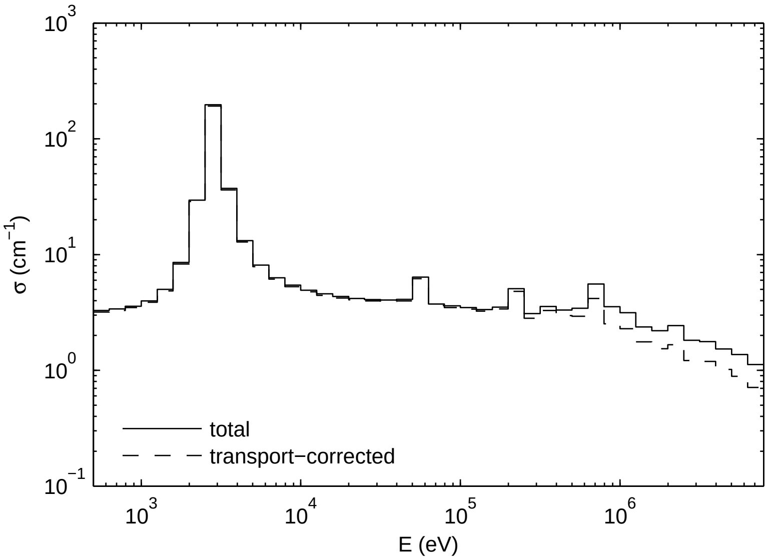

Figure A.3: Total cross section for Na-23.

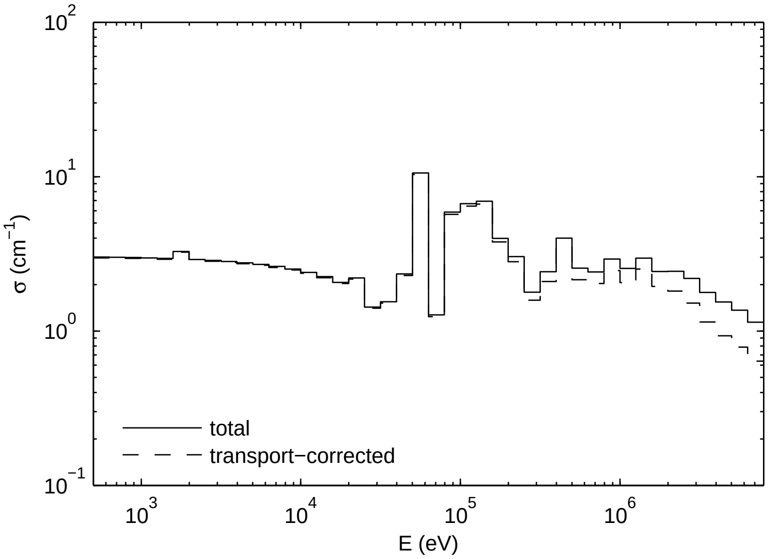

Figure A.4: Total cross section for Cr-52.

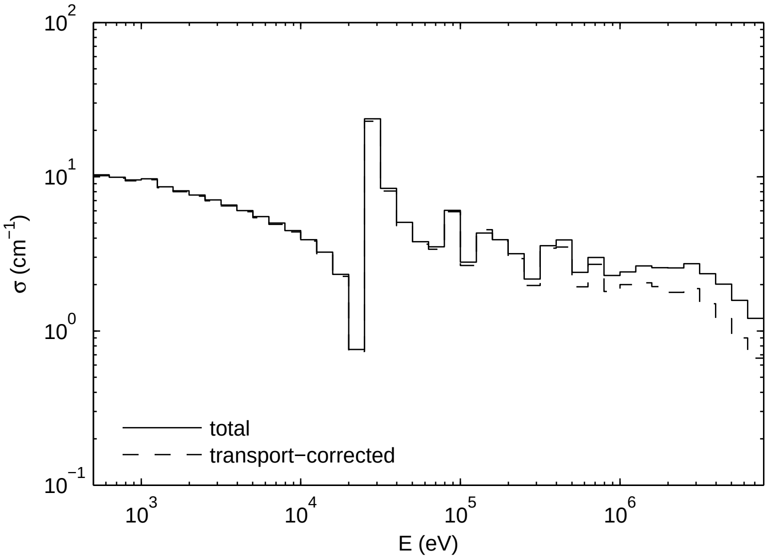

Figure A.5: Total cross section for Fe-56.

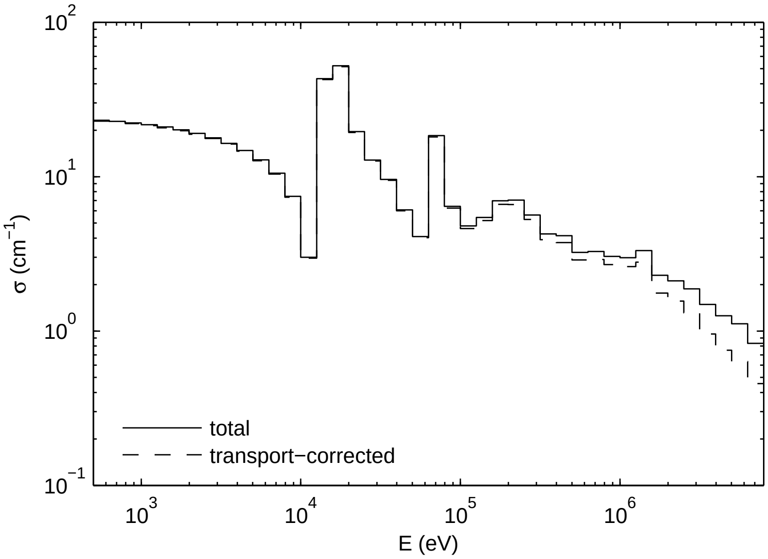

Figure A.6: Total cross section for Ni-58.

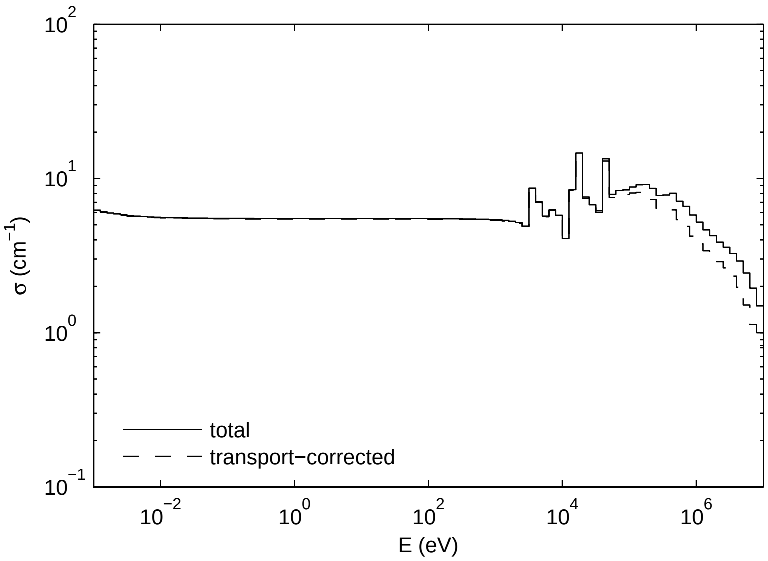

Figure A.7: Total cross section for Zr-90.

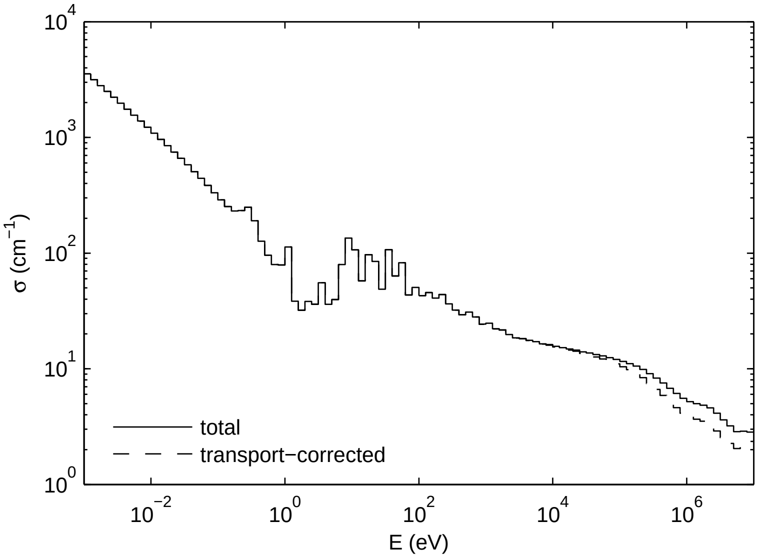

Figure A.8: Total cross section for U-235.

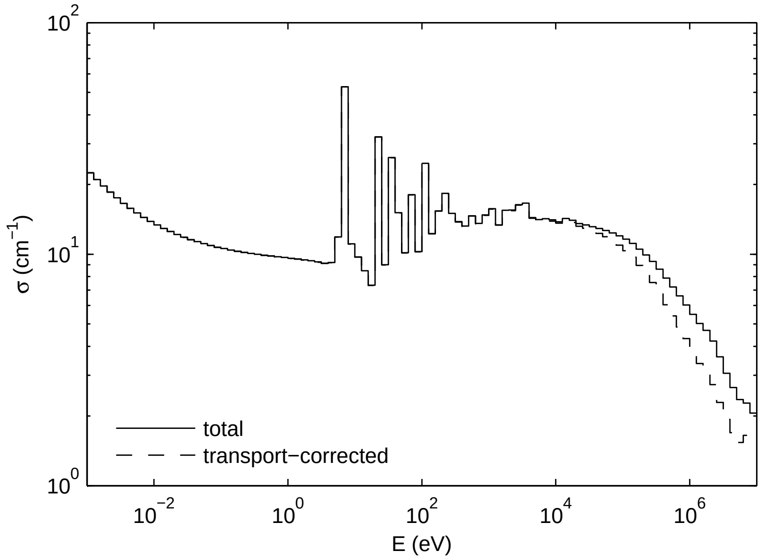

Figure A.9: Total cross section for U-238.

Appendix B

Code Manual

The code used to implement the methods described in this work has been called VBUDSII, after the VBUDS (Visualiziation: Burnup, Depletion, Spectra) collision probability spectral and burnup code developed by Schneider (2002). A description of the modules and data structures of VBUDSII is given in this appendix.

Currently, three versions of VBUDSII exist. The first is the development version. The second is a cleaned-up version of the development version that was created after the code yielded accurate results. The third is a loose object-oriented implementation, named VBUDSIIOO, that was created solely for debugging purposes. All results presented in this work were generated using the development version. All three versions are implemented in MATLAB . Except for one section, this manual focuses on the cleaned-up version of VBUDSII .

B.1 Modules

The VBUDSII code is intended to (1) provide self-shielded multigroup cross sections given a cell geometry, (2) produce the energy-dependent neutron spectrum, (3) perform diffusion on three homogeneous regions or fuel campaigns,

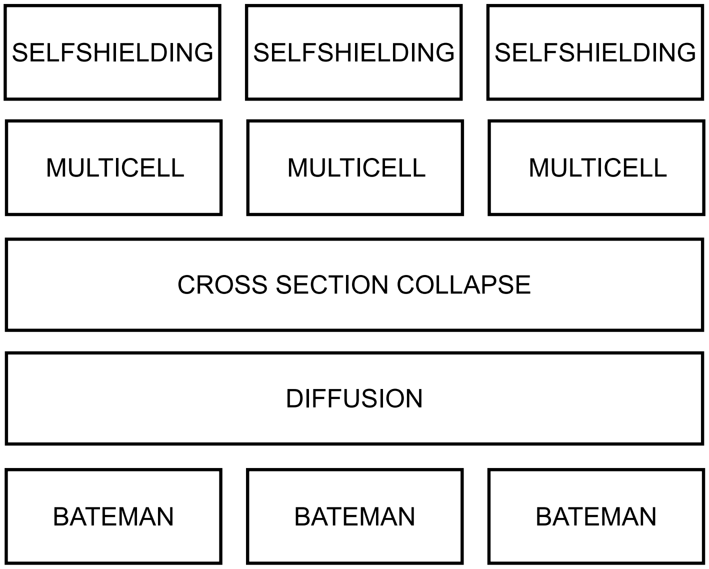

Figure B.1: The modules planned for VBUDSII. Development on the BATEMAN and DIFFUSION modules was performed at Cornell, and this thesis discusses the MULTICELL module. The cross section collapse is the procedure of collapsing many-group cross sections into few-group (e.g. 3) cross sections averaged over space. The cross section collapse and diffusion modules are performed across all three regions in the system, while the self-shielding, spectral solution, and burnup are performed individually for each of the three regions.

and (4) update the system's material composition through a burnup calculation. The interconnection of these modules is shown in Figure B.1. In this work, only (1) and (2) are completely implemented. However, progress has been made on (3) and (4). The manual only describes (1) and (2).

DATAPROCESSING Microscopic cross section (as well as ν and $\bar{\mu}$) tables are interpolated in temperature and (the initially guessed) background cross section to obtain system-dependent data. The microscopic cross sections are cell-collapsed to obtain macroscopic cross sections for each cell, and the derived quantities $\nu\Sigma_f$ and Σ_t are computed. The transport correction is also applied before Σ_t is formed. Then, new values for the background cross section are obtained from SELFSHIELDING and part of MULTICELL, and the interpolation in background cross section is performed again. This is repeated until the background cross section everywhere converges.

SELFSHIELDING The background cross section is calculated for each nuclide in each cell (at all energies). This module requires that $\mathbf{\Pi}$ has been calculated.

MULTICELL This module contains the function `CreatePi()` that computes $\mathbf{\Pi}$, as well as the function `SpectralMatrix()` that places the macroscopic cross section data into matrix Equation 2.31 and the function `SpectralSolve()` that solves the matrix equation.

DIFFUSION While much code is present for this module, it is not integrated.

BATEMAN This module has not been written, but the `Bateman()` function provides pseudocode for its implementation.

A brief description of all relevant functions is provided here. We begin with functions that are not part of one of the above modules. All variables mentioned here are described shortly.

`Vbudsii()` This is the main function, and requires `Param` and `Geom` as input. This function calls all modules in turn. The function returns `Results`, `Param`, `Geom`, and `Lib`.

`initparam()` Initializes a `Param` `struct` that can be used as an input to `Vbudsii()`. This can be used as a reference to see what fields are required to run the code.

`initgeom()` Initialize a `Geom` `struct`.

`initunitcell()` Initialize a `uc` `struct`, which becomes part of the `Geom` `struct`.

`Preprocessor()` Checks `Geom` for errors, runs `InterpFissionSpectrum()` and initializes `Results`.

`Postprocessor()` Performs post-processing of data, such as creating plots.

Note the code is organized using MATLAB packages, which are created using folders whose names are prepended with an addition ("+") symbol. To call `Vbudsii()`, the user types:

```
vbudsii.Vbudsii(Param, Geom)
```

in the command window, assuming the folder `+vbudsii/` is on the MATLAB path. We now describe the remaining functions.

B.1.1 Dataprocessing

`ResolveXS()` Runs most of the other functions in this module to interpolate data and perform self-shielding.

`ResolveTemp()` Interpolates `Lib` by temperature and stores the results in `Results.Region().Cell().fine().z().t()`.

`ResolveXS()` Interpolates data in `Results` by background cross section σ_0 and stores the results in `Results.Region().Cell().fine().z().s`.

`InterpFissionSpectrum()` Returns a Watt fission spectrum.

`CollapseCell()` Computes the derived quantities $\nu\Sigma_f$, Σ_t, and manages the $(n, 2n)$ interaction if requested. Creates cell-level macroscopic cross sections and stores them in `Results.Region().Cell().fine().value`.

`CollapseData()` Calculates reaction rates and few-group cross sections and stores them in `Results.Region().Cell().one()` and `Results.Region().Cell().few()`

`CollapseEnergy()` Collapses cross sections in energy to create few-group cross sections from fine-group cross sections. This is called by `CollapseData()`.

B.1.2 Selfshielding

`Selfshielding()` Runs `CalculateS0()` for all cells and nuclides.

`CalculateS0()` Uses $\mathbf{\Pi}$ and the cell-level macroscopic cross section to compute the escape and background cross section for a particular cell and nuclide. The results are stored in `Results.Region().SEscapes` and `Results.Region().S0s`

B.1.3 Multicell

`Multicell()` Calls `SpectralMatrix()` and `SpectralSolve()` once $\mathbf{\Pi}$ has been calculated.

`CreatePi()` Computes the transport probabilities matrix $\mathbf{\Pi}$, which is stored in `Results.Region().PI`.

`SpectralMatrix()` Arranges cross section data and $\mathbf{\Pi}$ into a matrix equation.

`SpectralSolve()` Solves the matrix equation returned by `SpectralMatrix()` and obtains the multiplication factor and spectrum of the reactor. This function also computes the power density of the system. These results are stored in `Results.Region().kInf` and `Results.Region().spectralFlux`.

B.2 Data Structures

All of the important data that flows through VBUDSII is stored in one of four `structs`. Parameters that control options are stored in `Param`, and the definition of the reactor geometry and composition is stored in `Geom`. These two `structs` constitute the input to VBUDSII . All data output is stored in `Results`, and the cross section library is stored in `Lib`. These are often abbreviated, respectively, as `p`, `g`, `R`, and `L`. The VBUDSIIOO code implements `structs` that are substantially improved over the ones presented here, and it is hoped that the project migrates to the use of those `structs` instead of those presented here. Figure B.2 illustrates how the data structures are connected to the modules.

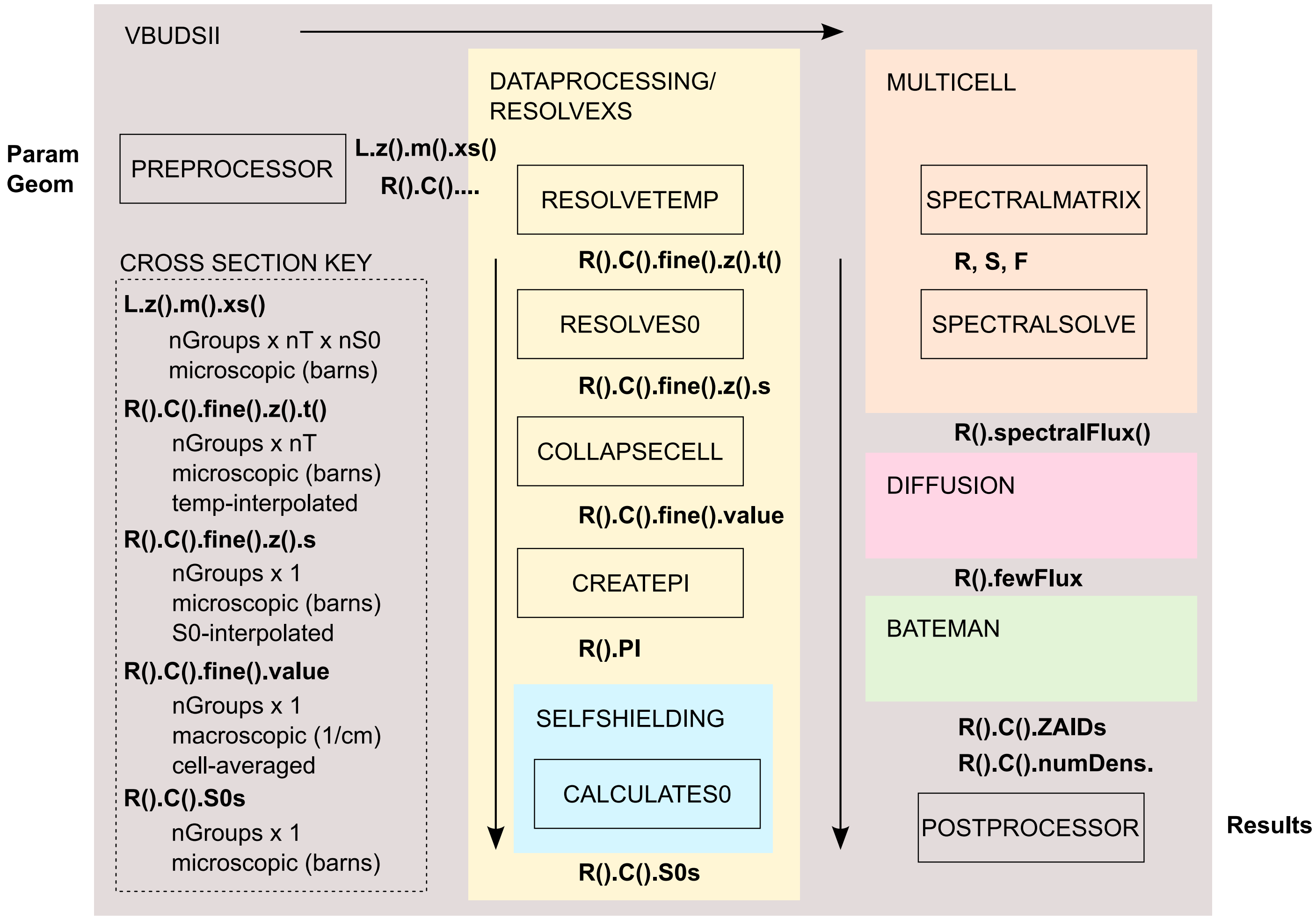

Figure B.2: VBUDSII data structures. The fields of the structures are shown next to the function that creates them. “R” represents for the `Results.Region` `struct`, “C” represents the `Results.Region.Cell` `struct`, and “R, S, F” represent the three matrices returned by `SpectralMatrix()` for the removal, scattering source, and fission source. Parenthesis pairs in the figure are interpreted as “for all” (e.g. “R()” means “for all regions”).

B.2.1 Parameter Input

The `Param` struct can be initialized by the function `initgeom()`. The text of that function is placed here for reference. Descriptions are placed as comments.

```
p = struct('nFineGroups', [], ... % length(fineGroupDef)-1
             'nFewGroups', [], ... % length(fewGroupDef)-1
           'fineGroupDef', [], ... % units of eV
            'fewGroupDef', [], ... % units of eV
             'nTimeSteps', 1, ... % For burnup.
              'resolveXS', 1, ... % Self-shielding?
                'verbose', 1, ... % VBUDSII talks to you.
           'XSLibraryMAT', '', ... % located in +vbudsii/+data/
           'S0iterthresh', 0.00001, ...
                 'doPlot', 0 ... % Plots output.
    );
```

Some of the fields above require further description and are described below.

`fineGroupDef` Fine-group energy group structure for spectral calculations. There are $G + 1$ values for G groups, since the left and right edge of each group must be provided. Units are {eV}.

`fewGroupDef` Few-group energy group structure for few-group reaction rates and diffusion (when it is implemented). Units are {eV}.

`resolveXS` A value of `false` tells VBUDSII skip the self-shielding of cross sections. In this case, the field `p.constS0` can be specified to set the value of σ_0 to be used for all cross sections.

`XSLibraryMAT` Name of the library MAT-file to load. The file must be located in the `+vbudsii/+data/` directory.

`S0iterthresh` Threshold for error in σ_0 between iterations. This controls when the self-shielding iterations stop.

Other fields that may be used in the development version of the code are presented in Section B.4.

B.2.2 Geometry Input

The geometry input allows the specification of multiple reactor regions. These would be used in the diffusion of neutrons between regions, such as if the reactor was composed of three fuel campaigns. In this work only one region is used.

```
g = struct('nRegions', [],...
          'regionDef', struct('name','',...
                      'uc', struct(), ...
                  'nCells',[],...
              'relVolumes',[],... % unitless
           'isFissionable',[],... % logical
                 'cellDef',struct('name','',...
                         'isFissionable',[],... % logical
                             'initZAIDs',[],...
                      'initNumDensities',[],... % #/b/cm
                           'initDensity',[],... % g/cm^3
                              'initTemp',[]...   % K
    )));
```

The `isFissionable` fields are actually filled in by the code, and the user does not set them. The ZAID identifies a nuclide by concatenating its proton number with its atomic number. For example, the ZAID for U-235 is 92235. The user must also specify the unit cell `uc` for each region. An example of this `struct` is provided below for a four-cell region.

```
uc = struct();
uc.pinPitch = sqrt(0.538^2 * pi)*[1 1]; % cm
uc.pinDiam = 2 * 0.289*[1 1]; % cm
uc.f = [0.75 0.25]; % alpha % unitless weight, sums to 1.
uc.g = [0.75 0.25]; % beta % unitless weight, sums to 1.
uc.sauerConst.mod = 2.35;
uc.sauerConst.fuel = 5.00;
uc.PinCellNames = {'UOX fuel', 'IMF fuel'};
uc.AntipinCellNames = {'UOX coolant', 'IMF coolant'};
```

The last two fields must match the names given on the `name` field in the `Geom`, or g, `struct`. An example `Geom` input is provided:

```
g.regionDef(1).uc = uc;

enrichment = .026;
density_H2O = .72; % g/cm^3
density_UO2 = 11; %g/cm^3
[ao, wo, N_H2O] = matl([1001 2;
                       8016 1], 1, density_H2O);
[ao, wo, N_UO2] = matl([92235 enrichment;
                        92238 1-enrichment;
                        8016 2], 1, density_UO2);

% define
% one region with two cells: UO2 and H2O
g.nRegions = 1;
g.regionDef(1).name = 'campaign1';
g.regionDef(1).nCells = 2;
g.regionDef(1).relVolumes = [...
    pi/4*uc.pinDiam^2, ...
    uc.pinPitch^2-pi/4*uc.pinDiam^2 ...
    ];
% UO2 cell
g.regionDef(1).cellDef(1).name = 'fuel';
g.regionDef(1).cellDef(1).initZAIDs = [92235 92238 8016];
g.regionDef(1).cellDef(1).initNumDensities = N_UO2';
g.regionDef(1).cellDef(1).initDensity = density_UO2;
g.regionDef(1).cellDef(1).initTemp = 900;
% H2O cell
g.regionDef(1).cellDef(2).name = 'coolant';
g.regionDef(1).cellDef(2).initZAIDs = [11 8016];
g.regionDef(1).cellDef(2).initNumDensities = [N_H2O];
g.regionDef(1).cellDef(2).initDensity = density_H2O;
g.regionDef(1).cellDef(2).initTemp = 600;
```

We have made use of the `matl()` function, which is described in Section B.3.

B.2.3 Results

The `Results` `struct` consists of only two fields, though all of the output is stored in `Region`, which is itself a `struct`. `Region` is initialized by the following:

```
Region = struct('spectralFlux',[],... % nGroups x nCells
    'fewFlux',[],... % unused so far.
    'kInf',[],... % highest eigenvalue.
    'kInfall', [], ... % all eigenvalues.
    'relativePower',[],... % scalar.
    'PI',[],... % nCells x nCells x nFineGroups
    'powerDensity',[],... % scalar.
    'few',struct('value',[]),... % few-group XS's.
    'pi2cellIdxs', [], ... % index mapping, internal.
    'cell2piIdxs', [], ... % index mapping, internal.
    'Cell',struct('spectralFlux',[],... % fine flux result.
         'fewFlux',[],... % collapsed flux.
         'kInf',0,... % unused.
         'temp',0,... % taken from Geom.
         'ZAIDs',[],... % row vector of composition.
         'numDensities',[],... % row vector (n/b/cm)
         'S0s',[],... % background xs, nFineGroups x nCells
         'SEscapes',[],... % escape xs, nFineGroups x nCells
         'one',struct('RR',[],... % 1-group data
                      'value',[],...
                      'z',struct('t',[],...
                                 's',[], ...
                                 'RR',[])),...
         'few',struct('RR', [], ... % few-group data
                      'value',[],...
                      'z',struct('t',[],...
                                 's',[], ...
                                 'RR', [])),...
         'fine',struct('value',[],... % fine-group data
                       'z',struct('t',[],...
                                  's',[]))));
```

The `Results` `struct` is then given by the following:

```
Results = struct('runtime', [], ...
                 'Region', struct());
```

The `Results` `struct` stores all unshielded microscopic and self-shielded microscopic and macroscopic cross sections, as well as reaction rates, the multiplication factor, and reactor composition (since if burnup is performed the composition may change). In the last few fields, `one`, `few`, and `fine`, the `z` subfield stores cross section data for each nuclide (ZAID), and accordingly the length of this field is the number of nuclides in the cell. The `t` subfield provides temperature-interpolated cross sections as an `nGroups x nS0s` matrix, where "S0" is the base-10 logarithm of the background cross section $\log_{10}(\sigma_0)$. The `s` subfield provides self-shielded temperature-interpolated cross sections as an `nGroups x 1` array.

Some examples of the use of the `Results` `struct` at the MATLAB command prompt are provided below.

```
R = Results.Region(1);
R =
     spectralFlux: [42x2 double]
          fewFlux: [3x1 double]
             kInf: 1.5393
          kInfall: [84x1 double]
    relativePower: [2.4000e-11 0]
               PI: [2x2x42 double]
     powerDensity: 6.0000e-12
              few: [1x52 struct]
      pi2cellIdxs: [2 1]
      cell2piIdxs: [2 1]
             Cell: [1x2 struct]
              one: [1x52 struct]
>> length(R.Cell)
ans =
     2
```

The length of `R.Cell` is 2 because there are two cells in region 1.

`Cell` itself is a `struct` containing cell-specific information:

```
>> R.Cell(1)
ans =
     spectralFlux: [42x1 double]
          fewFlux: [3x1 double]
             kInf: 0
             temp: 900 % in Kelvin.
            ZAIDs: [92235 92238 8016] % nuclides
     numDensities: [0.0063 0.0183 0.0492] % densities
              S0s: [42x3 double] % nGroups x nZAIDs
         SEscapes: [42x3 double] % nGroups x nZAIDs
              one: [1x52 struct] % 1-g data
              few: [1x52 struct] % few-group data
             fine: [1x52 struct] % fine-group data
    relativePower: 0
```

The fine-group microscopic and macroscopic cross sections are stored in the `fine` subfield. The length of this subfield is the number of MT's present in the data library associated with this VBUDSII run.

```
>> size(R.Cell(1).fine)
ans =
     1    52 % there are 52 MT's (reactions) in L.
```

MT = 7 yields the total cross section.

```
>> R.Cell(1).fine(L.MT(7))
ans =
    value: [42x1 double] % macroscopic for cell
        z: [1x3 struct] % microscopic for each ZAID
```

The length of `z` is 3 because there are 3 nuclides in the cell.

```
>> R.Cell(1).fine(L.MT(7)).z(1)
ans =
    t: [42x7 double] % nGroups x nTemps; interp'd by T
    s: [42x1 double] % nGroups x 1; interp'd by S0
```

The `t` subfield is populated by `ResolveTemp()`, and the `s` subfield is populated by `ResolveS0()`. See the code for more information.

B.2.4 Data Library

The `Lib struct` has the following form:

```
L = struct('groupDef',p.p.groupDef,... % eV
              'nGroups',length(p.p.groupDef)-1,...
              'ZAIDs',ZAIDs,...
              'MTs',MTs,...
              'mainMTs',mainMTs,...
              'Ts',Ts,...
              'S0s',log10(S0s),...
              'ZAID',[],...
              'MT',[],...
              'mainMT',[],...
              'T',[],...
              'S0',[],...
              'MTmap',[],...
              'z',struct('isFissionable',[],...
                         'inelasticMTs',[],...
                    'm',struct('hasResonances',[],...
                          'xs',[])));
```

The fields `ZAIDs`, `MTs`, `mainMTs`, `Ts`, and `S0s` list the values that are available for each of these settings. The fields `ZAID`, `MT`, `mainMT`, `T`, and `S0` sparse arrays (except for `S0`, which is a `container.Map`) that are used to index into the library. All the cross section data is stored in the `z` field, and its usage is shown below. The `mainMTs` field is used in the code for looping through only the relevant interactions. The names of the relevant MT's is found in the `L.MTmap` `container.Map`.

The following presents three examples of library usage.

```
% Total cross section for hydrogen bound in water.
L.z(L.ZAID(11)).m(L.MT(7)).xs(:,L.T(600),L.S0(10))
% Capture for O-16 at T = 600 and S0 = 10.
L.z(L.ZAID(8016)).m(L.MT(102)).xs(:,L.T(600),L.S0(10))
% Total scattering kernel for hydrogen bound in water.
L.z(L.ZAID(11)).m(L.MT(2)).xs(:,:,L.T(600),L.S0(10))
```

To accommodate our usage of ENDF data, we have introduced the following new symbols:

ZAID 11 Hydrogen bound in water (1001 is free gas hydrogen).

MT 7 The total cross section obtained from the sum of capture, fission, elastic scattering, and inelastic scattering. The total cross section is also modified by the inclusion of the (n,2n) cross section.

MT 8 This used to be the transport cross section, but is now deprecated.

MT 9 The $\nu\Sigma_f$ cross section.

The user may create a library of this form by any means, but we have used the functions `FarmXS()` and `FarmXSVIIp1()` to do so.

B.3 Utility Functions

A number of functions have been developed that aid the use of VBUDSII , but that are not part of VBUDSII . Most of these have been placed in the `util/` folder of VBUDSII .

`FarmXS()` Runs NJOY to create `Lib`. The input to this function is fairly detailed.

`FarmXSVIIp1()` Runs NJOY to create `Lib`. The input to this function is fairly detailed. This version is more up to date and works only with ENDFVII. It has only been tested ENDFVII.1.

`VbudsiiVsMcnpx()` Takes the output of VBUDSII and MCNPX runs and creates a report of the results.

`VbudsiiVsVbudsii()` Takes the output of two MCNPX runs and creates a report of the results.

`matl()` Provides density information for a mixture of isotopes.

`visualizelibrary()` Takes a `Lib` and plots various cross sections across the library.

`writeMCNPXSpectralInput()` Writes an MCNPX input file, given a VBUDSII input. This has been replaced by the `McnpxWrap` class.

`writeSerpentSpectralInput()` Writes an SERPENT input file, given a VBUDSII input. This has been replaced by the `SerpentWrap` class.

printnuclearplot() Creates a latex plot of cross section or flux data.

TallyPull2() Parses an MCNPX output file. Written by Geoff Recktenwald.

McnpxWrap A class to create, run, and post-process an MCNPX run generated from a VBUDSII input.

SerpentWrap A class to create, run, and post-process a SERPENT run generated from a VBUDSII input.

B.4 Development Options

In the course of development of VBUDSII , many “tweaks” were made to the code so that it could operate in different ways depending on certain fields of the `Param` `struct`. For instance, at some point in development the method for computing the fission spectrum was changed. In order to maintain the behavior of the code, the new method was implemented by placing the following line in the code:

```
if isfield(p, ’useWatt’) && p.useWatt
```

This line checks to see if the field `useWatt` exists in the `Param` `struct`. If it does and if its logical value is `true`, then the Watt fission spectrum is used in lieu of a tabulated fixed-group array. These options are listed below, along with the function in which the option is present. Only a brief description is given; refer to the code for more information.

Many of the available options lead to incorrect and invalid results. They exist because they were used to investigate errors. As a result, a new version of the code exists in which all options that lead to incorrect results have been stripped. None of the options need to be specified for the code to run, though to obtain correct results a number of the options must be used. In the cleaned-up version of the code, by default the options that lead to correct results are used. However, the following options are still present in the new version and their use still makes sense: **`usen2nn`**, **`homogenize`**, **`constS0`**, **`AntipinCellNames`** and **`PinCellNames`**, and **`mode`**.

Sample usage of most of these options is provided in the scripts folder.

`immutableMyMTs` in `MakeLibrary()`, `CollapseCell()`: Chooses between three options for the creation of derived cross sections ($\nu\Sigma_f$, Σ_t and possibly Σ_{tr}).

`transscale` in `CollapseCell()`: Scales the transport cross section for all materials using the value of `p.transscale`.

`transscalewateronly` in `CollapseCell()`: Same as `transscale`, except that only the water cross section, if it exists, is scaled.

`macromultNuFission` in `CollapseCell()`: Toggles whether the $\nu\Sigma_f$ cross section is computed before or after macroscopic cross sections are formed. It is incorrect to do the latter.

`macromultTransport` in `CollapseCell()`: Same as above, except for the transport cross section.

`usen2nn` in `CollapseCell()`: If an (n,2n) kernel is available (MT = 16), then it is correctly subtracted from the absorption cross section and added to the scattering kernel. The method used to include the (n,2n) cross section is much like that used in WIMS-D (Leszczynski et al., 2007).

`stammlertransport` in `CollapseCell()`: Uses the method presented clearly by Stamm'ler for performing the transport correction. In this method, the scattering kernel is reduced. If this option is not used then transport is created as a separate cross section.

`nomubarforo16` in `CollapseCell()`: $\bar{\mu}$ for O-16 is set to zero.

`nomubarforu235` in `CollapseCell()`: Same as above, but for U-235.

`nomubarforu238` in `CollapseCell()`: Same as above, but for U-238.

`hboumdu` in `CollapseCell()`: A crude piecewise linear approximation for the bound hydrogen $\bar{\mu}$ is used.

`homogenize` in `ResolveXS()`: Uses `CreatePi4()` for the four-cell homogeneity correction described in Section 2.6. Must specify four cells to use this.

`useMCNPXTalyXS` in `ResolveXS()`, `Multicell()`: Uses MCNPX cross sections in the code, when possible, instead of the cross sections in `Lib`. The value of the field must be a modified `TallyP` `struct` returned by Geoff Recktenwald's `TallyPull2()` or the `McnpxWrap` class.

`iFuelPin` in `ResolveXS()`: Specifies the index of the fuel pin in `Geom`.

`addInInelastic` in `ResolveXS()`: Adds the inelastic cross section (ENDF MT=51 through MT=91) to the elastic cross section (MT = 2). Only relevant when using MCNPX cross sections; the addition of the inelastic cross section is otherwise taken care of by `MakeLibrary()`.

`runtimeXSplots` in `ResolveXS()`: Creates cross section plots while VBUDSII is running.

`doctorKernel` in `ResolveXS()`, `InterpFissionSpectrum()`: Edit the kernel so that only one element is nonzero.

`MCNPXhomewaterkernel` in `ResolveXS()`: Replace water's scattering kernel by the one provided as the value of `p.MCNPXhomewaterkernel`.

`MCNPXwaterCorrectionFactor` in `MakeLibrary()`: Scales all water cross sections by the value of this field.

`MCNPXcorrectTheseMTIDXS` in `MakeLibrary()`: Deprecated. If an external library is used for water, then chooses which cross sections from this external library are used.

`MCNPXcorrectKernel` in `MakeLibrary()`: Similar to the above.

`waterPiece` in `MakeLibrary()`: Use an external library for the scattering kernel for hydrogen bound in water.

`waterWhole` in `MakeLibrary()`: Deprecated. Use an external library for the scattering kernel for ZAID 222, which used to be water.

`noInelastic` in `MakeLibrary()`: Stops the combining inelastic scattering into elastic scattering.

`zeromu` in `MakeLibrary()`: Set $\bar{\mu}$ to zero for all nuclides.

`waterTransportFromParts` in `MakeLibrary()`: Deprecated. Affects how the transport cross section for water (ZAID 222) is computed.

`useWatt` in `InterpFissionSpectrum()`: Uses a Watt fission spectrum instead of a tabulated one. The parameters for the spectrum are taken from the MCNPX manual (Pelowitz, 2011).

`AntipinCellNames` in `CreatePi()`: If the user has specified a particular numbering of cells, use this prescribed order.

`PinCellNames` in `CreatePi()`: See above.

`totalfortransport` in `CreatePi()`: Use the total cross section instead of the transport cross section in computing $\mathbf{\Pi}$. If `stammlertransport` is used, then transport is undefined, and total *it* is transport.

`useTimsModTransmission` in `CreatePi()`: Uses an external function to compute the transmission probability for antipin cells.

`saveProbabilities` in `CreatePi()`: Saves collision probabilities to a MAT file.

`weightbydE` in `SpectralMatrix()`: When computing the fission source, weight the source terms by the width of the corresponding energy group.

`transportfortotal` in `SpectralMatrix()`: Uses the transport cross section in the place of the total cross section in the sink term of Equation 2.31.

`mode` in `Vbudsii()`: Chooses the mode in which to run VBUDSII .

B.5 Further Work

The following are ideas for the extension of VBUDSII.

Input validation This is done in various places, but should be more thorough and all validation should occur in `Preprocessor()`.

Group structure Currently, the user specifies the group structure in the `Param struct` as well as through the data library. The group structure should only be set through the data library. Ideally, the library is pre-made for two or three group structures, and the user can choose which is used.

Library management Currently, VBUDSII spends most of its time loading the cross section library into memory. This becomes less important when the code is used up for burnup simulations, as the library is only loaded once. However, as more materials are added, not all of the current library can be loaded into the memory at once. It is suggested that a more efficient scheme is used for loading the data, and for also exploiting the sparsity of most of this data to save memory. Additionally, $\bar{\mu}$ is tabulated for all the temperatures and background cross sections that the other data is stored for. However, $\bar{\mu}$ does not vary with background cross section. Some memory can be saved here.

Coupled data structures Right now, accessing cross sections in `Region` requires the `L.MTs` field to map from MT numbers to `struct` indices. It would be better to use the format implemented in VBUDSIIOOS in which MT is never used beyond the library. Another place where data structures are coupled to each other is through the fact that the user specifies the group structure in the `Param` `struct`, but this should really be specified only in `Lib`.

Region struct initialization Currently the `Region` `struct` is initialized through many loops. This should be done in an alternative way.

Descriptive variables Currently, the cross section interaction types are specified using the undescriptive ENDF MT numbers. These were put into use in a time when text space was sacred, but now it is possible to use descriptive names for each interaction. The VBUDSIIOO code uses a much better format for the `Results` `struct`, and this must be consulted before any future work is done.

Object oriented The use of highly nested `structs` obviates that VBUDSII should be an object oriented code. For example, there could exist a Region class which contains a list of Cell objects. Then, the Cell objects would contain a list of Nuclide objects, thereby describing its composition. This would alleviate the headache that is the `Results struct`.

Additional cell geometries `CreatePi()` can be written for other cell geometries, such as hexagonal lattices.

Fission spectrum Right now, one fission spectrum is used for all fissionable nuclides. However, there are slight differences in the fission spectrum between nuclides, as given by different Watt parameters. This could be accounted for, and would require a change in how the fission spectrum is incorporated. However, it is not expected that the accuracy of the code would change much by implementing this.

Diffusion and burnup modules Partial work has been done on both of these modules, but completing these modules would yield a fully functional reactor analysis code. This would require the inclusion of a burnable poison in the materials in order to control the criticality of the system.

Leakage A leakage term can be included in the integral transport to so that finite geometries can be modeled. This is done in (**?**).

(n,2n) reaction The (n,2n) interaction has been integrated into the code, but its use needs to be tested.

Multi-region The results of the homogeneity correction indicate that more work can be done in improving the code's ability to model a geometry of more than two cells.

B.6 Development Notes

In this section we briefly mention the issues that arose in attempting to match our results to those produced by MCNPX . These issues are related to obtaining the proper data rather than on errors in the method or theory. The issues are sorted by importance so that the first is the most important.

Transport correction Most theoretical texts only provide a cursory explanation of the transport correction. When tasked with developing a multigroup model, more information about the use of the transport correction is required. Stamm'ler and Abbate (1983) provides such information. He details that the transport correction is applied to the diagonal terms of the scattering kernel. Previously, the transport correction was only applied to the total cross section. This leads to cross sections that are inconsistent with themselves and results consequently suffer grave inaccuracies.

Bound hydrogen In a water-moderated system, it is well known that at thermal energies hydrogen cannot be modeled as a free gas. If using a nuclear data processing code like NJOY (MacFarlane and Muir, 1994), that code must be made aware that hydrogen data should be obtained for the case that it is bound in a water molecule. The scattering cross section for hydrogen bound in water is much higher at thermal energies than it is for free hydrogen. Additionally, the average scattering angle $\bar{\mu}$ departs from its constant value of 2/3 at thermal energies, and approaches zero. Much time was spent ensuring that such modified data was obtained. As an alternative, Schneider (2002) and Ozgener and Ozgener (2005) chose to use theoretical models for the hydrogen scattering cross section. Schneider (2002) still obtained $\bar{\mu}$ using NJOY , but Ozgener and Ozgener (2005) computed $\bar{\mu}$ directly from his theoretical model (the Nelkin kernel) using Equation 2.8.

Temperature The temperature of the reactor has a substantial effect on the shape and location of the thermal peak of the flux, and also affects the multiplication factor of the system through the Doppler broadening of resonances. Some time was spent ensuring that the libraries obtained from NJOY for all nuclides, but in particular for hydrogen, were obtained for the proper temperature.

Fission spectrum The fission spectrum used with the fission source affects the shape of the fast peak of the spectrum. It is important that when benchmarking the method against other codes, the same fission spectrum is used. In this work a Watt fission spectrum is used using parameters taken directly from the MCNPX manual (Pelowitz, 2011) for induced fission.

Inelastic scattering Inelastic scattering makes a non-negligible contribution to the scattering kernel at high incident energies for most nuclides other than hydrogen. The process of obtaining this data from NJOY is somewhat painful. Though NJOY suggests that the single `MT` number 4 can be used to obtain the total inelastic cross section, the user must actually request `MT` 51 through 91, which provide the individual contributions to inelastic scattering from each excited state, and sum the result.

Index

MCNPX , 120
NJOY , 17

Abstract, vi
Acknowledgments, v
angular flux, 9
Appendices, 86

background cross section, 52
bateman, 95
Bell correction factor, 40
Bibliography, 127
blackness, 32
Bondarenko, 52

Carlvik approximation, 40
Carlvik's two-term rational approximation, 56
cell pair, 43

Dancoff, 3
Dancoff factor, 56
Dedication, iv
diffusion, 95
dilution grid, 61
DRAGON, 2

eigenvalue, 28
equivalence relation, 54
escape cross section, 55

fission spectrum, 8
flat flux approximation, 24
four-cell transport, 43

geometric series, 35

homogeneity, 45

infinite dilution, 53, 61

leakage, 24, 26

mean chord length, 39
mean free path, 38, 46

narrow resonance (NR) approximation, 61

probability
  escape, 32
  transmission, 32

rational approximations, 38
reciprocity, 42

Sauer approximation, 40
scattering
  isotropic, 13
self-shielding, 95
self-shielding factor, 58
subgroup, 48

transmission probability, 42
transport
  integral, 24
  multigroup, 18
transport correction, 14, 31, 39, 41, 42, 119
  multigroup, 22
transport equation, 6

VBUDS, 30
VBUDSII, 93

Watt fission spectrum, 111
Wigner’s rational approximation, 39
WIMS, 2

Bibliography

Barratt, J. G., 2003. Calculation of the Dancoff Factor in Uniform Fuel Rod Arrays. Ph.D. thesis, Cornell University.

Bell, G. I., Glasstone, S., 1970. Nuclear Reactor Theory. Litton Educational Publishing, Inc., New York.

Bondarenko, I. I., 1964. Group Constants for Nuclear Reactor Calculations. Consultants Bureau Enterprises, Inc., New York.

Chiba, G., 2003. A Combined Method to Evaluate the Resonance Self Shielding Effect in Power Fast Reactor Fuel Assembly Calculation. Journal of Nuclear Science and Technology 40 (7), 537–543.

Cullen, D. E., 1974. Application of the Probability Table Method to Multigroup Calculations of Neutron Transport. Nuclear Science and Engineering 55, 387.

Dembia, C. L., Recktenwald, G. D., Deinert, M. R., 2012. Bondarenko method for obtaining group cross sections in a multi-region collision probability model.

Gopalakrishnan, V., Ganesan, S., 1998. Self-shielding and Energy Dependence of Dilution Cross-section in the Resolved Resonance Region. Annals of Nuclear Energy 25 (11), 839–857.

Hebert, A., 1997. Advances in the Development of a Subgroup Method for the Self-Shielding of Resonant Isotopes in Arbitrary Geometries. Nuclear Science and Engineering 126, 245–263.

Hébert, A., 2005. The Ribon Extended Self-Shielding Model. Nuclear Science and Engineering 151, 1–24.

Hébert, A., 2007. A Review of Legacy and Advanced Self-Shielding Models for Lattice Calculations. Nuclear Science and Engineering 155, 310–320.

Hébert, A., Marleau, G., 1991. Generalization of the Stamm'ler method for the self-shielding of resonant isotopes in arbitrary geometries. Nuclear Science and Engineering 108, 230–239.

Huang, S.-E., Wang, K., Yao, D., Aug. 2011. An Advanced Approach to Calculation of Neutron Resonance Sself-Shielding. Nuclear Engineering and Design 241 (8), 3051–3057.

Hwang, R. N., 1982. An Overview of Current Resonance Theory for Fast-Reactor Applications. Annals of Nuclear Energy 9, 31–44.

Joo, H. G., Kim, G. Y., Pogosbekyan, L., Jul. 2009. Subgroup Weight Generation Based on Shielded Pin-Cell Cross Section Conservation. Annals of Nuclear Energy 36 (7), 859–868.

Kidman, R. B., Schenter, R. E., Hardie, R. W., Little, W. W., 1972. The Shielding Factor Method of Generating Multigroup Cross Sections for Fast Reactor Analysis. Nuclear Science and Engineering 48, 189–201.

Lamarsh, J. R., 1965. Introduction to Nuclear Reactor Theory. Addison-Wesley Publishing Company, Inc., Reading.

Leszczynski, F., Aldama, D. L., Trkov, A., 2007. WIMS-D Library Update. Tech. rep., International Atomic Energy Agency, Austria.

MacFarlane, R. E., Muir, D. W., 1994. The NJOY Nuclear Data Processing System Version 91. Tech. rep., Los Alamos National Laboratory, Los Alamos.

Matavosian, R., 2007. Complex Problems Arising in the Collision Probability Theory for Neutron Transport. Ph.D. thesis, The University of Texas at Austin.

Nasr, M., Roushdy, H., 1991. Generalization of the Livolant-Jeanpierre method of resonance self-shielding. Annals of Nuclear Energy 19 (2), 59–63.

Ozgener, B., Ozgener, H., Jun. 2005. Multiregion, Multigroup Collision Probability Method with White Boundary Condition for Light Water Reactor Thermalization Calculations. Journal of Quantitative Spectroscopy and Radiative Transfer 92 (4), 417–445.

Pelowitz, D. P., 2011. MCNPX User's Manual Version 2.7.0. Tech. Rep. LA-CP-11-00438, Los Alamos National Laboratory.

Powney, D. J., Newton, T. D., 2004. Overview of The WIMS 9 Resonance Treatment. Tech. Rep. 1, ANSWERS.

Rowlands, J., 1999. LWR Pin Cell Benchmark Intercomparisons. Tech. Rep. September, OECD.

Saller, T., 2009. Resonance self shielding in collision probability models. Undergraduate thesis, The University of Texas at Austin.

Schneider, E., Deinert, M., Cady, K., Oct. 2006a. A Computationally Simple Model for Determining the Time Dependent Spectral Neutron Flux in a Nuclear Reactor Core. Journal of Nuclear Materials 357, 19–30.

Schneider, E. A., 2002. A Physical and Economic Model of the Nuclear Fuel Cycle. Ph.D. thesis, Cornell University.

Schneider, E. A., Barratt, J. G., Cady, K. B., Deinert, M. R., 2005. Visualization of Coupled Spectral and Burnup Calculations : an Intuition-building Tool. In: Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition.

Schneider, E. A., Deinert, M. R., Cady, K. B., 2006b. Burnup Simulations of_an Inert Matrix Fuel Using a Two Region Multigroup Reactor Physics Model. In: PHYSOR-2006, ANS Topical Meeting on Reactor Physics. Vancouver.

Stacey, W., 2001. Nuclear Reactor Physics. Wiley-Interscience.

Stamm'ler, R. J. J., Abbate, M. J., 1983. Methods of Steady-State Reactor Physics in Nuclear Design. Academic Press, London.

Yamamoto, A., 2008. Evaluation of Background Cross Section for Heterogeneous and Complicated Geometry by the Enhanced Neutron Current Method. Journal of Nuclear Science and Technology 45 (12), 1287–1292.

Vita

Christopher Lee Dembia was born in Flushing, New York on November 30, 1989, the son of Robert Dembia and Suzy Lee Dembia. He received a Bachelor of Science degree in Engineering from Cornell University in 2012. His interest in nuclear energy originated from the advice that his high school physics teacher Mr. Kundmueller gave to him at the beginning of 2008, that he should do something about global warming. During the summer of 2008, while on the job at the New York State Department of Transportation near his home, he would read news articles about next generation reactors and peruse course catalogs for graduate nuclear engineering programs.

Two summers later, in 2010, he was an inaugural member of the Toshiba-Westinghouse Fellowship Program at the Pennsylvania State University, where he performed experimental work on the hydriding of zirconium hydride fuel rods under Ms. Melissa Marshall and Dr. Arthur Motta. The program's greatest contribution to him was through its focus on creating, preparing for, and giving technical presentations.

He spent his last year at Cornell University working with Dr. Bingham Cady and Dr. Geoff Recktenwald, through whom he was introduced to Dr. Mark Deinert at the 2011 Winter Meeting of the American Nuclear Society. He began his studies at the University of Texas at Austin in August 2011.

He finds the neutron transport equation to hold somewhat of a special place in the family of conservation equations. He enjoys the nature of neutronics problems, particularly the fact that the transport equation can describe the life of a neutron, starting with its birth with $\bar{\nu} - 1$ siblings, continuing through the ups an downs of the resonance years, and then life slows down until the chance of death becomes great and new life is made out of the old. And yet other neutrons experience a different lifestyle, one that is so quick that the neutron cannot distinguish one place it has been from another, that all is blurred, and life ends abruptly.

Permanent address: 6 Mineola Ave
Roslyn, NY 11576

This thesis was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.