DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

Exnihilo Transport Methods Manual

Tom Evans

Greg Davidson

Josh Jarrell

Steven Hamilton

Seth Johnson

Tara Pandya

Radiation Transport Group, Reactor and Nuclear Systems Division, Oak Ridge National Laboratory, One Bethel Valley Rd, Oak Ridge, TN 37831

Contents

List of Figures v

List of Tables vii

Chapter 1. Introduction 1

Chapter 2. Transport Basics 3
1. Basic Transport Definitions 3
2. Transport Equations 4

Part 1. Deterministic Methods 9

Chapter 3. Multigroup Energy Approximation 11

Chapter 4. Collocation Angular Methods 15
1. Angular Discretization of Scattering Terms 15
2. Angular Discretization of External Sources 18
3. Angular Collocation Discretization 19
4. Operator Form of the Discrete Ordinates Equation 20

Chapter 5. Spectral Angular Methods 23
1. Spherical Harmonics 23
2. Simplified Spherical Harmonics 25
3. Eigenvalue Form of the SP_N Equations 29
4. Adjoint Form of the SP_N Equations 30

Chapter 6. Spatial Discretizations of the S_N Equations 33
1. Cell-Balance Schemes 34
2. Finite Element Schemes 35
3. Step Characteristics 39

Chapter 7. Spatial Discretizations of the SP_N Equations 41
1. Finite Volume Discretization of the SP_N Equations 41

Chapter 8. Transport Solution Methods 47
1. Eigenvalue Solvers for the SP_N Equations 47

Part 2. Stochastic Methods 51

Chapter 9. The Monte Carlo Method 53
1. Probability Distribution Functions 53
2. Distribution Sampling 55
3. Monte Carlo Errors 57

Chapter 10. Parallel Monte Carlo Methods 63

1. Multiple Set Overlapping Domain Decomposition Algorithm 63
2. Fission Site Rebalance 65
3. Domain-Decomposed Solving Algorithms 68

Part 3. Application-Specific Methods 75

Chapter 11. Methods for Reactor Analysis 77
1. Mesh Generation 77
2. Cross Section Generation 78

Bibliography 81

Appendix A. Spherical Harmonics Expansions 83

Appendix B. Mathematical Properties and Identities 87

List of Figures

2.1 Coordinate system used in Denovo. 4

3.1 Multigroup energy grid used in Denovo. 11

4.1 2D coordinate system used in Denovo. Symmetry is defined about the XY plane. 17

6.1 General mesh cell in Denovo used to derive discrete spatial equations. The adjacent cell points are given using the notation $N \rightarrow +x$, $F \rightarrow -x$, $L \rightarrow -y$, $R \rightarrow +y$, $B \rightarrow -z$, and $T \rightarrow +z$. 33

7.1 Three-dimensional, Cartesian mesh cell. 41

7.2 SP_3 matrix sparsity pattern for a $2 \times 2 \times 4$ spatial grid and 4 energy groups. 45

7.3 Sparsity of 56 group $\mathbf{\Sigma}_n$ matrix. 46

9.1 The probability density function $f(x)$ along with a more simple PDF, $g(x)$. The ratio of $g(x)$ and $f(x)$, $h(x)$ is plotted in the lower graph. The upper limit of $h(x)$ is h_{max}. 56

10.1 Example of MSOD geometry. In this example, $N_s = 4$ and the entire geometry is represented within each set. 64

10.2 Illustration of a mesh tally reduction with a 5×5 tally mesh and a 2×1 boundary mesh yielding 2 blocks per set. In (a) a target mesh is defined for each block. In (b) a source mesh is defined for each block. The boundary mesh overlap regions are indicated by the lightly shaded areas. In (c) a global scatter is performed from the source mesh to the target mesh. In (d) an intra-set reduction is performed to get a final, global tally on each block. 65

10.3 Rebalance boundaries across sets. 67

10.4 Illustration of block-to-block communication between sets during fission bank rebalance. Communication is constrained to equivalent blocks in adjacent sets. 68

11.1 PWR pin-cell meshed with an 8×8 Cartesian grid. 77

List of Tables

8.1 Classification of eigensolvers by linear system solution required 49

CHAPTER 1

Introduction

Exnihilo is a transport code system that contains both deterministic and stochastic (Monte Carlo) solvers. The deterministic solvers are contained in the Denovo [**1**] package while the Monte Carlo solvers are implemented in the Shift package. This document explains the mathematical and numerical methods employed within Exnihilo. It is not a users guide, nor is it a code developer's manual. Naturally, there is cross-over between the code implmentation and mathematical description of any given method, and these are included for the purpose of greater understanding where appropriate. The purpose of this document is to formally describe the equations that are solved by the various packages inside of Exnihilo. In general, linkages are not made between a given numerical method and the Exnihilo code base. This is largely due to the fact that the Exnihilo code base changes far more rapidly than the actual methods that it is solving. Thus, this documentation would go rapidly out of date if cross-references were made between the methods and the code that implements them. Instead, developers are encouraged to look at the Doxygen-produced code documentation that explains what each translation unit in the code is doing. Where necessary we will make distinctions between methods that are implemented in Exnihilo and methods that are documented to provide better understanding of the broader field and provide context for the choosen methods in Exnihilo.

This document is organized in three parts. Part 1 describes the deterministic methods implemented in the Exnihilo Denovo package. Part 2 describes the Monte Carlo (stochastic) methods implemented in the Exnihilo Shift package. Part 3 describes application-specific numerical algorithms (e.g. mesh generation) that are implemented in Exnihilo front-end (Omnibus, Insilico) and support packages (Transcore, Physica, Geometria). The fundamental model equations of radiation transport that are solved within Exnihilo are introduced in Chap. 2.

CHAPTER 2

Transport Basics

This chapter defines the basic equations and unknowns in radiation transport. We should emphasize that Exnihilo is primarily a linear, neutral particle transport system. Thus, charged particle transport methods (electrons, ions, etc) and nonlinear transport (radiative transfer) are not widely discussed here.

1. Basic Transport Definitions

The principal unknown in radiation transport calculations is the *angular flux*,

$$\psi \equiv \psi(\mathbf{r}, \mathbf{\Omega}, E, t) \,, \tag{2.1}$$

which, in 3D, is a function of 7-independent variables, (x, y, z) in space, (θ, φ) in angle, E in energy, and t in time. The *particle density*, N, is related to the angular flux via,

$$\psi(\mathbf{r}, \mathbf{\Omega}, E, t) = vN(\mathbf{r}, \mathbf{\Omega}, E, t) \,, \tag{2.2}$$

where v is the particle velocity. The zeroth angular moment of the angular flux is the *scalar flux*,

$$\phi(\mathbf{r}, E, t) = \int_{4\pi} \psi(\mathbf{r}, \mathbf{\Omega}, E, t) \, d\mathbf{\Omega} \,. \tag{2.3}$$

The first angular moment of the angular flux is the *current*,

$$\mathbf{J}(\mathbf{r}, E, t) = \int_{4\pi} \hat{\mathbf{\Omega}} \psi(\mathbf{r}, \mathbf{\Omega}, E, t) \, d\mathbf{\Omega} \,. \tag{2.4}$$

The second angular moment of the angular flux is the *pressure* tensor,

$$\mathbf{P}(\mathbf{r}, E, t) = \int_{4\pi} \hat{\mathbf{\Omega}}\hat{\mathbf{\Omega}} \psi(\mathbf{r}, \mathbf{\Omega}, E, t) \, d\mathbf{\Omega} \,. \tag{2.5}$$

Units are defined in the CGS system such that $\mathbf{r} = (x, y, z)$ is in cm, $\mathbf{\Omega} = (\theta, \varphi)$ is in str, and E is in eV representing space, angle, and energy respectively. The coordinate system used in Exnihilo is illustrated in Fig. 2.1. The unit vector of particle direction, $\hat{\mathbf{\Omega}}$, is defined in this coordinate system,

$$\hat{\mathbf{\Omega}} = \mu\hat{\mathbf{e}}_x + \eta\hat{\mathbf{e}}_y + \xi\hat{\mathbf{e}}_z \,, \tag{2.6}$$

where

$$\xi = \cos\theta \,, \tag{2.7a}$$

$$\mu = \sqrt{1 - \xi^2} \cos\varphi \,, \tag{2.7b}$$

$$\eta = \sqrt{1 - \xi^2} \sin\varphi \,. \tag{2.7c}$$

The normalization of $\mathbf{\Omega}$ in Exnihilo is such that

$$\int_{4\pi} d\mathbf{\Omega} = \int_0^{2\pi} \int_0^{\pi} \sin\theta \, d\theta d\varphi = \int_0^{2\pi} \int_{-1}^{1} d\xi d\varphi = 4\pi \,. \tag{2.8}$$

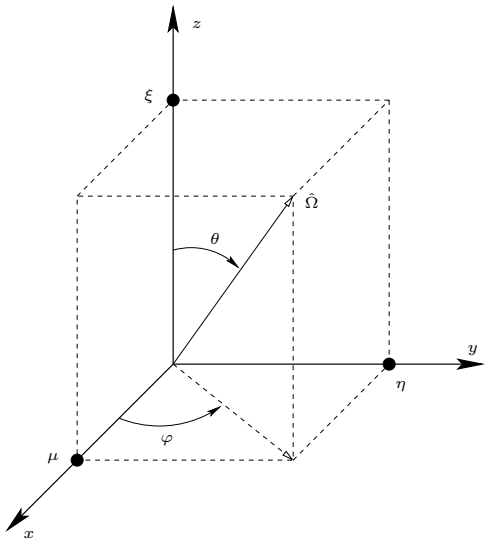


FIGURE 2.1. Coordinate system used in Denovo.

2. Transport Equations

This text describes numerical methods to solve radiation transport problems. The particular focus is on linear, neutral particle physics, which is sufficient to describe the transport of neutrons and photons (gammas) in most problems of interest in nuclear engineering and technology. Nonlinear radiative transfer and charged particle transport problems are beyond the scope of this text. Furthermore, because the focus is on numerical methods, we do not spend time deriving the transport equation. The interested reader is directed to Refs. [**2**–**4**] for detailed derivations of the transport equation.

The steady-state Boltzmann transport equation solved in Exnihilo has two forms: fixed-source and eigenvalue. The fixed-source form is

$$\hat{\mathbf{\Omega}} \cdot \nabla\psi(\mathbf{r},\mathbf{\Omega},E) + \sigma(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) = \int_0^\infty \int_{4\pi} \sigma_{\mathrm{s}}(\mathbf{r},\mathbf{\Omega}'\to\mathbf{\Omega},E'\to E)\psi(\mathbf{r},\mathbf{\Omega}',E')\,d\mathbf{\Omega}'dE' + q_e(\mathbf{r},\mathbf{\Omega},E)\,. \quad (2.9)$$

The eigenvalue form is

$$\hat{\mathbf{\Omega}} \cdot \nabla\psi(\mathbf{r},\mathbf{\Omega},E) + \sigma(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) = \int_0^\infty \int_{4\pi} \sigma_{\mathrm{s}}(\mathbf{r},\mathbf{\Omega}'\to\mathbf{\Omega},E'\to E)\psi(\mathbf{r},\mathbf{\Omega}',E')\,d\mathbf{\Omega}'dE' + \frac{1}{k}\frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty \int_{4\pi} \nu\sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},\mathbf{\Omega}',E')\,d\mathbf{\Omega}dE'\,. \quad (2.10)$$

In Eqs. 2.9 and (2.10),

σ	total interaction cross section in cm^{-1}
σ_{s}	double differential macroscopic scattering cross section in cm^{-1}
ν	neutrons produced per fission event
χ	fission spectrum
σ_{f}	macroscopic fission cross section in cm^{-1}

The time-dependent form of the Boltzmann equation is

$$\frac{1}{v}\frac{\partial\psi(\mathbf{r},\mathbf{\Omega},E,t)}{\partial t}+\hat{\mathbf{\Omega}}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E,t)+\sigma(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E,t)$$
$$=\int_0^\infty\int_{4\pi}\sigma_{\mathrm{s}}(\mathbf{r},\mathbf{\Omega}'\rightarrow\mathbf{\Omega},E'\rightarrow E)\psi(\mathbf{r},\mathbf{\Omega}',E',t)\,d\mathbf{\Omega}'dE'$$
$$+\frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},\mathbf{\Omega}',E',t)\,d\mathbf{\Omega}dE'\,. \quad (2.11)$$

Here, v is the particle velocity ($v=c$, the speed of light, for photons). Equations (2.9) and Eq. (2.11) obey the boundary condition

$$\psi(\mathbf{r},\mathbf{\Omega},E)=\Gamma\,,\quad \mathbf{r}\in\partial V\,,\quad \hat{\mathbf{\Omega}}\cdot\mathbf{n}<0\,, \quad (2.12)$$

which defines the incoming flux on all problem boundaries with outgoing normal $\mathbf{n}$. The eigenvalue equation must have $\Gamma=0$ on all problem boundaries (no sources can exist in eigenvalue problems). All three equations can admit various reflecting boundary conditions in which

$$\Gamma=\psi(\mathbf{r},-\mathbf{\Omega},E)\,,\quad \mathbf{r}\in\partial V\,. \quad (2.13)$$

The integro-differential nature of the transport equation combined with the large dimensionality makes transport problems particularly difficult to solve. Nonetheless, despite its complex form, the transport equation is really a simple expression of particle balance, loss = production. This can be observed by integrating Eq. (2.9) over space, angle, and energy and applying Eqs. (2.4) and (2.5),

$$\int_V[\nabla\cdot\mathbf{J}(\mathbf{r})+R-S-q(\mathbf{r})]=0\,. \quad (2.14)$$

Because $\sigma(E)\phi(E)dE$ is the total number of reactions in dE, integrating over all energies must give the total reaction rate,

$$R=\int_E\sigma(E)\phi(E)\,dE=\sigma\phi\,. \quad (2.15)$$

Similarly, the term

$$\int_{E'}\int_{\Omega'}\sigma_{\mathrm{s}}(\mathbf{\Omega}'\rightarrow\mathbf{\Omega},E'\rightarrow E)\psi(\mathbf{\Omega}',E')\,d\mathbf{\Omega}'dE'd\mathbf{\Omega}dE \quad (2.16)$$

is the total number of particles emitted in $dEd\mathbf{\Omega}$ due to scattering. Thus, integrating over all energies and angles gives the total scattering rate

$$S=\int_E\int_\Omega\left[\int_0^\infty\int_{4\pi}\sigma_{\mathrm{s}}(\mathbf{\Omega}'\rightarrow\mathbf{\Omega},E'\rightarrow E)\psi(\mathbf{\Omega}',E')\,d\mathbf{\Omega}'dE'\right]d\mathbf{\Omega}dE=\sigma_{\mathrm{s}}\phi\,. \quad (2.17)$$

Using these definitions and recognizing that the integrand is required to vanish gives the particle balance (conservation) equation,

$$\nabla\cdot\mathbf{J}(\mathbf{r})+\big(\sigma(\mathbf{r})-\sigma_{\mathrm{s}}(\mathbf{r})\big)\phi(\mathbf{r})=q(\mathbf{r})\,. \quad (2.18)$$

2.1. Operator Notation. In much of later chapters in this document (especially Chap. 8) we will find it convenient to express the transport equation using operator notation. Here, we define the continuous, fixed-source and eigenvalue forms of the transport equation in operator notation:

$$\hat{\mathrm{L}}\psi=\hat{\mathrm{S}}\psi+\hat{\mathrm{Q}}\qquad\text{(fixed-source)}\,, \quad (2.19)$$

$$\hat{\mathrm{L}}\psi=\hat{\mathrm{S}}\psi+\frac{1}{k}\hat{\chi}\hat{\mathrm{F}}\psi\quad\text{(eigenvalue)}\,. \quad (2.20)$$

with

$$\hat{\mathrm{L}} = \hat{\mathbf{\Omega}} \cdot \nabla + \sigma_{\mathrm{t}} \,, \tag{2.21}$$

$$\hat{\mathrm{S}} = \iint d\mathbf{\Omega}' dE' \, \sigma_{\mathrm{s}}(\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}', E' \to E) \,, \tag{2.22}$$

$$\hat{\mathrm{Q}} = q_e(\mathbf{r}, \mathbf{\Omega}, E) \,, \tag{2.23}$$

$$\hat{\mathrm{F}} = \iint d\mathbf{\Omega}' dE' \, \nu\sigma_{\mathrm{f}}(E') \,, \tag{2.24}$$

and, following Eq. (2.3),

$$\phi = \hat{\mathrm{D}}\psi \,. \tag{2.25}$$

The eigenvalue equation can now be easily manipulated into the standard form of a generalized eigenvalue equation,

$$(\hat{\mathrm{L}} - \hat{\mathrm{S}})\psi = \frac{1}{k}\hat{\chi}\hat{\mathrm{F}}\psi \,. \tag{2.26}$$

To form a standard (homogeneous) eigenvalue equation, We define the fission source

$$\Gamma = \hat{\mathrm{F}}\psi \,, \tag{2.27}$$

such that

$$\begin{aligned} \Gamma &= \iint d\mathbf{\Omega}' dE' \, \nu\sigma_{\mathrm{f}}(E')\psi(\mathbf{\Omega}, E') \\ &= \int dE' \, \nu\sigma_{\mathrm{f}}(E')\phi(E') \end{aligned} \tag{2.28}$$

Finally, rearranging terms:

$$k\psi = (\hat{\mathrm{L}} - \hat{\mathrm{S}})^{-1}\hat{\chi}\hat{\mathrm{F}}\psi \,, \tag{2.29}$$

multiplying by $\hat{\mathrm{F}}$ gives

$$\hat{\mathrm{A}}\Gamma = k\Gamma \,, \tag{2.30}$$

where the $\hat{\mathrm{A}}$ is the *fission operator* (or *fission matrix* in discrete space),

$$\hat{\mathrm{A}} = \hat{\mathrm{F}}(\hat{\mathrm{L}} - \hat{\mathrm{S}})^{-1}\hat{\chi} \tag{2.31}$$

2.2. Adjoint Form of the Transport Equation. The solution to the adjoint form of the transport equation is useful in hybrid Monte Carlo/deterministic transport methods and sensitivity analysis. Equations (2.9) and (2.10) are not self-ajoint; the adjoint form of Eq. (2.9) is

$$\begin{aligned} &- \hat{\mathbf{\Omega}} \cdot \nabla\psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) + \sigma(\mathbf{r}, E)\psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) \\ &\qquad = \int_0^\infty \int_{4\pi} \sigma_{\mathrm{s}}(\mathbf{r}, \hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}', E \to E')\psi^\dagger(\mathbf{r}, \mathbf{\Omega}', E') \, d\mathbf{\Omega}' dE' + q^\dagger(\mathbf{r}, \mathbf{\Omega}, E) \,. \end{aligned} \tag{2.32}$$

Similarly, the adjoint form of the eigenvalue equation is

$$\begin{aligned} - \hat{\mathbf{\Omega}} \cdot \nabla\psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) + \sigma(\mathbf{r}, E)\psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) &= \int_0^\infty \int_{4\pi} \sigma_{\mathrm{s}}(\mathbf{r}, \hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}', E \to E')\psi^\dagger(\mathbf{r}, \mathbf{\Omega}', E') \, d\mathbf{\Omega}' dE' \\ &\quad + \frac{1}{k^\dagger} \frac{\nu\sigma_{\mathrm{f}}(\mathbf{r}, E)}{4\pi} \int_0^\infty \int_{4\pi} \chi(\mathbf{r}, E')\psi^\dagger(\mathbf{r}, \mathbf{\Omega}', E') \, d\mathbf{\Omega} dE' \,. \end{aligned} \tag{2.33}$$

The adjoint equation obeys the boundary condition

$$\psi^\dagger(\mathbf{r}, \mathbf{\Omega}, E) = 0 \,, \quad \mathbf{r} \in \partial\Gamma \,, \quad \hat{\mathbf{\Omega}} \cdot \hat{n} > 0 \,, \tag{2.34}$$

which defines the *outgoing* flux on all problem boundaries. The adjoint source is defined

$$q^\dagger(\mathbf{r}, \mathbf{\Omega}, E) = \frac{1}{4\pi}\Sigma(\mathbf{r}, E)\delta(\mathbf{r} - \mathbf{r}')\delta(E - E') \,, \tag{2.35}$$

where Σ is a normalized response function in a detector region defined in the phase-space of $\mathbf{r}'$ and E'.

Writing the adjoint eigenvalue equation in operator form gives

$$\hat{\mathrm{A}}^\dagger \Gamma^\dagger = k^\dagger \Gamma^\dagger \,, \tag{2.36}$$

where

$$\hat{\mathrm{A}}^\dagger = \hat{\chi}^\dagger (\hat{\mathrm{L}}^\dagger - \hat{\mathrm{S}}^\dagger)^{-1} \hat{\mathrm{F}}^\dagger \,, \tag{2.37}$$

and

$$\Gamma^\dagger = \hat{\chi}^\dagger \psi^\dagger \,. \tag{2.38}$$

Now, the inner product of two vectors is

$$(x, y) = \int x^* y \, dp \,, \tag{2.39}$$

where * denotes complex conjugation. Additionally, for a given linear operator $\hat{\mathrm{A}}$,

$$(x, \hat{\mathrm{A}} y) = (\hat{\mathrm{A}} y, x)^* = (y, \hat{\mathrm{A}}^\dagger x)^* = (\hat{\mathrm{A}}^\dagger x, y) \,. \tag{2.40}$$

The forward and adjoint eigenvalue equations are,

$$\hat{\mathrm{A}} \Gamma = k \Gamma \,, \tag{2.30}$$

$$\hat{\mathrm{A}}^\dagger \Gamma^\dagger = k^\dagger \Gamma^\dagger \,. \tag{2.36}$$

Operating on the left-side of Eq. (2.30) by $\Gamma^\dagger$ and operating on the right-side of Eq. (2.36) by Γ and taking the inner-product gives

$$(\Gamma^\dagger \hat{\mathrm{A}} \Gamma) = k (\Gamma^\dagger, \Gamma) \,, \tag{2.41}$$

$$(\hat{\mathrm{A}}^\dagger \Gamma^\dagger, \Gamma) = k^\dagger (\Gamma^\dagger, \Gamma) \,. \tag{2.42}$$

From Eq. (2.40) subtracting these two equations yields

$$0 = (k - k^\dagger)(\Gamma^\dagger, \Gamma) \,. \tag{2.43}$$

For common eigenpair modes, the inner product $(\Gamma^\dagger, \Gamma) \neq 0$ except in trivial cases where $\Gamma = \Gamma^\dagger = 0$; thus, we have

$$k = k^\dagger \,. \tag{2.44}$$

2.3. Diffusion Approximation. When the angular flux is linearly anisotropic, the transport equation limits to a diffusion equation. To show this, we will derive the diffusion equation using two approaches: a weighted residual (Galerkin) method and through asymptotic expansion. For both derivations we consider the 1D, monoenergetic form of Eq. (2.9) with isotropic scattering,

$$\mu \frac{\partial \psi}{\partial x} + \sigma \psi = \frac{\sigma_\mathrm{s}}{4\pi} \phi + \frac{q}{4\pi} \,, \tag{2.45}$$

where $\mu = \cos\theta$. First, we consider a weighted residual method in which the angular flux is expanded in the basis $\{1, \mu\}$,

$$\psi = \phi_0 + \mu \phi_1 \,. \tag{2.46}$$

The weighted-residual is formed by applying a weighting factor, w_n, to Eq. (2.45) and integrating over angle,

$$2\pi \int_{-1}^{1} w_n \Big[\mu \frac{\partial \psi}{\partial x} + \sigma \psi - \frac{\sigma_\mathrm{s}}{4\pi} \phi - \frac{q}{4\pi} \Big] d\mu = 0 \,, \quad n = 1, 2 \,. \tag{2.47}$$

Here, we apply a Galerkin weighting method in which the weights are equal to the basis functions,

$$w_0 = 1 \,, \quad w_1 = \mu \,.$$

Substituting the expansion of ψ and using these weights gives two equations for the unknowns $\{\phi_0, \phi_1\}$,

$$2\pi \int_{-1}^{1} \Big[\mu \frac{\partial}{\partial x}(\phi_0 + \mu\phi_1) + \sigma(\phi_0 + \mu\phi_1) - \frac{\sigma_s}{4\pi}\phi - \frac{q}{4\pi}\Big] d\mu = 0 \,, \tag{2.48}$$

$$2\pi \int_{-1}^{1} \mu \Big[\mu \frac{\partial}{\partial x}(\phi_0 + \mu\phi_1) + \sigma(\phi_0 + \mu\phi_1) - \frac{\sigma_s}{4\pi}\phi - \frac{q}{4\pi}\Big] d\mu = 0 \,. \tag{2.49}$$

Carrying out the integrations gives

$$\frac{4\pi}{3}\frac{\partial \phi_1}{\partial x} + 4\pi\sigma\phi_0 = \sigma_s\phi + q \,, \tag{2.50}$$

$$\frac{\partial \phi_0}{\partial x} + \sigma\phi_1 = 0 \,. \tag{2.51}$$

Solving for ϕ_1 using the second equation and substituting into the first equation, we obtain

$$-\frac{\partial}{\partial x}\frac{4\pi}{3\sigma}\frac{\partial \phi_0}{\partial x} + 4\pi\sigma\phi_0 = \sigma_s\phi + q \,. \tag{2.52}$$

The relationship between ϕ_0 and ϕ_1 can be established by Eq. (2.3),

$$\phi = 2\pi \int_{-1}^{1} (\phi_0 + \mu\phi_1)\, d\mu = 4\pi\phi_0 \,. \tag{2.53}$$

Combining the previous two equations gives the correct diffusion equation,

$$-\frac{\partial}{\partial x}\frac{1}{3\sigma}\frac{\partial \phi}{\partial x} + (\sigma - \sigma_s)\phi = q \,. \tag{2.54}$$

Part 1

Deterministic Methods

CHAPTER 3

Multigroup Energy Approximation

All of the deterministic methods in Exnihilo use the *multigroup* energy discretization. Details on the basics of the multigroup method can be found in Ref. [**2**]. To begin the multigroup energy approximation we first write the fixed-source transport equation (2.9) while suppressing dependencies on $\mathbf{r}$ and $\mathbf{\Omega}$,

$$\hat{\mathbf{\Omega}} \cdot \nabla\psi(E) + \sigma(E)\psi(E) = \int_0^\infty \sigma_{\mathrm{s}}(E' \to E)\psi(E')\, dE' + q_e(E')\,. \tag{3.1}$$

First, we define a discrete energy grid as shown in Fig. 3.1. Now we expand ψ as follows:

$$\psi(E) = \sum_{k=0}^{G} \psi_k B_k(E)\,, \tag{3.2}$$

where $B_k(E)$ obeys the following properties

$$B_k(E) = 0 \quad \forall \quad E \notin [E_{k+1}, E_k]\,, \tag{3.3}$$

$$\int_{E_{k+1}}^{E_k} B_k(E)\, dE = 1\,. \tag{3.4}$$

Integrating Eq. (3.2) over group g gives

$$\begin{aligned} \int_{E_{g+1}}^{E_g} \psi(E)\, dE &= \sum_{k=0}^{G} \psi_k \int_{E_{g+1}}^{E_g} B_k(E)\, dE \\ &= \psi_g\,. \end{aligned} \tag{3.5}$$

The weight-function for each group is defined

$$W_g(E) = \begin{cases} 0 & E \notin [E_{g+1}, E_g]\,, \\ 1 & E \in [E_{g+1}, E_g]\,. \end{cases} \tag{3.6}$$

Now, expand $\psi(E)$ in Eq. (3.1),

$$\begin{aligned} &\hat{\mathbf{\Omega}} \cdot \nabla\Big(\sum_{k=0}^{G} \psi_k B_k(E)\Big) + \sigma(E)\Big(\sum_{k=0}^{G} \psi_k B_k(E)\Big) \\ &\qquad = \int_0^\infty \sigma_{\mathrm{s}}(E' \to E)\Big(\sum_{g'=0}^{G} \psi_{g'} B_{g'}(E')\Big)\, dE' + \Big(\sum_{k=0}^{G} q_{ek} B_k(E)\Big)\,. \end{aligned} \tag{3.7}$$

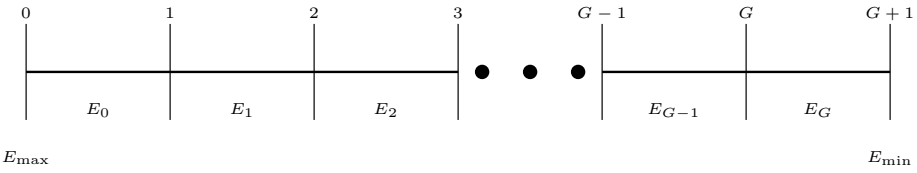


FIGURE 3.1. Multigroup energy grid used in Denovo.

Next, a weighted-residual expression of the transport equation is obtained by multiplying by $W_g(E)$ and integrating over all energies:

$$\int_0^\infty W_g(E)\hat{\mathbf{\Omega}}\cdot\nabla\Big(\sum_{k=0}^{G}\psi_k B_k(E)\Big)\,dE+\int_0^\infty W_g(E)\sigma(E)\Big(\sum_{k=0}^{G}\psi_k B_k(E)\Big)\,dE$$
$$=\int_0^\infty W_g(E)\int_0^\infty \sigma_{\mathrm{s}}(E'\to E)\Big(\sum_{g'=0}^{G}\psi_{g'}B_{g'}(E')\Big)\,dE'dE+\int_0^\infty W_g(E)\Big(\sum_{k=0}^{G}q_{ek}B_k(E)\Big)\,dE\,. \quad (3.8)$$

The definition of the weight function, W_g, converts integrals over all energies into integrals over group g,

$$\int_0^\infty W_g(E)\,dE=\int_{E_{g+1}}^{E_g}dE\,. \quad (3.9)$$

Applying this expression, we proceed to evaluate each term in Eq. (3.8).

Rearranging the sum and integral operators, the streaming term in Eq. (3.8) is

$$\hat{\mathbf{\Omega}}\cdot\nabla\Big(\sum_{k=0}^{G}\psi_k\int_{E_{g+1}}^{E_g}B_k(E)\,dE\Big)=\hat{\mathbf{\Omega}}\cdot\nabla\psi_g\,, \quad (3.10)$$

where the group flux, ψ_g, is defined in Eq. (3.5). Similarly, the source term can be evaluated

$$\sum_{k=0}^{G}q_{ek}\int_{E_{g+1}}^{E_g}B_k(E)\,dE=q_{eg}\,. \quad (3.11)$$

The collision term is becomes

$$\sum_{k=0}^{G}\psi_k\int_{E_{g+1}}^{E_g}\sigma(E)B_k(E)\,dE=\sigma_g\psi_g\,, \quad (3.12)$$

where

$$\sigma_g=\int_{E_{g+1}}^{E_g}\sigma(E)B_g(E)\,dE\,. \quad (3.13)$$

The scattering term is slightly more complicated. Rearranging terms gives

$$\sum_{g'=0}^{G}\psi_{g'}\int_{E_{g+1}}^{E_g}\int_0^\infty\sigma_{\mathrm{s}}(E'\to E)B_{g'}(E')\,dE'dE=\sum_{g'=0}^{G}\psi_{g'}\int_{E_{g+1}}^{E_g}\int_{E_{g'+1}}^{E_{g'}}\sigma_{\mathrm{s}}(E'\to E)B_{g'}(E')\,dE'dE\,, \quad (3.14)$$

where we have utilized the definition of $B_g(E)$ to recognize that the integral over all energies, E', will be zero everywhere except in group g'. The scattering integral becomes

$$\sum_{g'=0}^{G}\psi_{g'}\int_{E_{g+1}}^{E_g}\int_{E_{g'+1}}^{E_{g'}}\sigma_{\mathrm{s}}(E'\to E)B_{g'}(E')\,dE'dE=\sum_{g'=0}^{G}\sigma_{\mathrm{s}gg'}\psi_{g'}\,, \quad (3.15)$$

where

$$\sigma_{\mathrm{s}\,g'\to g}=\sigma_{\mathrm{s}gg'}=\int_{E_{g+1}}^{E_g}\int_{E_{g'+1}}^{E_{g'}}\sigma_{\mathrm{s}}(E'\to E)B_{g'}(E')\,dE'dE\,. \quad (3.16)$$

Putting everything together and including space/angle dependencies, the multigroup transport equation is

$$\hat{\mathbf{\Omega}}\cdot\nabla\psi^g(\mathbf{r},\mathbf{\Omega})+\sigma^g(\mathbf{r})\psi^g(\mathbf{r},\mathbf{\Omega})=\sum_{g'=0}^{G}\int_{4\pi}\sigma_{\mathrm{s}}^{gg'}(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega})\psi^{g'}(\mathbf{r},\mathbf{\Omega}')\,d\mathbf{\Omega}'+q_e^g(\mathbf{r},\mathbf{\Omega})\,, \quad (3.17)$$

where we have written the group indices as superscripts to provide additional clarity when we perform discretizations in angle and space. As an additional note, the energy-integrated angular flux is defined in the multigroup approximation,

$$\psi = \int_0^\infty \psi(E)\, dE = \sum_{g=0}^{G} \psi_g \int_0^\infty B_g(E)\, dE = \sum_{g=0}^{G} \psi_g \,. \tag{3.18}$$

CHAPTER 4

Collocation Angular Methods

Many deterministic transport methods treat the angular terms on the left-hand side of Eq. (3.17) using a finite-element collocation method. Foremost among these are discrete ordinates (S_N), method of characteristics (MOC), and short characteristics schemes. In each of these methods, the right-hand side scattering kernel is expanded using Spherical Harmonics. A finite element collocation discretization is then applied to the angular terms. The expansion of the scattering and source terms are described in §§ 1–2. The angular collocation scheme is explained in § 3.

1. Angular Discretization of Scattering Terms

Equation (3.17) is still an integro-differential equation in which the principal unknown, ψ, is present in the scattering integral. Additionally, the multigroup scattering cross section is a function of angle. Accordingly, we need some method for dealing with the angular dependence of the scattering source. We tackle this problem by recognizing that σ_{s} is only a function of the cosine between the incoming and outgoing angles (the polar scattering angle assuming azimuthal symmetry). Therefore, we can expand the scattering cross section in Legendre polynomials,

$$\sigma_{\mathrm{s}}^{gg'}(\mathbf{\Omega}'\cdot\mathbf{\Omega}) = \sum_{l=0}^{N} \frac{2l+1}{4\pi} P_l(\mathbf{\Omega}'\cdot\mathbf{\Omega})\sigma_{\mathrm{s}\,l}^{gg'} \ . \tag{4.1}$$

Integrating over all angles we calculate the total scattering cross section as follows:

$$\begin{aligned}
\sigma_{\mathrm{s}}^{gg'} &= \int_{4\pi} \sigma_{\mathrm{s}}^{gg'}(\mathbf{\Omega}'\cdot\mathbf{\Omega})\, d\mathbf{\Omega} \equiv 2\pi \int_{-1}^{1} \sigma_{\mathrm{s}}^{gg'}(\mu_o)\, d\mu_o \\
&= \sum_{l=0}^{N} \frac{2l+1}{2} \sigma_{\mathrm{s}\,l}^{gg'} \int_{-1}^{1} P_0(\mu_o) P_l(\mu_o)\, d\mu_o \\
&= \sigma_{\mathrm{s}\,0}^{gg'} \ ,
\end{aligned} \tag{4.2}$$

where we have used the orthogonality of Legendre polynomials listed in Chap. B.

Applying Eq. (4.1) in Eq. (3.17) gives a scattering source defined by

$$q_s^g(\mathbf{\Omega}) = \sum_{g'=0}^{G} \int_{4\pi} \sum_{l=0}^{N} \frac{2l+1}{4\pi} P_l(\mathbf{\Omega}'\cdot\mathbf{\Omega}) \sigma_{\mathrm{s}\,l}^{gg'} \psi^{g'}(\mathbf{\Omega}')\, d\mathbf{\Omega}' \ , \tag{4.3}$$

where we have suppressed the spatial dependence, $\mathbf{r}$. The addition theorem of Spherical Harmonics can be used to evaluate the Legendre function, $P_l(\mathbf{\Omega}'\cdot\mathbf{\Omega})$,

$$P_l(\mathbf{\Omega}'\cdot\mathbf{\Omega}) = \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_{lm}(\mathbf{\Omega}) Y_{lm}^*(\mathbf{\Omega}') \ , \tag{4.4}$$

where the Y_{lm} are defined in Eq. (A.2). The scattering must be real; therefore, we can follow a methodology similar to the techniques described in Chap. A that shows how to expand a real-valued function using complex

Spherical Harmonics. First, the expansion is split into positive and negative components of m,

$$P_l(\mathbf{\Omega}' \cdot \mathbf{\Omega}) = \frac{4\pi}{2l+1}\Big[Y_{l0}(\mathbf{\Omega})Y_{l0}(\mathbf{\Omega}') + \sum_{m=1}^{l}\big(Y_{lm}(\mathbf{\Omega})Y_{lm}^*(\mathbf{\Omega}') + Y_{l-m}(\mathbf{\Omega})Y_{l-m}^*(\mathbf{\Omega}')\big)\Big] . \tag{4.5}$$

Examining the $m = 0$ term gives the following result

$$Y_{l0} = \sqrt{\frac{2l+1}{4\pi}}\, P_{l0} = Y_{l0}^e , \tag{4.6}$$

where the Y^e are defined in Eq. (A.20).

Expanding the Spherical Harmonics into real and imaginary components as shown in Eq. (A.15), the sum over $m > 0$ becomes

$$\sum_{m=1}^{l}\Big(\hat{Y}_{lm}^e(\mathbf{\Omega})\hat{Y}_{lm}^e(\mathbf{\Omega}') + \hat{Y}_{lm}^o(\mathbf{\Omega})\hat{Y}_{lm}^o(\mathbf{\Omega}') + \hat{Y}_{l-m}^e(\mathbf{\Omega})\hat{Y}_{l-m}^e(\mathbf{\Omega}') + \hat{Y}_{l-m}^o(\mathbf{\Omega})\hat{Y}_{l-m}^o(\mathbf{\Omega}')\Big) , \tag{4.7}$$

where the imaginary terms have been set to zero because the scattering must be real. Using Eqs. (B.7) and (B.8), the summation becomes

$$\sum_{m=1}^{l}\Big(2\hat{Y}_{lm}^e(\mathbf{\Omega})\hat{Y}_{lm}^e(\mathbf{\Omega}') + 2\hat{Y}_{lm}^o(\mathbf{\Omega})\hat{Y}_{lm}^o(\mathbf{\Omega}')\Big) \tag{4.8}$$

Comparing Eqs. (A.13) and (A.14) with Eqs. (A.20) and (A.21) leads to the following relationships,

$$\hat{Y}_{lm}^e = \frac{1}{\sqrt{2}}Y_{lm}^e , \quad \hat{Y}_{lm}^o = \frac{1}{\sqrt{2}}Y_{lm}^o . \tag{4.9}$$

After applying these equations in the $m > 0$ terms and combining with the $m = 0$ term described above, the expression for $P_l(\mathbf{\Omega} \cdot \mathbf{\Omega}')$ is

$$P_l(\mathbf{\Omega}' \cdot \mathbf{\Omega}) = \frac{4\pi}{2l+1}\Big[Y_{l0}^e(\mathbf{\Omega})Y_{l0}^e(\mathbf{\Omega}') + \sum_{m=1}^{l}\big(Y_{lm}^e(\mathbf{\Omega})Y_{lm}^e(\mathbf{\Omega}') + Y_{lm}^o(\mathbf{\Omega})Y_{lm}^o(\mathbf{\Omega}')\big)\Big] , \tag{4.10}$$

where, as shown in Chap. A, the Y^e and Y^o form an orthonormal basis.

Returning to the scattering source defined in Eq. (4.3), Eq. (4.10) provides the Legendre polynomial for the cosine of the scattering angle, and

$$q_s^g(\mathbf{\Omega}) = \sum_{g'=0}^{G}\int_{4\pi}\sum_{l=0}^{N}\Big[Y_{l0}^e(\mathbf{\Omega})Y_{l0}^e(\mathbf{\Omega}') + \sum_{m=1}^{l}\big(Y_{lm}^e(\mathbf{\Omega})Y_{lm}^e(\mathbf{\Omega}') + Y_{lm}^o(\mathbf{\Omega})Y_{lm}^o(\mathbf{\Omega}')\big)\Big]\sigma_{\mathrm{s}\,l}^{gg'}\psi^{g'}(\mathbf{\Omega}')\, d\mathbf{\Omega}' . \tag{4.11}$$

Rearranging terms and defining

$$\phi_{lm}^g = \int_{4\pi} Y_{lm}^e(\mathbf{\Omega})\psi^g(\mathbf{\Omega})\, d\mathbf{\Omega} , \quad m \geq 0 , \tag{4.12}$$

$$\vartheta_{lm}^g = \int_{4\pi} Y_{lm}^o(\mathbf{\Omega})\psi^g(\mathbf{\Omega})\, d\mathbf{\Omega} , \quad m > 0 , \tag{4.13}$$

which follows directly from Eqs. (A.28) and (A.29), the scattering source becomes

$$q_s^g(\mathbf{r}, \mathbf{\Omega}) = \sum_{g'=0}^{G}\sum_{l=0}^{N}\sigma_{\mathrm{s}\,l}^{gg'}(\mathbf{r})\Big[Y_{l0}^e(\mathbf{\Omega})\phi_{l0}^{g'}(\mathbf{r}) + \sum_{m=1}^{l}\big(Y_{lm}^e(\mathbf{\Omega})\phi_{lm}^{g'}(\mathbf{r}) + Y_{lm}^o(\mathbf{\Omega})\vartheta_{lm}^{g'}(\mathbf{r})\big)\Big] . \tag{4.14}$$

Equation (4.14) is the multigroup anisotropic scattering source that is defined by the order of the Legendre expansion, P_N, of the scattering. For a given P_N order, $(N+1)^2$ moments are required to integrate the scattering operator. The moments in Eqs. (4.12) and (4.13) are the *angular flux moments* or, simply, flux moments.

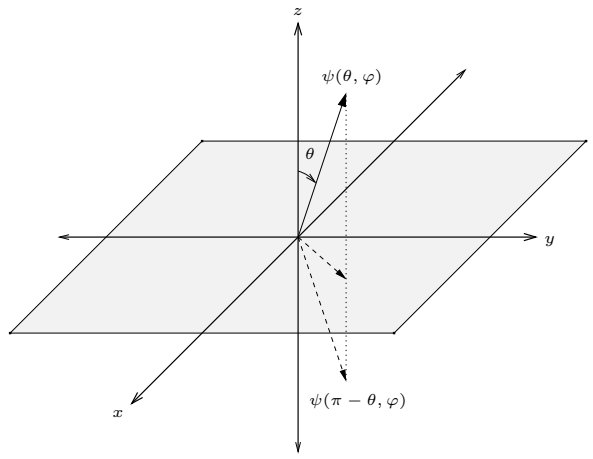


FIGURE 4.1. 2D coordinate system used in Denovo. Symmetry is defined about the XY plane.

The scalar flux is defined in Eq.(2.3) as the zeroth moment of the angular flux. Therefore, we have

$$\phi^g = \int_{4\pi} \psi^g \, d\mathbf{\Omega} = \sqrt{4\pi} \int_{4\pi} Y_{00}^e \psi^g \, d\mathbf{\Omega} = \sqrt{4\pi}\phi_{00}^g \, . \tag{4.15}$$

The current is defined

$$\begin{aligned} \mathbf{J}^g &= \int_{4\pi} \left[\mu\psi^g \hat{\mathbf{e}}_x + \eta\psi^g \hat{\mathbf{e}}_y + \xi\psi^g \hat{\mathbf{e}}_z \right] d\mathbf{\Omega} \\ &= -\sqrt{\frac{4\pi}{3}}\phi_{11}^g \hat{\mathbf{e}}_x - \sqrt{\frac{4\pi}{3}}\vartheta_{11}^g \hat{\mathbf{e}}_y + \sqrt{\frac{4\pi}{3}}\phi_{10}^g \hat{\mathbf{e}}_z \, . \end{aligned} \tag{4.16}$$

More concisely,

$$J_x^g = -\sqrt{\frac{4\pi}{3}}\phi_{11}^g \, , \quad J_y^g = -\sqrt{\frac{4\pi}{3}}\vartheta_{11}^g \, , \quad J_z^g = \sqrt{\frac{4\pi}{3}}\phi_{10}^g \, , \tag{4.17}$$

for $\mathbf{J} = J_x\hat{\mathbf{e}}_x + J_y\hat{\mathbf{e}}_y + J_z\hat{\mathbf{e}}_z$.

1.1. Scattering Expansions in 2D Geometry. To expand the scattering kernel in 2D geometry, we must define a symmetry plane. We choose the XY-plane as the symmetry plane; although the XZ-plane could be chosen as well. The resulting 2D XY coordinate system is shown in Fig. 4.1. The anisotropic scattering source is defined in Eq. (4.14). Using symmetry, the number of terms required to define this source can be reduced. The symmetry illustrated in Fig. 4.1 yields the following constraint equation

$$\psi(\theta, \varphi) = \psi(\pi - \theta, \varphi) \, , \quad 0 \le \theta \le \frac{\pi}{2} \, . \tag{4.18}$$

With this constraint in mind, we expand the integral in Eq. (4.12) as follows

$$\phi_{lm} = \int_0^{2\pi} d\varphi \int_0^{\pi} Y_{lm}^e(\theta, \varphi)\psi(\theta, \varphi) \sin\theta \, d\theta \, . \tag{4.19}$$

The inner integral over the polar angle can be split into 2 components,

$$\int_0^{\pi} Y_{lm}^e(\theta, \varphi)\psi(\theta, \varphi) \sin\theta \, d\theta = \int_0^{\pi/2} Y_{lm}^e(\theta, \varphi)\psi(\theta, \varphi) \sin\theta \, d\theta + \int_{\pi/2}^{\pi} Y_{lm}^e(\theta, \varphi)\psi(\theta, \varphi) \sin\theta \, d\theta \, . \tag{4.20}$$

Transforming variables to $\theta' = \pi - \theta$ in the second integral on the right-hand side gives

$$\int_{\pi/2}^{\pi} Y_{lm}^e(\theta, \varphi)\psi(\theta, \varphi) \sin\theta \, d\theta = \int_0^{\pi/2} Y_{lm}^e(\pi - \theta, \varphi)\psi(\pi - \theta, \varphi) \sin(\pi - \theta) \, d\theta \, . \tag{4.21}$$

Applying $\sin(\pi-\theta)=\sin\theta$ and the symmetry constraint to the preceding two equations and substituting into Eq. (4.19), the even angular flux moments become

$$\phi_{lm}=\int_0^{2\pi}d\varphi\int_0^{\pi/2}\big(Y_{lm}^e(\theta,\varphi)+Y_{lm}^e(\pi-\theta,\varphi)\big)\psi(\theta,\varphi)\sin\theta\,d\theta\,. \tag{4.22}$$

Substituting Eq. (A.20) into the preceding expression gives

$$\phi_{lm}=\int_0^{2\pi}d\varphi\int_0^{\pi/2}D_{lm}\cos m\varphi\Big(P_{lm}(\cos\theta)+P_{lm}\big(\cos(\pi-\theta)\big)\Big)\psi(\theta,\varphi)\sin\theta\,d\theta\,. \tag{4.23}$$

Using $\cos(\pi-\theta)=-\cos\theta$ and applying the following identity [**5**],

$$P_{lm}(-x)=(-1)^{l+m}P_{lm}(x)\,, \tag{4.24}$$

we derive the following formula for calculating the even angular flux moments,

$$\phi_{lm}=\int_0^{2\pi}d\varphi\int_0^{\pi/2}\big(1+(-1)^{l+m}\big)Y_{lm}^e(\theta,\varphi)\psi(\theta,\varphi)\sin\theta\,d\theta\,. \tag{4.25}$$

Using the same procedure for the odd moments, the angular flux moments are defined

$$\phi_{lm}=\int_0^{2\pi}d\varphi\int_0^{1}\big(1+(-1)^{l+m}\big)Y_{lm}^e(\theta,\varphi)\psi(\theta,\varphi)\,d\xi\,, \tag{4.26}$$

$$\vartheta_{lm}=\int_0^{2\pi}d\varphi\int_0^{1}\big(1+(-1)^{l+m}\big)Y_{lm}^o(\theta,\varphi)\psi(\theta,\varphi)\,d\xi\,, \tag{4.27}$$

where we have written the integrals over $d\theta$ using the substitution $\xi=\cos\theta$. Equations (4.26) and (4.27) are the 2D equivalents of Eqs. (4.12) and (4.13) when symmetry is defined about the XY plane.

A brief examination of Eqs. (4.26) and (4.27) reveals that for all odd sums of $l+m$ (ie. $l+m\in\{2k+1;\forall k\in Z\}$) the angular moments vanish. This results in $(N+1)(N+2)/2$ angular moments for P_N scattering whereas 3D calculations require $(N+1)^2$ moments. Furthermore, we only need to consider quadrature angles in the $0\le\xi\le1$ directions and the factor of 2 that results when $l+m$ is even preserves the integration over 4π; this is a natural result of symmetry in the polar direction. Three-dimensional space contains 8 octants that consist of 4 octants for $+\xi$ and 4 octants for $-\xi$. In 2D space, only the 4 octants in $+\xi$ are required. In 2D geometry these 4 octants are referred to as quadrants.

2. Angular Discretization of External Sources

In most cases, the external source will be a known function of angle. For example, an isotropic external source is

$$q_e^g(\mathbf{\Omega})=\frac{s^g}{4\pi}\,. \tag{4.28}$$

However, in certain cases (e.g. coupling to a k-eigenvalue calculation) the external source is defined in moments. Equation (A.27) gives the Spherical Harmonics expansion of a real-valued function. Applying the same methodology gives the expansion of the external source

$$q_e^g(\mathbf{\Omega})=\sum_{l=0}^{N}\Big[Y_{l0}^e(\mathbf{\Omega})q_{l0}^g+\sum_{m=1}^{l}\big(Y_{lm}^e(\mathbf{\Omega})q_{lm}^g+Y_{lm}^o(\mathbf{\Omega})s_{lm}^g\big)\Big]\,, \tag{4.29}$$

where the spatial dependence has been suppressed. The even and odd source moments are defined

$$q_{lm}^g=\int_{4\pi}Y_{lm}^e(\mathbf{\Omega})q_e^g(\mathbf{\Omega})\,d\mathbf{\Omega}\,,\quad m\ge0\,, \tag{4.30}$$

$$s_{lm}^g=\int_{4\pi}Y_{lm}^o(\mathbf{\Omega})q_e^g(\mathbf{\Omega})\,d\mathbf{\Omega}\,,\quad m>0\,. \tag{4.31}$$

Now, using Eqs.(4.14) and (4.29), we write the entire source as

$$\begin{aligned} Q^g(\mathbf{r},\mathbf{\Omega}) &= q_s^g(\mathbf{r},\mathbf{\Omega}) + q_e^g(\mathbf{r},\mathbf{\Omega}) \\ &= \sum_{g'=0}^{G}\sum_{l=0}^{N}\sigma_{\mathrm{s}\,l}^{gg'}(\mathbf{r})\Big[Y_{l0}^e(\mathbf{\Omega})\phi_{l0}^{g'}(\mathbf{r}) + \sum_{m=1}^{l}\left(Y_{lm}^e(\mathbf{\Omega})\phi_{lm}^{g'}(\mathbf{r}) + Y_{lm}^o(\mathbf{\Omega})\vartheta_{lm}^{g'}(\mathbf{r})\right)\Big] \\ &\quad + \sum_{l=0}^{N}\Big[Y_{l0}^e(\mathbf{\Omega})q_{l0}^g(\mathbf{r}) + \sum_{m=1}^{l}\left(Y_{lm}^e(\mathbf{\Omega})q_{lm}^g(\mathbf{r}) + Y_{lm}^o(\mathbf{\Omega})s_{lm}^g(\mathbf{r})\right)\Big]\,. \end{aligned} \tag{4.32}$$

3. Angular Collocation Discretization

Combining Eqs. (3.17) and (4.32) gives the multigroup transport equation with the scattering expanded in Spherical Harmonics,

$$\hat{\mathbf{\Omega}}\cdot\nabla\psi^g(\mathbf{\Omega}) + \sigma^g\psi^g(\mathbf{\Omega}) = Q^g(\mathbf{\Omega}) \tag{4.33}$$

where the spatial dependence has been suppressed. While Spherical Harmonics have been used to expand the scattering sources (and possibly the external source), we still have a dependence on $\mathbf{\Omega}$ that needs to be resolved. We apply the discrete ordinates (S_N) approximation, which is a collocation method in angle. Solving Eq. (4.33) at discrete angular locations requires the following equation,

$$\hat{\mathbf{\Omega}}_a\cdot\nabla\psi_a^g + \sigma^g\psi_a^g = \sum_{g'=0}^{G}\sum_{l=0}^{N}\sigma_{\mathrm{s}\,l}^{gg'}\left[Y_{l0}^e(\mathbf{\Omega}_a)\phi_{l0}^{g'} + \sum_{m=1}^{l}\left(Y_{lm}^e(\mathbf{\Omega}_a)\phi_{lm}^{g'} + Y_{lm}^o(\mathbf{\Omega}_a)\vartheta_{lm}^{g'}\right)\right] + q_e^g(\mathbf{\Omega}_a)\,, \tag{4.34}$$

where $\psi_a^g \equiv \psi^g(\mathbf{\Omega}_a)$. The angles are integrated by a quadrature rule such that

$$\int_{4\pi} d\mathbf{\Omega} = \sum_{a=1}^{n} w_a = 4\pi\,, \tag{4.35}$$

where w_a are the quadrature weights, and n is the total number of angles. Different quadrature sets have different numbers of unknowns. The Level-Symmetric quadrature set has $n = N(N+2)$ unknowns for an S_N approximation.

Given that many angles result from even low-order S_N approximations, we see why the source has been expanded in Spherical Harmonics. Consider, an S_8 calculation has 80 angles per unknown location per group. For a P_3 expansion 16 moments are required to define the source, a factor of 5 reduction in memory storage. An S_{16} calculation could be used for more accuracy with a P_3 calculation resulting in a factor of 18 savings in memory.

Using the quadrature integration rule the flux moments in Eqs. (4.12) and (4.13) are evaluated using

$$\phi_{lm}^g = \sum_{a=1}^{n} Y_{lm}^e(\mathbf{\Omega}_a)\psi_a^g w_a\,, \tag{4.36}$$

$$\vartheta_{lm}^g = \sum_{a=1}^{n} Y_{lm}^o(\mathbf{\Omega}_a)\psi_a^g w_a\,. \tag{4.37}$$

Similarly, the source moments in Eqs. (4.30) and (4.31) are calculated using

$$q_{lm}^g = \sum_{a=1}^{n} Y_{lm}^e(\mathbf{\Omega}_a)q_e^g(\mathbf{\Omega}_a)w_a\,, \tag{4.38}$$

$$s_{lm}^g = \sum_{a=1}^{n} Y_{lm}^o(\mathbf{\Omega}_a)q_e^g(\mathbf{\Omega}_a)w_a\,. \tag{4.39}$$

The S_N method will be conservative if the quadrature set effectively integrates the even and odd Spherical Harmonics. If the orthogonality conditions in Eqs. (A.23) and (A.24) are preserved, then integrating the

anisotropic scattering should yield,

$$\int_{4\pi} q_s^g(\mathbf{\Omega})\, d\mathbf{\Omega} = \sum_{a=1}^{n} q_s^g(\mathbf{\Omega}_a) w_a = \sqrt{4\pi}\sigma_{\mathrm{s}\,0}^{gg'}\phi_{00}^{g'} = \sigma_s^{gg'}\phi^{g'}\ , \tag{4.40}$$

which will yield a conservative particle balance equation.

4. Operator Form of the Discrete Ordinates Equation

The multigroup discrete ordinates, or S_N equation is (see Eq. (4.34))

$$\begin{aligned}\hat{\mathbf{\Omega}}_a \cdot \nabla\psi_a^g(\mathbf{r}) + \sigma^g(\mathbf{r})\psi_a^g(\mathbf{r}) \\ = \sum_{g'=0}^{G}\sum_{l=0}^{N}\sigma_{\mathrm{s}\,l}^{gg'}(\mathbf{r})\Big[Y_{l0}^e(\mathbf{\Omega}_a)\phi_{l0}^{g'}(\mathbf{r}) + \sum_{m=1}^{l}\big(Y_{lm}^e(\mathbf{\Omega}_a)\phi_{lm}^{g'}(\mathbf{r}) + Y_{lm}^o(\mathbf{\Omega}_a)\vartheta_{lm}^{g'}(\mathbf{r})\big)\Big] + q_e^g(\mathbf{r},\mathbf{\Omega}_a)\ .\end{aligned} \tag{4.34}$$

We defer the spatial treatment of Eq. (4.34) until Chap. 6. This equation can be written using a concise operator notation that helps illuminate numerical solution techniques. The operator form of Eq. (4.34) is

$$\mathbf{L}\Psi = \mathbf{MS}\Phi + Q\ . \tag{4.41}$$

Here we use the convention that bold letters represent discrete operators or matrices and script symbols and letters represent vectors.

The sizes of the operators in Eq. (4.41) are determined from the following dimensions:

$$\begin{aligned} N_g &= \text{number of groups}\ , \\ t &= \text{number of moments}\ , \\ n &= \text{number of angles}\ , \\ N &= P_N \text{ order}\ , \\ N_c &= \text{number of cells}\ , \\ N_e &= \text{number of unknowns per cell}\ . \end{aligned} \tag{4.42}$$

Now, we define

$$a = N_g \times n \times N_c \times N_e\ , \tag{4.43}$$

$$f = N_g \times t \times N_c \times N_e\ . \tag{4.44}$$

Equation (4.41) can then be defined in terms of the sizes of the operators,

$$(a \times a)(a \times 1) = (a \times f)(f \times f)(f \times 1) + (a \times 1)\ . \tag{4.45}$$

More specifically, with the groups defined over the range $g \in [0, G]$, at each spatial unknown we can write

$$\mathbf{L}\begin{pmatrix}\Psi_0\\ \Psi_1\\ \Psi_2\\ \vdots\\ \Psi_G\end{pmatrix} = \begin{pmatrix}\mathbf{M} & 0 & 0 & 0 & 0\\ 0 & \mathbf{M} & 0 & 0 & 0\\ 0 & 0 & \mathbf{M} & 0 & 0\\ 0 & 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & 0 & \mathbf{M}\end{pmatrix}\begin{pmatrix}\mathbf{S}_{00} & \mathbf{S}_{01} & \mathbf{S}_{02} & \cdots & \mathbf{S}_{0G}\\ \mathbf{S}_{10} & \mathbf{S}_{11} & \mathbf{S}_{12} & \cdots & \mathbf{S}_{1G}\\ \mathbf{S}_{20} & \mathbf{S}_{21} & \mathbf{S}_{22} & \cdots & \mathbf{S}_{2G}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{S}_{G0} & \mathbf{S}_{G1} & \mathbf{S}_{G2} & \cdots & \mathbf{S}_{GG}\end{pmatrix}\begin{pmatrix}\Phi_0\\ \Phi_1\\ \Phi_2\\ \vdots\\ \Phi_G\end{pmatrix} + \begin{pmatrix}Q_0\\ Q_1\\ Q_2\\ \vdots\\ Q_G\end{pmatrix}\ , \tag{4.46}$$

Here, Ψ_g and Q_g are vectors of size n,

$$\Psi_g = \begin{pmatrix}\psi_1^g & \psi_2^g & \psi_3^g & \cdots \psi_n^g\end{pmatrix}^T\ , \tag{4.47}$$

$$Q_g = \begin{pmatrix}Q_1^g & Q_2^g & Q_3^g & \cdots Q_n^g\end{pmatrix}^T\ . \tag{4.48}$$

The spatial unknowns are implicit in the above matrices and will be investigated in Chaps. 6 and 8.

The operator $\mathbf{M}$ is the moment-to-discrete matrix. It is used to project harmonic moments onto discrete angle space, and it is defined

$$\mathbf{M} = \begin{pmatrix} Y_{00}^{e}(\mathbf{\Omega}_1) & Y_{10}^{e}(\mathbf{\Omega}_1) & Y_{11}^{o}(\mathbf{\Omega}_1) & Y_{11}^{e}(\mathbf{\Omega}_1) & Y_{20}^{e}(\mathbf{\Omega}_1) & \cdots & Y_{NN}^{o}(\mathbf{\Omega}_1) & Y_{NN}^{e}(\mathbf{\Omega}_1) \\ Y_{00}^{e}(\mathbf{\Omega}_2) & Y_{10}^{e}(\mathbf{\Omega}_2) & Y_{11}^{o}(\mathbf{\Omega}_2) & Y_{11}^{e}(\mathbf{\Omega}_2) & Y_{20}^{e}(\mathbf{\Omega}_2) & \cdots & Y_{NN}^{o}(\mathbf{\Omega}_2) & Y_{NN}^{e}(\mathbf{\Omega}_2) \\ Y_{00}^{e}(\mathbf{\Omega}_3) & Y_{10}^{e}(\mathbf{\Omega}_3) & Y_{11}^{o}(\mathbf{\Omega}_3) & Y_{11}^{e}(\mathbf{\Omega}_3) & Y_{20}^{e}(\mathbf{\Omega}_3) & \cdots & Y_{NN}^{o}(\mathbf{\Omega}_3) & Y_{NN}^{e}(\mathbf{\Omega}_3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ Y_{00}^{e}(\mathbf{\Omega}_n) & Y_{10}^{e}(\mathbf{\Omega}_n) & Y_{11}^{o}(\mathbf{\Omega}_n) & Y_{11}^{e}(\mathbf{\Omega}_n) & Y_{20}^{e}(\mathbf{\Omega}_n) & \cdots & Y_{NN}^{o}(\mathbf{\Omega}_n) & Y_{NN}^{e}(\mathbf{\Omega}_n) \end{pmatrix} . \tag{4.49}$$

The moments of the angular flux are calculated from discrete angular fluxes using the moment-to-discrete matrix,

$$\phi = \mathbf{D}\psi \, . \tag{4.50}$$

Clearly, the discrete form of $\mathbf{D}$ is defined by the quadrature integration rules in Eqs. (4.36) and (4.37),

$$\mathbf{D} = \mathbf{M}^T\mathbf{W} \, . \tag{4.51}$$

where $\mathbf{W}$ is an $(a \times a)$ diagonal matrix of the quadrature weights. Using the definition of $\mathbf{M}$ from Eq. (4.45), the size of $\mathbf{D}$ can be determined from

$$\mathbf{D} \equiv (f \times a)(a \times a) = (f \times a) \, . \tag{4.52}$$

The $\mathbf{D}$ matrix has the following form

$$\mathbf{D} = \begin{pmatrix} w_1 Y_{00}^{e}(\mathbf{\Omega}_1) & w_2 Y_{00}^{e}(\mathbf{\Omega}_2) & w_3 Y_{00}^{e}(\mathbf{\Omega}_3) & \cdots & w_n Y_{00}^{e}(\mathbf{\Omega}_n) \\ w_1 Y_{10}^{e}(\mathbf{\Omega}_1) & w_2 Y_{10}^{e}(\mathbf{\Omega}_2) & w_3 Y_{10}^{e}(\mathbf{\Omega}_3) & \cdots & w_n Y_{10}^{e}(\mathbf{\Omega}_n) \\ w_1 Y_{11}^{o}(\mathbf{\Omega}_1) & w_2 Y_{11}^{o}(\mathbf{\Omega}_2) & w_3 Y_{11}^{o}(\mathbf{\Omega}_3) & \cdots & w_n Y_{11}^{o}(\mathbf{\Omega}_n) \\ w_1 Y_{11}^{e}(\mathbf{\Omega}_1) & w_2 Y_{11}^{e}(\mathbf{\Omega}_2) & w_3 Y_{11}^{e}(\mathbf{\Omega}_3) & \cdots & w_n Y_{11}^{e}(\mathbf{\Omega}_n) \\ w_1 Y_{20}^{e}(\mathbf{\Omega}_1) & w_2 Y_{20}^{e}(\mathbf{\Omega}_2) & w_3 Y_{20}^{e}(\mathbf{\Omega}_3) & \cdots & w_n Y_{20}^{e}(\mathbf{\Omega}_n) \\ \vdots & \vdots & \vdots & & \vdots \\ w_1 Y_{NN}^{o}(\mathbf{\Omega}_1) & w_2 Y_{NN}^{o}(\mathbf{\Omega}_2) & w_3 Y_{NN}^{o}(\mathbf{\Omega}_3) & \cdots & w_n Y_{NN}^{o}(\mathbf{\Omega}_n) \\ w_1 Y_{NN}^{e}(\mathbf{\Omega}_1) & w_2 Y_{NN}^{e}(\mathbf{\Omega}_2) & w_3 Y_{NN}^{e}(\mathbf{\Omega}_3) & \cdots & w_n Y_{NN}^{e}(\mathbf{\Omega}_n) \end{pmatrix} . \tag{4.53}$$

Also, even though $\mathbf{M}$ projects angular flux moments onto discrete angular flux space, in general $\psi \neq \mathbf{M}\phi$ unless $\mathbf{M} = \mathbf{D}^{-1}$. This condition is met when using the *Galerkin* quadrature set, but most quadrature sets do not satisfy this requirement. The moments of the angular flux vector are defined

$$\Phi_g = \begin{pmatrix} \phi_{00}^{g} & \phi_{10}^{g} & \vartheta_{11}^{g} & \phi_{11}^{g} & \phi_{20}^{g} & \cdots & \vartheta_{NN}^{g} & \phi_{NN}^{g} \end{pmatrix}^T , \tag{4.54}$$

The scattering cross sections are defined

$$\mathbf{S}_{gg'} = \begin{pmatrix} \sigma_{\mathrm{s}\,0}^{gg'} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\mathrm{s}\,1}^{gg'} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\mathrm{s}\,1}^{gg'} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\mathrm{s}\,1}^{gg'} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\mathrm{s}\,2}^{gg'} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\mathrm{s}\,N}^{gg'} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\mathrm{s}\,N}^{gg'} \end{pmatrix} . \tag{4.55}$$

Here the block-matrix $\mathbf{S}_{gg'}$ defines scattering cross sections for particles that scatter from group g' into group g. The lower triangular part of $\mathbf{S}$ represents down-scattering, the diagonal represents in-group scattering, and the upper diagonal is up-scattering.

CHAPTER 5

Spectral Angular Methods

In Chap. 4 finite element collocation methods were used to discretize the angular variable in Eq. (3.17). Another class of methods exist in which the angular terms are treated using function expansions. The two most widely used of these methods are the Spherical Harmonics (P_N) and simplified Spherical Harmonics (SP_N) methods.

1. Spherical Harmonics

The Spherical Harmonics, or P_N equations, can be written in multiple dimensions. In planar geometry the spherical harmonics reduce to Legendre polynomials. Since the Legendre polynomials are considerably easier to manipulate, we begin our derivation by examining the planar P_N equations. Afterwards, we will extend the equations to 3-dimensions using the complete spherical harmonics.

1.1. P_N Equations. We begin the derivation of the planar P_N equations from the steady-state, one-dimensional, monoenergetic transport equation,

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\sigma(x)\psi(x,\mu)=\int_{4\pi}\sigma_{\mathrm{s}}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'+\frac{q(x)}{4\pi}\,, \tag{5.1}$$

with boundary conditions,

$$\psi(x,\mu)=\psi_{\mathrm{b}}(x,\mu)\,,\quad x\in\partial V\,. \tag{5.2}$$

Here, the standard definitions hold:

$\psi(x,\mu)$	angular flux in particles·cm^{-2}·str^{-1}
$\sigma(x)$	total interaction cross section in cm^{-1}
$\sigma_{\mathrm{s}}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')$	scattering cross section through angle $\mu_0=\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}'$
$q(x)$	isotropic source in particles·cm^{-3}

The P_N equations are obtained by expanding the angular flux and scattering in Legendre polynomials (this requires spherical harmonics in two and three dimensions and non-cartesian geometry):

$$\psi(\mu)=\sum_{n=0}^{N}\frac{2n+1}{4\pi}\phi_n P_n(\mu)\,, \tag{5.3}$$

$$\sigma_{\mathrm{s}}(\mu_0)=\sum_{m=0}^{N}\frac{2m+1}{4\pi}\sigma_{\mathrm{s}m}P_m(\mu_0)\,, \tag{5.4}$$

where $\mu_0=\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}'$. In what follows we shall make use of the following properties of Legendre polynomials:

$$\int_{-1}^{1}P_n(\mu)P_m(\mu)\,d\mu=\frac{2}{2n+1}\delta_{nm}\,,\qquad\text{(orthogonality)} \tag{5.5}$$

$$(2n+1)\mu P_n(\mu)=(n+1)P_{n+1}(\mu)+nP_{n-1}(\mu)\,,\qquad\text{(recursion)} \tag{5.6}$$

$$P_l(\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')=\frac{4\pi}{2n+1}\sum_{m=-l}^{l}Y_{lm}(\mathbf{\Omega})Y_{lm}^{*}(\mathbf{\Omega}')\,,\qquad\text{(addition theorem)} \tag{5.7}$$

Expanding the addition theorem we obtain

$$P_l(\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}') = \frac{4\pi}{2n+1}\left[Y_{l0}(\mathbf{\Omega})Y_{l0}^*(\mathbf{\Omega}') + \sum_{m=1}^{l}\left(Y_{l-m}(\mathbf{\Omega})Y_{l-m}^*(\mathbf{\Omega}') + Y_{lm}(\mathbf{\Omega})Y_{lm}^*(\mathbf{\Omega}')\right)\right] .$$

In planar geometry there is no azimuthal dependence and only $m = 0$ terms are required. Also, the spherical harmonics reduce to Legendre polynomials in planar geometry,

$$Y_{l0} = \sqrt{\frac{2l+1}{4\pi}}P_{l0} = \sqrt{\frac{2l+1}{4\pi}}P_l .$$

Combining these two equations, the addition theorem in planar geometry is

$$P_l(\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}') = P_l(\mu_0) = P_l(\mu)P_l(\mu') . \tag{5.8}$$

From orthogonality we have

$$\phi_n = 2\pi\int_{-1}^{1} P_n(\mu)\psi(\mu)\,d\mu . \tag{5.9}$$

Applying the expansions in Eqs. (5.3) and (5.4) in Eq. (5.1) gives

$$\mu\frac{\partial}{\partial x}\left[\sum_n \frac{2n+1}{4\pi}\phi_n P_n(\mu)\right] + \sigma\sum_n \frac{2n+1}{4\pi}\phi_n P_n(\mu) = 2\pi\int_{-1}^{1}\sum_m \frac{2m+1}{4\pi}\sigma_{sm}P_m(\mu_0)\sum_n \frac{2n+1}{4\pi}\phi_n P_n(\mu)\,d\mu' + \frac{q}{4\pi} , \tag{5.10}$$

where we have suppressed the x dependence. The P_N equations are obtained by multiplying by $P_m(\mu)$ and integrating by $\int_{-1}^{1} d\mu$. Equation (5.6) is used to remove μP_n from the derivative term. Equation (5.8) is used in the scattering expansion to remove the μ_0 dependence. Orthogonality is used to remove all the remaining Legendre polynomials. The resulting system of equations is

$$\frac{\partial}{\partial x}\Big[\frac{n}{2n+1}\phi_{n-1} + \frac{n+1}{2n+1}\phi_{n+1}\Big] + \Sigma_n\phi_n = q\delta_{n0} , \quad n = 0, 1, 2, \ldots, N , \tag{5.11}$$

where

$$\Sigma_n = \sigma - \sigma_{sn} . \tag{5.12}$$

Equation (5.11) defines a system of $N + 1$ equations that requires closure in order to deal with the ϕ_{n+1} term in the differential operator. The common method for closing the equations is to set this term to zero, $\phi_{N+1} = 0$. As an example, the P_3 equations are

$$\begin{aligned}
\frac{\partial}{\partial x}(\phi_1) + \Sigma_0\phi_0 &= q ,\\
\frac{1}{3}\frac{\partial}{\partial x}(\phi_0 + 2\phi_2) + \Sigma_1\phi_1 &= 0 ,\\
\frac{1}{5}\frac{\partial}{\partial x}(2\phi_1 + 3\phi_3) + \Sigma_2\phi_2 &= 0 ,\\
\frac{1}{7}\frac{\partial}{\partial x}(3\phi_2) + \Sigma_3\phi_3 &= 0 .
\end{aligned} \tag{5.13}$$

1.2. P_N Boundary Conditions. For this work we consider 3 types of boundary conditions:

- vacuum
- isotropic flux
- reflecting

For vacuum and isotropic flux we will employ the Marshak boundary conditions. The Marshak conditions approximately satisfy Eq. (5.2) at the boundary and are consistent with the P_N approximation. The generalized Marshak boundary condition is

$$2\pi \int_{\mu_{\text{in}}} P_i(\mu)\psi(\mu)\,d\mu = 2\pi \int_{\mu_{\text{in}}} P_i(\mu)\psi_{\text{b}}(\mu)\,d\mu\,, \quad i = 1, 3, 5, \ldots, N\,. \tag{5.14}$$

Expanding ψ using Eq. (5.3) gives

$$2\pi \int_{\mu_{\text{in}}} P_i(\mu) \sum_{n=0}^{N} \frac{2n+1}{4\pi}\phi_n P_n(\mu)\,d\mu = 2\pi \int_{\mu_{\text{in}}} P_i(\mu)\psi_{\text{b}}(\mu)\,d\mu\,, \quad i = 1, 3, 5, \ldots, N\,. \tag{5.15}$$

Equation (5.15) yields $(N+1)/2$ fully coupled equations at each boundary. Thus, it fully closes the $N+1$ P_N equations given in Eq. (5.11).

Once again, as an example we consider the P_3 equations. The Marshak conditions on the low boundary are derived using

$$2\pi \int_0^1 P_1(\mu) \sum_{n=0}^{3} \frac{2n+1}{4\pi}\phi_n P_n(\mu)\,d\mu = 2\pi \int_0^1 P_1(\mu)\psi_{\text{b}}(\mu)\,d\mu \tag{5.16}$$

$$2\pi \int_0^1 P_3(\mu) \sum_{n=0}^{3} \frac{2n+1}{4\pi}\phi_n P_n(\mu)\,d\mu = 2\pi \int_0^1 P_3(\mu)\psi_{\text{b}}(\mu)\,d\mu \tag{5.17}$$

Assuming an isotropic flux on the boundary,

$$\psi_{\text{b}}(\mu) = \frac{\phi_{\text{b}}}{4\pi}\,, \tag{5.18}$$

the P_3 Marshak boundary conditions are

$$\frac{1}{2}\phi_0 + \phi_1 + \frac{5}{8}\phi_2 = \frac{1}{2}\phi_{\text{b}}\,, \tag{5.19}$$

$$-\frac{1}{8}\phi_0 + \frac{5}{8}\phi_2 + \phi_3 = -\frac{1}{8}\phi_{\text{b}}\,. \tag{5.20}$$

As stated above, all of the moments are coupled in the boundary conditions. For a vacuum condition, $\phi_{\text{b}} = 0$.

Reflecting boundary conditions are more straightforward. The only conditions that make physical sense in this case is to set all the odd moments to zero

$$\phi_i = 0\,, \quad i = 1, 3, 5, \ldots, N\,. \tag{5.21}$$

In the P_1 approximation this is equivalent to setting the current to zero at each boundary. From Eq. (5.9)

$$\phi_1 = 2\pi \int_{-1}^{1} \mu\psi(\mu)\,d\mu = J = 0\,. \tag{5.22}$$

This treatment also yields $(N+1)/2$ equations on each boundary and effectively closes the system.

We note that both of these boundary treatments contain asymmetric components when $N \in \{\text{even}\}$. Thus, we only consider odd sets of P_N (SP_N) equations.

2. Simplified Spherical Harmonics

The Simplified P_N (SP_N) approximation is a three-dimensional extension of the plane-geometry P_N equations. It was originally proposed by Gelbard [**6**] who applied heuristic arguments to justify the approximation. Since that time, both asymptotic [**7**–**9**] and variational [**10**] analyses have verified Gelbard's approach.

Here, we derive the SP_N equations using the original method of Gelbard. The presentation closely follows Refs. [**9**] and [**11**].

2.1. SP_N Equations. As mentioned in above the SP_N method is based on heuristic arguments; however, several studies have performed both asymptotic and variational analysis that have confirmed the original *ad hoc* approximations. In this note, we shall apply the heuristic approximation. The reader is directed towards Refs. [**7**–**9**] for more details on asymptotic derivations of the equations and Ref. [**10**] for a variational analysis of the SP_N equations.

We begin by applying the multigroup approximation (Chap. 3) to Eq. (5.1) yielding

$$\mu\frac{\partial\psi^g(x,\mu)}{\partial x}+\sigma^g(x)\psi^g(x,\mu)=\sum_{g'=0}^{G}\int_{4\pi}\sigma_{\mathrm{s}}^{gg'}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi^{g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'+\frac{q^g(x)}{4\pi}\,, \tag{5.23}$$

In the notation that follows we will employ the Einstein Summation convention in which identical indices are implicitly summed over the range $1,\ldots,3$,

$$a_ib_i=\sum_{i=1}^{3}a_ib_i=\mathbf{A}\cdot\mathbf{B}\,. \tag{5.24}$$

To form the SP_N equations the following substitutions are made in Eq. (5.11):

- $\frac{\partial}{\partial x}\rightarrow\frac{\partial}{\partial x_i}$,
- convert odd moments to $\phi_{n,i}$,
- use odd-order equations to remove odd moments from the even-order equations .

For boundary conditions a similar process holds except that $\pm\frac{\partial}{\partial x}\rightarrow n_i\frac{\partial}{\partial x_i}$, where $\hat{\mathbf{n}}=n_i\mathbf{i}+n_j\mathbf{j}+n_k\mathbf{k}$ is the outward normal at a boundary surface and $\mu\rightarrow|\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{n}}|$. Using Eq. (5.11) and the rules described above, we have

$$\frac{\partial}{\partial x_i}\Big[\frac{n}{2n+1}\phi^g_{n-1,i}+\frac{n+1}{2n+1}\phi^g_{n+1,i}\Big]+\sum_{g'}(\sigma^g\delta_{gg'}-\sigma^{gg'}_{sn})\phi^{g'}_n=q^g\delta_{n0}\,,\qquad n=0,2,4,\ldots,N\,, \tag{5.25}$$

$$\frac{\partial}{\partial x_i}\Big[\frac{n}{2n+1}\phi^g_{n-1}+\frac{n+1}{2n+1}\phi^g_{n+1}\Big]+\sum_{g'}(\sigma^g\delta_{gg'}-\sigma^{gg'}_{sn})\phi^{g'}_{n,i}=0\,,\qquad n=1,3,5,\ldots,N\,. \tag{5.26}$$

Equations (5.25) and (5.26) are more easily expressed in operator notation over groups by defining

$$\Phi_n=\begin{pmatrix}\phi^0_n & \phi^1_n & \ldots & \phi^G_n\end{pmatrix}^T\,, \tag{5.27}$$

$$\Phi_{n,i}=\begin{pmatrix}\phi^0_{n,i} & \phi^1_{n,i} & \ldots & \phi^G_{n,i}\end{pmatrix}^T\,, \tag{5.28}$$

$$\mathbf{q}=\begin{pmatrix}q^0 & q^1 & \ldots & q^G\end{pmatrix}^T\,, \tag{5.29}$$

and

$$\mathbf{\Sigma}_n=\begin{pmatrix}(\sigma^0-\sigma^{00}_{sn}) & -\sigma^{01}_{sn} & \ldots & -\sigma^{0G}_{sn}\\ -\sigma^{10}_{sn} & (\sigma^1-\sigma^{11}_{sn}) & \ldots & -\sigma^{1G}_{sn}\\ \vdots & \vdots & \ddots & \vdots\\ -\sigma^{G0}_{sn} & -\sigma^{G1}_{sn} & \ldots & (\sigma^G-\sigma^{GG}_{sn})\end{pmatrix}\,. \tag{5.30}$$

Thus, at any given spatial location, Φ_n and $\Phi_{n,i}$ are length N_g vectors, and $\mathbf{\Sigma}$ is a $(N_g\times N_g)$ matrix. Using Eq. (5.26) to solve for the odd moments gives

$$\Phi_{n,i}=-\mathbf{\Sigma}_n^{-1}\frac{\partial}{\partial x_i}\Big[\frac{n}{2n+1}\Phi_{n-1}+\frac{n+1}{2n+1}\Phi_{n+1}\Big]\,. \tag{5.31}$$

Substituting Eq. (5.31) into Eq. (5.25) yields

$$-\frac{\partial}{\partial x_i}\Bigg[\frac{n}{2n+1}(\boldsymbol{\Sigma}_{n-1}^{-1})\frac{\partial}{\partial x_i}\Big(\frac{n-1}{2n-1}\Phi_{n-2}+\frac{n}{2n-1}\Phi_n\Big)+ \frac{n+1}{2n+1}(\boldsymbol{\Sigma}_{n+1}^{-1})\frac{\partial}{\partial x_i}\Big(\frac{n+1}{2n+3}\Phi_n+\frac{n+2}{2n+3}\Phi_{n+2}\Big)\Bigg]+ \boldsymbol{\Sigma}_n\Phi_n=\mathbf{q}\delta_{n0}\,,\quad m=0,2,\ldots,N\,. \tag{5.32}$$

Equation (5.32) defines the $(N+1)/2$ SP_N equations. These are a series of elliptic, second-order equations, each of which has a diffusion-like form.

Using Eq. (5.32), the four SP_7 equations are

$$\begin{aligned} -\nabla\cdot\frac{1}{3}\boldsymbol{\Sigma}_1^{-1}\nabla(\Phi_0+2\Phi_2)+\boldsymbol{\Sigma}_0\Phi_0&=\mathbf{q}\,,\\ -\nabla\cdot\Big[\frac{2}{15}\boldsymbol{\Sigma}_1^{-1}\nabla(\Phi_0+2\Phi_2)+\frac{3}{35}\boldsymbol{\Sigma}_3^{-1}\nabla(3\Phi_2+4\Phi_4)\Big]+\boldsymbol{\Sigma}_2\Phi_2&=0\,,\\ -\nabla\cdot\Big[\frac{4}{63}\boldsymbol{\Sigma}_3^{-1}\nabla(3\Phi_2+4\Phi_4)+\frac{5}{99}\boldsymbol{\Sigma}_5^{-1}\nabla(5\Phi_4+6\Phi_6)\Big]+\boldsymbol{\Sigma}_4\Phi_4&=0\,,\\ -\nabla\cdot\Big[\frac{6}{143}\boldsymbol{\Sigma}_5^{-1}\nabla(5\Phi_4+6\Phi_6)+\frac{7}{195}\boldsymbol{\Sigma}_7^{-1}\nabla(7\Phi_6)\Big]+\boldsymbol{\Sigma}_6\Phi_6&=0\,. \end{aligned} \tag{5.33}$$

A quick view of these equations reveals that certain linear combinations of moments appear together in the derivative terms. Performing the following variable transformation allows the gradient-term to operate on a single unknown in each moment equation,

$$\begin{aligned} \mathbb{U}_1&=\Phi_0+2\Phi_2\,,\\ \mathbb{U}_2&=3\Phi_2+4\Phi_4\,,\\ \mathbb{U}_3&=5\Phi_4+6\Phi_6\,,\\ \mathbb{U}_4&=7\Phi_6\,. \end{aligned} \tag{5.34}$$

The inverse of this system is

$$\begin{aligned} \Phi_0&=\mathbb{U}_1-\frac{2}{3}\mathbb{U}_2+\frac{8}{15}\mathbb{U}_3-\frac{16}{35}\mathbb{U}_4\,,\\ \Phi_2&=\frac{1}{3}\mathbb{U}_2-\frac{4}{15}\mathbb{U}_3+\frac{8}{35}\mathbb{U}_4\,,\\ \Phi_4&=\frac{1}{5}\mathbb{U}_3-\frac{6}{35}\mathbb{U}_4\,,\\ \Phi_6&=\frac{1}{7}\mathbb{U}_4\,. \end{aligned} \tag{5.35}$$

Substituting Eqs. (5.34) and (5.35) into Eq. (5.33) and successively removing the lower order gradient terms from each equation results in the following concise form

$$-\nabla\cdot\mathbb{D}_n\nabla\mathbb{U}_n+\sum_{m=1}^{4}\mathbb{A}_{nm}\mathbb{U}_m=\mathbb{Q}_n\,,\quad n=1,2,3,4\,. \tag{5.36}$$

The effective diffusion coefficients in the multigroup problem are the $(N_g\times N_g)$ matrices defined by

$$\mathbb{D}_1=\frac{1}{3}\boldsymbol{\Sigma}_1^{-1}\,,\quad \mathbb{D}_2=\frac{1}{7}\boldsymbol{\Sigma}_3^{-1}\,,\quad \mathbb{D}_3=\frac{1}{11}\boldsymbol{\Sigma}_5^{-1}\,,\quad \mathbb{D}_4=\frac{1}{15}\boldsymbol{\Sigma}_7^{-1}\,. \tag{5.37}$$

The source is a $(N_g\times 1)$ column vector for each moment,

$$\mathbb{Q}_1=\mathbf{q}\,,\quad \mathbb{Q}_2=-\frac{2}{3}\mathbf{q}\,,\quad \mathbb{Q}_3=\frac{8}{15}\mathbf{q}\,,\quad \mathbb{Q}_4=-\frac{16}{35}\mathbf{q}\,. \tag{5.38}$$

Defining the coefficient matrices

$$\mathbf{c}^{(1)} = \begin{pmatrix} 1 & -\frac{2}{3} & \frac{8}{15} & -\frac{16}{35} \\ -\frac{2}{3} & \frac{4}{9} & -\frac{16}{45} & \frac{32}{105} \\ \frac{8}{15} & -\frac{16}{45} & \frac{64}{225} & -\frac{128}{525} \\ -\frac{16}{35} & \frac{32}{105} & -\frac{128}{525} & \frac{256}{1225} \end{pmatrix}, \tag{5.39}$$

$$\mathbf{c}^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{5}{9} & -\frac{4}{9} & \frac{8}{24} \\ 0 & -\frac{4}{9} & \frac{16}{45} & -\frac{32}{105} \\ 0 & \frac{32}{105} & -\frac{32}{105} & \frac{64}{245} \end{pmatrix}, \tag{5.40}$$

$$\mathbf{c}^{(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{9}{25} & -\frac{54}{175} \\ 0 & 0 & -\frac{45}{175} & \frac{324}{1225} \end{pmatrix}, \tag{5.41}$$

$$\mathbf{c}^{(4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{13}{49} \end{pmatrix}, \tag{5.42}$$

we can write the blocks of the $\mathbb{A}$ matrix as

$$\mathbb{A}_{nm} = \sum_{i=1}^{4} \mathbf{c}_{nm}^{(i)} \mathbf{\Sigma}_i \,. \tag{5.43}$$

Equation (5.36) is the form of the SP_N equations that we will use in the remainder of this paper. Setting $\Phi_2 = \Phi_4 = \Phi_6 = 0$ gives the SP_1 equation,

$$-\nabla \cdot \frac{1}{3} \mathbf{\Sigma}_1^{-1} \nabla \Phi_0 + \mathbf{\Sigma}_0 \Phi_0 = \mathbf{q} \,. \tag{5.44}$$

This equation is identical in form to the standard multigroup diffusion equation; the only difference between the SP_1 and multigroup diffusion equations is that the off-diagonal terms in $\mathbf{\Sigma}_1$ are retained in the SP_1 equation. Equivalently, the SP_3 equations are obtained by setting $\Phi_4 = \Phi_6 = 0$, and the SP_5 equations result from setting $\Phi_6 = 0$.

The P_7 Marshak boundary conditions are obtained by carrying out the integrations in Eq. (5.15) to produce

$$\begin{aligned} \frac{1}{2}\Phi_0 + \Phi_1 + \frac{5}{8}\Phi_2 - \frac{3}{16}\Phi_4 + \frac{13}{128}\Phi_6 &= \frac{1}{2}\mathbf{s} \,, \\ -\frac{1}{8}\Phi_0 + \frac{5}{8}\Phi_2 + \Phi_3 + \frac{81}{128}\Phi_4 - \frac{13}{64}\Phi_6 &= -\frac{1}{8}\mathbf{s} \,, \\ \frac{1}{16}\Phi_0 - \frac{25}{128}\Phi_2 + \frac{81}{128}\Phi_4 + \Phi_5 + \frac{325}{512}\Phi_6 &= \frac{1}{16}\mathbf{s} \,, \\ -\frac{5}{128}\Phi_0 + \frac{7}{64}\Phi_2 - \frac{105}{512}\Phi_4 + \frac{325}{512}\Phi_6 + \Phi_7 &= -\frac{5}{128}\mathbf{s} \,, \end{aligned} \tag{5.45}$$

where

$$\mathbf{s} = \begin{pmatrix} \phi_{\mathrm{b}}^0 & \phi_{\mathrm{b}}^1 & \dots & \phi_{\mathrm{b}}^G \end{pmatrix}^T \,. \tag{5.46}$$

These equations are converted into SP_N boundary conditions using the same procedure that was used to form Eq. (5.57). Eq. (5.31) is used to remove the odd moments ($\{\Phi_{1,i}, \Phi_{3,i}, \Phi_{5,i}, \Phi_{7,i}\}$) from Eq. (5.45), and

the SP_N boundary approximation,

$$\pm\frac{\partial}{\partial x} \rightarrow \hat{\mathbf{n}}\cdot\nabla\,,$$

is used for the gradient terms. Finally, Eqs. (5.34) and (5.35) are used to transform the resulting system into $\{\mathbb{U}_1 \ldots \mathbb{U}_4\}$. Application of these steps gives the SP_N Marshak boundary conditions,

$$-\,\hat{\mathbf{n}}\cdot\mathbb{J}_n + \sum_{m=1}^{4}\mathbb{B}_{nm}\mathbb{U}_m = \mathbb{S}_n\,, \tag{5.47}$$

where

$$\mathbb{S}_1 = \frac{1}{2}\mathbf{s}\,,\quad \mathbb{S}_2 = -\frac{1}{8}\mathbf{s}\,,\quad \mathbb{S}_3 = \frac{1}{16}\mathbf{s}\,,\quad \mathbb{S}_4 = -\frac{5}{128}\mathbf{s}\,. \tag{5.48}$$

The moments are coupled on the boundary through the $\mathbb{B}$ matrix, each block of which is an $(N_g \times N_g)$ multiple of the identity matrix, i.e.,

$$\mathbb{B}_{nm} = \mathbf{b}_{nm}\mathbf{I}_{N_g}\,, \tag{5.49}$$

where $\mathbf{b}_{nm}$ is the (n,m) entry in the coefficient matrix

$$\mathbf{b} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{8} & \frac{1}{16} & -\frac{5}{128} \\ -\frac{1}{8} & \frac{7}{24} & -\frac{41}{384} & \frac{1}{16} \\ \frac{1}{16} & -\frac{41}{384} & \frac{407}{1920} & -\frac{233}{2560} \\ -\frac{5}{128} & \frac{1}{16} & -\frac{233}{2560} & \frac{3023}{17920} \end{pmatrix}\,. \tag{5.50}$$

The current, $\mathbb{J}_n$, is a length N_g vector; it is related to the flux by Fick's Law,

$$\mathbb{J}_n = -\mathbb{D}_n\nabla\mathbb{U}_n\,. \tag{5.51}$$

The P_N boundary conditions for reflecting surfaces are given in Eq. (5.21). Applying the SP_N approximation to these boundary conditions yields

$$\nabla\mathbb{U}_n = 0\,,\quad n = 1,2,3,4\,. \tag{5.52}$$

This implies that $\hat{\mathbf{n}}\cdot\mathbb{J} = 0$ on the boundaries.

In summary, the SP_N equations are given in Eq. (5.57) and yield $(N+1)/2$ second-order equations. The SP_N Marshak boundary conditions are given in Eq. (5.47) for vacuum boundaries. Equation (5.52) gives reflecting boundary conditions. Each boundary condition yields $(N+1)/2$ first-order (Robin) conditions that closes the system of SP_N equations.

We note that this is not the only formation of the SP_N equations. References [**9**], [**12**], and [**13**] derive a canonical form of the SP_N equations that is based on the equivalence of the one-dimensional, planar P_N and S_{N+1} (discrete ordinates) equations. Starting from the S_{N+1} equations and using even-odd parity expansions of the angular flux, a system of SP_N equations is derived that is algebraically identical to the SP_N equations presented here. The principal advantages of this approach are that the fluxes are uncoupled at the boundary.

3. Eigenvalue Form of the SP_N Equations

The eigenvalue form of the 1-D transport equation, Eq. (5.23), is

$$\mu\frac{\partial\psi^g(x,\mu)}{\partial x} + \sigma^g(x)\psi^g(x,\mu) = \sum_{g'=0}^{G}\int_{4\pi}\sigma_{\mathrm{s}}^{gg'}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi^{g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}' + \frac{1}{k}\sum_{g'=0}^{G}\frac{\chi^g}{4\pi}\int_{4\pi}\nu\sigma_{\mathrm{f}}^{g'}(x)\,\psi^{g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'\,. \tag{5.53}$$

Expanding the eigenvalue term using the Eq. (5.3) and applying the orthogonalization property in Eq. (5.5) yields

$$\begin{aligned}\frac{1}{k}\sum_{g'=0}^{G}\frac{\chi^g}{4\pi}\int_{4\pi}\nu\sigma_{\mathrm{f}}^{g'}(x)\,\psi^{g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}' &= \frac{1}{k}\sum_{g'=0}^{G}\frac{\chi^g}{2}\int_{-1}^{1}\nu\sigma_{\mathrm{f}}^{g'}\left[\sum_n\frac{2n+1}{4\pi}\phi_n P_n(\mu)\right]d\mu' \\ &= \frac{1}{k}\sum_{g'=0}^{G}\frac{\chi^g}{4\pi}\nu\sigma_{\mathrm{f}}^{g'}\phi_n^{g'}\delta_{n0}\,.\end{aligned} \tag{5.54}$$

The eigenvalue form of the P_N equations proceeds by using this term for the source term in Eq. (5.10). Applying the SP_N approximation described in § 2.1 to the resulting multigroup, eigenvalue P_N equations gives

$$\begin{aligned}-\frac{\partial}{\partial x_i}\Bigg[\frac{n}{2n+1}(\mathbf{\Sigma}_{n-1}^{-1})\frac{\partial}{\partial x_i}\Big(\frac{n-1}{2n-1}\Phi_{n-2}+\frac{n}{2n-1}\Phi_n\Big)+ \\ \frac{n+1}{2n+1}(\mathbf{\Sigma}_{n+1}^{-1})\frac{\partial}{\partial x_i}\Big(\frac{n+1}{2n+3}\Phi_n+\frac{n+2}{2n+3}\Phi_{n+2}\Big)\Bigg]+ \\ \mathbf{\Sigma}_n\Phi_n=\frac{1}{k}\mathbf{F}\Phi_n\delta_{n0}\,,\quad m=0,2,\ldots,N\,.\end{aligned} \tag{5.55}$$

The fission matrix, $\mathbf{F}$ is defined

$$\mathbf{F}=\begin{pmatrix}\chi^0\nu\sigma_{\mathrm{f}}^0 & \chi^0\nu\sigma_{\mathrm{f}}^1 & \ldots & \chi^0\nu\sigma_{\mathrm{f}}^G \\ \chi^1\nu\sigma_{\mathrm{f}}^0 & \chi^1\nu\sigma_{\mathrm{f}}^1 & \ldots & \chi^1\nu\sigma_{\mathrm{f}}^G \\ \vdots & \vdots & \ddots & \vdots \\ \chi^G\nu\sigma_{\mathrm{f}}^0 & \chi^G\nu\sigma_{\mathrm{f}}^1 & \ldots & \chi^G\nu\sigma_{\mathrm{f}}^G\end{pmatrix}\,. \tag{5.56}$$

Converting the state unknowns from $\Phi\rightarrow\mathbb{U}$ via Eq. (5.34) gives the following eigensystem,

$$-\nabla\cdot\mathbb{D}_n\nabla\mathbb{U}_n+\sum_{m=1}^{4}\mathbb{A}_{nm}\mathbb{U}_m=\frac{1}{k}\sum_{m=1}^{4}\mathbb{F}_{nm}\mathbb{U}_{nm}\,,\quad n=1,2,3,4\,, \tag{5.57}$$

where

$$\mathbb{F}=\begin{pmatrix}\mathbf{F} & -\frac{2}{3}\mathbf{F} & \frac{8}{15}\mathbf{F} & -\frac{16}{35}\mathbf{F} \\ -\frac{2}{3}\mathbf{F} & \frac{4}{9}\mathbf{F} & -\frac{16}{45}\mathbf{F} & \frac{32}{105}\mathbf{F} \\ \frac{8}{15}\mathbf{F} & -\frac{16}{45}\mathbf{F} & \frac{64}{225}\mathbf{F} & -\frac{128}{525}\mathbf{F} \\ -\frac{16}{35}\mathbf{F} & \frac{32}{105}\mathbf{F} & -\frac{128}{525}\mathbf{F} & \frac{256}{1225}\mathbf{F}\end{pmatrix}\,. \tag{5.58}$$

4. Adjoint Form of the SP_N Equations

The adjoint form of the 1-D transport equation, Eq. (5.23), is

$$-\mu\frac{\partial\psi^{\dagger\,g}(x,\mu)}{\partial x}+\sigma^g(x)\psi^{\dagger\,g}(x,\mu)=\sum_{g'=0}^{G}\int_{4\pi}\sigma_{\mathrm{s}}^{g'g}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi^{\dagger\,g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'+\frac{q^{\dagger\,g}(x)}{4\pi}\,, \tag{5.59}$$

where $\psi^{\dagger\, g}$ is the adjoint flux for group g. Likewise, the adjoint form of Eq. (5.53) is

$$-\mu\frac{\partial\psi^{\dagger\, g}(x,\mu)}{\partial x}+\sigma^{g}(x)\psi^{\dagger\, g}(x,\mu)= \sum_{g'=0}^{G}\int_{4\pi}\sigma_{\mathrm{s}}^{g'g}(x,\hat{\mathbf{\Omega}}\cdot\hat{\mathbf{\Omega}}')\psi^{\dagger\, g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'+\frac{1}{k^{\dagger}}\sum_{g'=0}^{G}\frac{\chi^{g'}}{4\pi}\int_{4\pi}\nu\sigma_{\mathrm{f}}^{g}(x)\,\psi^{\dagger\, g'}(x,\mathbf{\Omega}')\,d\mathbf{\Omega}'\,. \quad (5.60)$$

Applying the P_N approximation to Eq. (5.59) gives

$$-\frac{\partial}{\partial x}\Big[\frac{n}{2n+1}\phi_{n-1}^{\dagger\, g}+\frac{n+1}{2n+1}\phi_{n+1}^{\dagger\, g}\Big]+\sum_{g'}(\sigma^{g}\delta_{gg'}-\sigma_{sn}^{g'g})\phi_{n}^{\dagger\, g'}=q^{\dagger\, g}\delta_{n0}\,,\quad n=0,1,2,\ldots,N\,. \quad (5.61)$$

Following the steps in § 2.1, the adjoint SP_N equations are

$$-\frac{\partial}{\partial x_i}\Big[\frac{n}{2n+1}\phi_{n-1,i}^{\dagger\, g}+\frac{n+1}{2n+1}\phi_{n+1,i}^{\dagger\, g}\Big]+\sum_{g'}(\sigma^{g}\delta_{gg'}-\sigma_{sn}^{g'g})\phi_{n}^{\dagger\, g'}=q^{\dagger\, g}\delta_{n0}\,,\quad n=0,2,4,\ldots,N\,, \quad (5.62)$$

$$-\frac{\partial}{\partial x_i}\Big[\frac{n}{2n+1}\phi_{n-1}^{\dagger\, g}+\frac{n+1}{2n+1}\phi_{n+1}^{\dagger\, g}\Big]+\sum_{g'}(\sigma^{g}\delta_{gg'}-\sigma_{sn}^{g'g})\phi_{n,i}^{\dagger\, g'}=0\,,\quad n=1,3,5,\ldots,N\,. \quad (5.63)$$

Using Eq. (5.63) to solve for the $\phi_{n,i}^{\dagger\, g}$ terms and substituting into Eq. (5.63) gives

$$-\frac{\partial}{\partial x_i}\Bigg[\frac{n}{2n+1}(\mathbf{\Sigma}_{n-1}^{\dagger})^{-1}\frac{\partial}{\partial x_i}\Big(\frac{n-1}{2n-1}\Phi_{n-2}^{\dagger}+\frac{n}{2n-1}\Phi_{n}^{\dagger}\Big)+ \frac{n+1}{2n+1}(\mathbf{\Sigma}_{n+1}^{\dagger})^{-1}\frac{\partial}{\partial x_i}\Big(\frac{n+1}{2n+3}\Phi_{n}^{\dagger}+\frac{n+2}{2n+3}\Phi_{n+2}^{\dagger}\Big)\Bigg]+ \mathbf{\Sigma}_{n}^{\dagger}\Phi_{n}^{\dagger}=\mathbf{q}^{\dagger}\delta_{n0}\,,\quad m=0,2,\ldots,N\,, \quad (5.64)$$

where

$$\Phi_{n}^{\dagger}=\begin{pmatrix}\phi_{n}^{\dagger\, 0} & \phi_{n}^{\dagger\, 1} & \ldots & \phi_{n}^{\dagger\, G}\end{pmatrix}^{T}\,, \quad (5.65)$$

$$\Phi_{n,i}^{\dagger}=\begin{pmatrix}\phi_{n,i}^{\dagger\, 0} & \phi_{n,i}^{\dagger\, 1} & \ldots & \phi_{n,i}^{\dagger\, G}\end{pmatrix}^{T}\,, \quad (5.66)$$

$$\mathbf{q}^{\dagger}=\begin{pmatrix}q^{\dagger\, 0} & q^{\dagger\, 1} & \ldots & q^{\dagger\, G}\end{pmatrix}^{T}\,, \quad (5.67)$$

and

$$\mathbf{\Sigma}_{n}^{\dagger}=\begin{pmatrix}(\sigma^{0}-\sigma_{sn}^{00}) & -\sigma_{sn}^{10} & \ldots & -\sigma_{sn}^{G0}\\ -\sigma_{sn}^{01} & (\sigma^{1}-\sigma_{sn}^{11}) & \ldots & -\sigma_{sn}^{G1}\\ \vdots & \vdots & \ddots & \vdots\\ -\sigma_{sn}^{0G} & -\sigma_{sn}^{1G} & \ldots & (\sigma^{G}-\sigma_{sn}^{GG})\end{pmatrix}\,. \quad (5.68)$$

Equations (5.64) through (5.68) constitute the adjoint, multigroup SP_N equations.

Equation (5.64) is identical in form to Eq. (5.32); thus, all of the machinery that was derived to solve the multigroup SP_N equations in § 2.1, starting with Eq. (5.34), can be used to solve the adjoint SP_N equations. The only requirements to convert the forward solver to an adjoint solver are:

(1) use an adjoint external source (response)
(2) take the transpose of the all of the cross section matrices because

$$\mathbf{\Sigma}_{n}^{\dagger}=\mathbf{\Sigma}_{n}^{T}\,. \quad (5.69)$$

For eigenvalue equations the fission matrix must be transposed as well because $\mathbf{F}^\dagger = \mathbf{F}^T$,

$$\mathbf{F}^\dagger = \begin{pmatrix} \chi^0\nu\sigma_{\mathrm{f}}^0 & \chi^1\nu\sigma_{\mathrm{f}}^0 & \dots & \chi^G\nu\sigma_{\mathrm{f}}^0 \\ \chi^0\nu\sigma_{\mathrm{f}}^1 & \chi^1\nu\sigma_{\mathrm{f}}^1 & \dots & \chi^G\nu\sigma_{\mathrm{f}}^1 \\ \vdots & \vdots & \ddots & \vdots \\ \chi^0\nu\sigma_{\mathrm{f}}^G & \chi^1\nu\sigma_{\mathrm{f}}^G & \dots & \chi^G\nu\sigma_{\mathrm{f}}^G \end{pmatrix}. \tag{5.70}$$

All other aspects of solving the adjoint eigenvalue form of the SP_N equations follows from § 3.

CHAPTER 6

Spatial Discretizations of the S_N Equations

Denovo provides the following spatial differencing options:

(1) weighted diamond difference (WDD) in 3D
(2) weighted diamond difference with zero flux-fixup (WDD-FF) in 3D
(3) theta-weighted diamond difference (TWD) in 3D
(4) linear-discontinuous galerkin finite element (LD) in 3D
(5) bilinear-discontinuous galerkin finite element (BLD) in 2D
(6) trilinear-discontinuous galerkin finite element (TLD) in 3D
(7) step characteristics (slice balance) (SC) in 2 and 3D

The transport operator can be inverted for all of the spatial schemes using identical sweeping strategies (described in Chap. 8).

These schemes fall into two general classifications: cell-balance and finite element schemes. The diamond-difference schemes—WDD, WDD-FF, TWD—are all cell-balance schemes [**2**, **14**]. TLD, BLD, and LD are finite element schemes [**15**]. The SC method can be written as a cell-balance scheme [**16**] or a finite element scheme [**17**]. The WDD, WDD-FF, TWD, LD, BLD, and TLD schemes are all second-order, and the SC scheme is first-order.

All the spatial schemes derived below are discretized on the mesh cell illustrated in Fig. 6.1. Also, for any given group, angle, and source, Eq. (4.34) can be reduced to

$$\hat{\mathbf{\Omega}} \cdot \nabla \psi(\mathbf{r}) + \sigma(\mathbf{r})\psi(\mathbf{r}) = s(\mathbf{r}) \,, \tag{6.1}$$

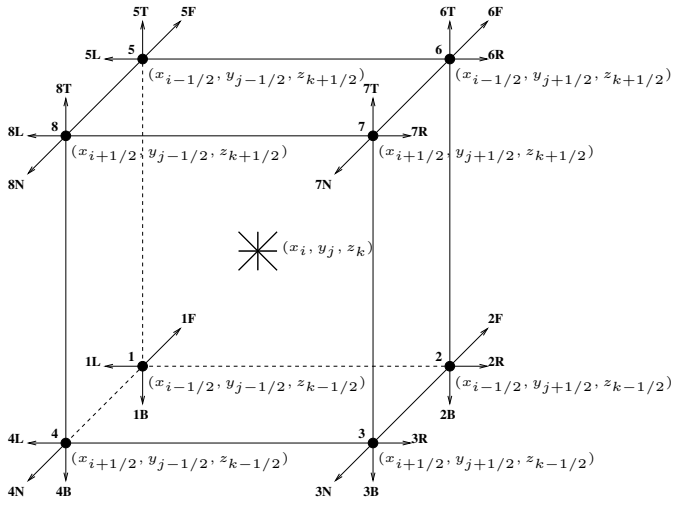


FIGURE 6.1. General mesh cell in Denovo used to derive discrete spatial equations. The adjacent cell points are given using the notation $N \to +x$, $F \to -x$, $L \to -y$, $R \to +y$, $B \to -z$, and $T \to +z$.

where $s(\mathbf{r})$ is a total accumulated source. In operator form, this equation is

$$\mathbf{L}\psi = s\,, \tag{6.2}$$

where $\mathbf{L}$ is the differential transport operator. We will be required to perform operations of the type

$$\psi = \mathbf{L}^{-1}s\,, \tag{6.3}$$

to solve discrete forms of Eq. (6.2). For all the spatial differencing schemes discussed below, $\mathbf{L}$ can be implicitly formed as a lower-left triangular matrix and inverted by "sweeping" through the mesh in the direction of particle flow. In effect, the discretized form of Eq. (6.1) is solved in each cell. The outgoing fluxes become input to the downwind cells, or in other words, each cell looks "upwind" to find its incoming fluxes. Once all the incoming fluxes are defined on the entering faces of a cell, the outgoing fluxes can be calculated, and the process is repeated until the entire mesh is solved for a given angle. For each cell the entering and exiting faces are defined by

$$\hat{\mathbf{\Omega}}\cdot\mathbf{n} < 0\,, \quad \text{(entering face)} \tag{6.4}$$

$$\hat{\mathbf{\Omega}}\cdot\mathbf{n} > 0\,, \quad \text{(exiting face)}\,. \tag{6.5}$$

Mathematically, this is called a *wavefront* solver. The operation $\mathbf{L}^{-1}$ is regularly referred to as a *sweep* in the nuclear engineering and transport communities. We will utilize this terminology repeatedly in Chap. 8.

1. Cell-Balance Schemes

The cell-balance equation can be derived by integrating Eq. (6.1) over the mesh cell in Fig. 6.1. Applying $\iiint(\cdot)\,dxdydz$ gives,

$$\frac{\mu}{\Delta_i}(\psi_{i+1/2}-\psi_{i-1/2}) + \frac{\eta}{\Delta_j}(\psi_{j+1/2}-\psi_{j-1/2}) + \frac{\xi}{\Delta_k}(\psi_{k+1/2}-\psi_{k-1/2}) + \sigma_{ijk}\psi_{ijk} = s_{ijk}\,. \tag{6.6}$$

The derivation of the WDD equations is described in detail in Ref. [**2**]. The diamond-difference method is derived by closing Eq. (6.6) with the average of the face-edge fluxes, which is equivalent to a Crank-Nicolson method in space. Solving the cell-balance equation with this closure yields the following system of equations,

$$\begin{aligned}
&\mu \gtrless 0\,,\ \eta \gtrless 0\,,\ \xi \gtrless 0\\
&\qquad \psi_{ijk} = \frac{s_{ijk} + \frac{2}{(1\pm\alpha_i)}\frac{|\mu|}{\Delta_i}\bar{\psi}_{i\mp1/2} + \frac{2}{(1\pm\alpha_j)}\frac{|\eta|}{\Delta_j}\bar{\psi}_{j\mp1/2} + \frac{2}{(1\pm\alpha_k)}\frac{|\xi|}{\Delta_k}\bar{\psi}_{k\mp1/2}}{\sigma_{ijk} + \frac{2}{(1\pm\alpha_i)}\frac{|\mu|}{\Delta_i} + \frac{2}{(1\pm\alpha_j)}\frac{|\eta|}{\Delta_j} + \frac{2}{(1\pm\alpha_k)}\frac{|\xi|}{\Delta_k}}\,,\\
&\qquad \psi_{i\pm1/2} = \frac{2}{(1\pm\alpha_i)}\psi_{ijk} - \frac{(1\mp\alpha_i)}{(1\pm\alpha_i)}\bar{\psi}_{i\mp1/2}\,,\\
&\qquad \psi_{j\pm1/2} = \frac{2}{(1\pm\alpha_j)}\psi_{ijk} - \frac{(1\mp\alpha_j)}{(1\pm\alpha_j)}\bar{\psi}_{j\mp1/2}\,,\\
&\qquad \psi_{k\pm1/2} = \frac{2}{(1\pm\alpha_k)}\psi_{ijk} - \frac{(1\mp\alpha_k)}{(1\pm\alpha_k)}\bar{\psi}_{k\mp1/2}\,.
\end{aligned} \tag{6.7}$$

Here, the $\bar{\psi}$ are the incoming fluxes on each face. The α terms are weighting factors such that $\alpha = 0$ gives the classic diamond-difference equations and $\alpha = \pm1$ gives the step-difference equations. Setting $\alpha = \pm1$ yields a first-order spatial differencing scheme. The default behavior of Denovo for WDD uses $\alpha = 0$, which gives the diamond-difference method.

Denovo provides a version of WDD that can correct negative fluxes. When the outgoing flux is less than zero, we set the face-edge flux to zero and recalculate ψ_{ijk} and the new edge fluxes. This process is repeated until all the outgoing fluxes are greater than or equal to zero. This method is nonlinear in that the corrected solution of Eq. (6.7) depends on ψ.

Another nonlinear cell-balance scheme is TWD [**14**]. This scheme uses the incoming fluxes to calculate weighting factors that permit the calculation of cell-centered and outgoing fluxes that vary smoothly between

the step and diamond-difference approximations. The weighting factors are calculated from the following system of equations,

$$\mu \gtrless 0\ ,\ \eta \gtrless 0\ ,\ \xi \gtrless 0$$

$$\begin{aligned} 1-a &= \frac{s_{ijk}V\theta_s + (|\eta|B\bar{\psi}_{j\mp 1/2} + |\xi|C\bar{\psi}_{k\mp 1/2})\theta + |\mu|A\bar{\psi}_{i\mp 1/2}}{(\sigma_{ijk}V + 2|\eta|B + 2|\xi|C)\bar{\psi}_{i\mp 1/2}}\ , \\ 1-b &= \frac{s_{ijk}V\theta_s + (|\mu|A\bar{\psi}_{i\mp 1/2} + |\xi|C\bar{\psi}_{k\mp 1/2})\theta + |\eta|B\bar{\psi}_{j\mp 1/2}}{(\sigma_{ijk}V + 2|\mu|A + 2|\xi|C)\bar{\psi}_{j\mp 1/2}}\ , \\ 1-c &= \frac{s_{ijk}V\theta_s + (|\mu|A\bar{\psi}_{i\mp 1/2} + |\eta|B\bar{\psi}_{j\mp 1/2})\theta + |\xi|C\bar{\psi}_{k\mp 1/2}}{(\sigma_{ijk}V + 2|\mu|A + 2|\eta|B)\bar{\psi}_{k\mp 1/2}}\ , \end{aligned} \tag{6.8}$$

where

$$A = \Delta_j\Delta_k\ ,\quad B = \Delta_i\Delta_k\ ,\quad C = \Delta_i\Delta_j\ ,\quad V = \Delta_i\Delta_j\Delta_k\ . \tag{6.9}$$

The theta-weighting factors, θ and θ_s, are set to values between 0 and 1. By default Denovo uses the *theta-weighted* model from the TORT code in which $\theta = \theta_s = 0.9$ [**18**].

The weighting parameters are bounded between the diamond-difference and step approximations,

$$\frac{1}{2} \le \{a, b, c\} \le 1\ . \tag{6.10}$$

Using these factors, the cell-centered and outgoing fluxes are

$$\begin{aligned} \psi_{ijk} &= \frac{s_{ijk}V + \frac{|\mu|A}{a}\bar{\psi}_{i\mp 1/2} + \frac{|\eta|B}{b}\bar{\psi}_{j\mp 1/2} + \frac{|\xi|C}{c}\bar{\psi}_{k\mp 1/2}}{s_{ijk}V + \frac{|\mu|A}{a} + \frac{|\eta|B}{b} + \frac{|\xi|C}{c}}\ , \\ \psi_{i\pm 1/2} &= \frac{1}{a}\psi_{ijk} - \frac{(1-a)}{a}\bar{\psi}_{i\mp 1/2}\ , \\ \psi_{j\pm 1/2} &= \frac{1}{b}\psi_{ijk} - \frac{(1-b)}{b}\bar{\psi}_{j\mp 1/2}\ , \\ \psi_{k\pm 1/2} &= \frac{1}{c}\psi_{ijk} - \frac{(1-c)}{c}\bar{\psi}_{k\mp 1/2}\ . \end{aligned} \tag{6.11}$$

TWD and WDD-FF produce uniformly positive fluxes when the source, s_{ijk}, is greater than zero.

2. Finite Element Schemes

Finite element schemes are derived from the weak form of Eq. (6.1) [**19**]. We begin the derivation by integrating Eq. (6.1) over a single element, e, and multiplying by a weighting function for the element, $w_{e\,n}$,

$$\int_{V_e} w_{e\,n}(\hat{\mathbf{\Omega}}\cdot\nabla\psi + \sigma\psi)\,dV = \int_{V_e} w_{e\,n}s\,dV\ , \tag{6.12}$$

where the element defined in Fig. 6.1 has $dV = dxdydz$. The weight functions are defined over the range $n \in [1, N]$ such that the set $w_{e\,n}$ is linearly independent. Furthermore, we make the assumption that all elements in the orthogonal grid have equal weight functions, so $w_{e\,n} \equiv w_n$. Applying Green's theorem to the gradient term gives the weak form of the transport equation,

$$\oint_{\partial V_e} w_n\hat{\mathbf{n}}\cdot\hat{\mathbf{\Omega}}\psi\,dA - \int_{V_e}\hat{\mathbf{\Omega}}\cdot\nabla w_n\psi\,dV + \int_{V_e} w_n\sigma\psi\,dV_e = \int_{V_e} w_n s\,dV\ ,\quad n = 1,\ldots,N\ . \tag{6.13}$$

Now, we expand the angular flux in the following basis,

$$\psi = \sum_{m=1}^{N} b_m(\mathbf{r})\psi_e^{(m)}\ ,\quad \mathbf{r}\in\partial V_e\ . \tag{6.14}$$

Applying the Galerkin finite element approximation in which $w_n = b_n$, Eq. (6.13) becomes

$$\oint_{\partial V_e} b_n\hat{\mathbf{n}}\cdot\hat{\mathbf{\Omega}}\psi\,dA - \mathbb{T}\Psi + \sigma_e\mathbb{M}\Psi = \mathbb{M}S\ , \tag{6.15}$$

where the elements of the matrices and vectors are defined

$$\begin{aligned}
[\mathbb{T}]_{nm} &= \int_{V_e} \hat{\mathbf{\Omega}} \cdot \nabla b_n b_m \, dV \,, \\
[\mathbb{M}]_{nm} &= \int_{V_e} b_n b_m \, dV \,, \\
\Psi &= \begin{pmatrix} \psi_e^{(1)} & \psi_e^{(2)} & \cdots & \psi_e^{(N)} \end{pmatrix}^T \,, \\
S &= \begin{pmatrix} s_e^{(1)} & s_e^{(2)} & \cdots & s_e^{(N)} \end{pmatrix}^T \,,
\end{aligned} \tag{6.16}$$

and the source has been expanded in the same basis as the angular flux. The angular fluxes in the surface term come from upwind cells or boundary conditions.

The size and composition of the matrices in Eq. (6.15) are dependent on the mesh and the basis functions chosen to represent the unknowns. The LD method defines a basis over the set $\{1, x, y, z\}$ with the shape functions [**15**],

$$b_1 = 1 \,, \quad b_2 = \frac{2(x - x_{ijk})}{\Delta_i} \,, \quad b_3 = \frac{2(y - y_{ijk})}{\Delta_j} \,, \quad b_4 = \frac{2(z - z_{ijk})}{\Delta_k} \,. \tag{6.17}$$

Using these shape functions in Eq. (6.15) and analytically evaluating the integrals give the following system of equations,

$$\begin{aligned}
\frac{\mu}{\Delta_i}(\psi_{i+1/2} - \psi_{i-1/2}) + \frac{\eta}{\Delta_j}(\psi_{j+1/2} - \psi_{j-1/2}) + \frac{\xi}{\Delta_k}(\psi_{k+1/2} - \psi_{k-1/2}) + \sigma\psi^a &= s^a \,, \\
\frac{3\mu}{\Delta_i}(\psi_{i+1/2} + \psi_{i-1/2} - 2\psi^a) + \frac{\eta}{\Delta_j}(\psi^x_{j+1/2} - \psi^x_{j-1/2}) + \frac{\xi}{\Delta_k}(\psi^x_{k+1/2} - \psi^x_{k-1/2}) + \sigma\psi^x &= s^x \,, \\
\frac{\mu}{\Delta_i}(\psi^y_{i+1/2} - \psi^y_{i-1/2}) + \frac{3\eta}{\Delta_j}(\psi_{j+1/2} + \psi_{j-1/2} - 2\psi^a) + \frac{\xi}{\Delta_k}(\psi^y_{k+1/2} - \psi^y_{k-1/2}) + \sigma\psi^y &= s^y \,, \\
\frac{\mu}{\Delta_i}(\psi^z_{i+1/2} - \psi^z_{i-1/2}) + \frac{\eta}{\Delta_j}(\psi^z_{j+1/2} - \psi^z_{j-1/2}) + \frac{3\xi}{\Delta_k}(\psi_{k+1/2} + \psi_{k-1/2} - 2\psi^a) + \sigma\psi^z &= s^z \,.
\end{aligned} \tag{6.18}$$

Here,

$$\psi_e^{(1)} = \psi^a \,, \quad \psi_e^{(2)} = \psi^x \,, \quad \psi_e^{(3)} = \psi^y \,, \quad \psi_e^{(4)} = \psi^z \,, \tag{6.19}$$

such that ψ^a is the average angular flux in the element and $\psi^{x|y|z}$ are the slopes in the (x, y, z) directions. The same notation is used for the expanded source, s.

Equations (6.14) and (6.17) are used to develop upwind expressions for ψ,

$$\psi_{i\pm1/2} = \psi^a \pm \psi^x \,, \quad \psi_{j\pm1/2} = \psi^a \pm \psi^y \,, \quad \psi_{k\pm1/2} = \psi^a \pm \psi^z \,, \quad \{\mu, \eta, \xi\} \gtrless 0 \,, \tag{6.20}$$

and the upwinded slopes are

$$\psi^{(y|z)}_{i\pm1/2} = \psi^{(y|z)} \,, \quad \psi^{(x|z)}_{j\pm1/2} = \psi^{(x|z)} \,, \quad \psi^{(x|y)}_{k\pm1/2} = \psi^{(x|y)} \,. \tag{6.21}$$

For each direction, these expressions are inserted into Eq. (6.18), and the resulting 4×4 system is solved for Ψ. For efficiency, the solution to the matrix is precomputed, and Ψ is calculated by evaluating four statements.

The TLD method is the only spatial discretization that maintains the asymptotic diffusion limit on the grid used in Denovo [**15**]. The TLD equations are derived by expanding Ψ in the basis

$$\{1, x, y, z, xy, yz, xz, xyz\} \,;$$

thus, there are eight unknowns per cell. We define the unknowns at each node of the cell as indicated by the cardinal numbers in Fig. 6.1. The basis functions are [**15**]

$$
\begin{aligned}
b_1 &= \Big(\frac{x_{i+1/2}-x}{\Delta_i}\Big)\Big(\frac{y_{j+1/2}-y}{\Delta_j}\Big)\Big(\frac{z_{k+1/2}-z}{\Delta_k}\Big)\,,\\
b_2 &= \Big(\frac{x_{i+1/2}-x}{\Delta_i}\Big)\Big(\frac{y-y_{j-1/2}}{\Delta_j}\Big)\Big(\frac{z_{k+1/2}-z}{\Delta_k}\Big)\,,\\
b_3 &= \Big(\frac{x-x_{i-1/2}}{\Delta_i}\Big)\Big(\frac{y-y_{j-1/2}}{\Delta_j}\Big)\Big(\frac{z_{k+1/2}-z}{\Delta_k}\Big)\,,\\
b_4 &= \Big(\frac{x-x_{i-1/2}}{\Delta_i}\Big)\Big(\frac{y_{j+1/2}-y}{\Delta_j}\Big)\Big(\frac{z_{k+1/2}-z}{\Delta_k}\Big)\,,\\
b_5 &= \Big(\frac{x_{i+1/2}-x}{\Delta_i}\Big)\Big(\frac{y_{j+1/2}-y}{\Delta_j}\Big)\Big(\frac{z-z_{k-1/2}}{\Delta_k}\Big)\,,\\
b_6 &= \Big(\frac{x_{i+1/2}-x}{\Delta_i}\Big)\Big(\frac{y-y_{j-1/2}}{\Delta_j}\Big)\Big(\frac{z-z_{k-1/2}}{\Delta_k}\Big)\,,\\
b_7 &= \Big(\frac{x-x_{i-1/2}}{\Delta_i}\Big)\Big(\frac{y-y_{j-1/2}}{\Delta_j}\Big)\Big(\frac{z-z_{k-1/2}}{\Delta_k}\Big)\,,\\
b_8 &= \Big(\frac{x-x_{i-1/2}}{\Delta_i}\Big)\Big(\frac{y_{j+1/2}-y}{\Delta_j}\Big)\Big(\frac{z-z_{k-1/2}}{\Delta_k}\Big)\,.
\end{aligned}
\tag{6.22}
$$

Substituting these into Eq. (6.15) and evaluating the integrals analytically, we derive the TLD equations for the cells illustrated in Fig. 6.1,

$$\frac{\mu\Delta_j\Delta_k}{36}\begin{pmatrix}-4\psi_{1F}-2\psi_{2F}-2\psi_{5F}-\psi_{6F}\\-2\psi_{1F}-4\psi_{2F}-\psi_{5F}-2\psi_{6F}\\4\psi_{3N}+2\psi_{4N}+2\psi_{7N}+\psi_{8N}\\2\psi_{3N}+4\psi_{4N}+\psi_{7N}+2\psi_{8N}\\-2\psi_{1F}-\psi_{2F}-4\psi_{5F}-2\psi_{6F}\\-\psi_{1F}-2\psi_{2F}-2\psi_{5F}-4\psi_{6F}\\2\psi_{3N}+\psi_{4N}+4\psi_{7N}+2\psi_{8N}\\\psi_{3N}+2\psi_{4N}+2\psi_{7N}+4\psi_{8N}\end{pmatrix}+\frac{\eta\Delta_i\Delta_k}{36}\begin{pmatrix}-4\psi_{1L}-2\psi_{4L}-2\psi_{5L}-\psi_{8L}\\4\psi_{2R}+2\psi_{3R}+2\psi_{6R}+\psi_{7R}\\2\psi_{2R}+4\psi_{3R}+\psi_{6R}+2\psi_{7R}\\-2\psi_{1L}-4\psi_{4L}-\psi_{5L}-2\psi_{8L}\\-2\psi_{1L}-\psi_{4L}-4\psi_{5L}-2\psi_{8L}\\2\psi_{2R}+\psi_{3R}+4\psi_{6R}+2\psi_{7R}\\\psi_{2R}+2\psi_{3R}+2\psi_{6R}+4\psi_{7R}\\-\psi_{1L}-2\psi_{4L}-2\psi_{5L}-4\psi_{8L}\end{pmatrix}$$

$$+\frac{\xi\Delta_i\Delta_j}{36}\begin{pmatrix}-4\psi_{1B}-2\psi_{2B}-\psi_{3B}-2\psi_{4B}\\-2\psi_{1B}-4\psi_{2B}-2\psi_{3B}-\psi_{4B}\\-\psi_{1B}-2\psi_{2B}-4\psi_{3B}-2\psi_{4B}\\-2\psi_{1B}-\psi_{2B}-2\psi_{3B}-4\psi_{4B}\\4\psi_{5T}+2\psi_{6T}+\psi_{7T}+2\psi_{8T}\\2\psi_{5T}+4\psi_{6T}+2\psi_{7T}+\psi_{8T}\\\psi_{5T}+2\psi_{6T}+4\psi_{7T}+2\psi_{8T}\\2\psi_{5T}+\psi_{6T}+2\psi_{7T}+4\psi_{8T}\end{pmatrix}$$

$$+\frac{\mu\Delta_j\Delta_k}{72}\begin{pmatrix}4&2&2&4&2&1&1&2\\2&4&4&2&1&2&2&1\\-2&-4&-4&-2&-1&-2&-2&-1\\-4&-2&-2&-4&-2&-1&-1&-2\\2&1&1&2&4&2&2&4\\1&2&2&1&2&4&4&2\\-1&-2&-2&-1&-2&-4&-4&-2\\-2&-1&-1&-2&-4&-2&-2&-4\end{pmatrix}\Psi \tag{6.23}$$

$$+\frac{\eta\Delta_i\Delta_k}{72}\begin{pmatrix}4&4&2&2&2&2&1&1\\-4&-4&-2&-2&-2&-2&-1&-1\\-2&-2&-4&-4&-1&-1&-2&-2\\2&2&4&4&1&1&2&2\\2&2&1&1&4&4&2&2\\-2&-2&-1&-1&-4&-4&-2&-2\\-1&-1&-2&-2&-2&-2&-4&-4\\1&1&2&2&2&2&4&4\end{pmatrix}\Psi$$

$$+\frac{\xi\Delta_i\Delta_j}{72}\begin{pmatrix}4&2&1&2&4&2&1&2\\2&4&2&1&2&4&2&1\\1&2&4&2&1&2&4&2\\2&1&2&4&2&1&2&4\\-4&-2&-1&-2&-4&-2&-1&-2\\-2&-4&-2&-1&-2&-4&-2&-1\\-1&-2&-4&-2&-1&-2&-4&-2\\-2&-1&-2&-4&-2&-1&-2&-4\end{pmatrix}\Psi+\sigma\frac{\Delta_i\Delta_j\Delta_k}{8}\Psi=\frac{\Delta_i\Delta_j\Delta_k}{8}S\,,$$

where Ψ and S are defined in Eq. (6.16) with $N=8$. Additionally, the mass matrices, $\mathbb{M}$, have been lumped.

The surface fluxes in Eq. (6.23) are upwinded from neighboring cells or are evaluated using the boundary conditions on problem boundaries. The adjacent node indices are illustrated in Fig. 6.1. Following the notation in Ref. [**15**],

$$\psi(\mathbf{r}_{el})=\begin{cases}\psi(\mathbf{r}_{el}^{-}) & \mathbf{n}_{el}\cdot\hat{\mathbf{\Omega}}>0\\ \psi(\mathbf{r}_{el}^{+}) & \mathbf{n}_{el}\cdot\hat{\mathbf{\Omega}}<0\end{cases},\quad \mathbf{r}_{el}\in\partial V_{el}\,. \tag{6.24}$$

Here, the point $\mathbf{r}_{el}^{+}$ is outside surface l of element e, and $\mathbf{r}_{el}^{-}$ is inside. Following this statement, on incoming faces we take ψ from the upwind cell (cell that the particle is exiting) or from the boundary conditions. On exiting faces, the value of ψ is set to the corresponding node value. For example, for a particle entering element e on the low x face $(-x)$ from element n, Eq. (6.24) determines the boundary fluxes on the incoming and outgoing x faces in Eq. (6.23) to be

$$\begin{aligned} &\text{incoming face:} && \psi_{1F} = \psi_n^{(4)} , && \psi_{2F} = \psi_n^{(3)} , && \psi_{5F} = \psi_n^{(8)} , && \psi_{6F} = \psi_n^{(7)} , \\ &\text{outgoing face:} && \psi_{3N} = \psi_e^{(3)} , && \psi_{4N} = \psi_e^{(4)} , && \psi_{7N} = \psi_e^{(7)} , && \psi_{8N} = \psi_e^{(8)} . \end{aligned}$$

Similar expressions are evaluated on the y and z faces. Thus, using Eq. (6.24) to define the upwind fluxes for any given direction, Eq. (6.23) is a fully determined 8×8 linear system of the form $\mathbf{P}\Psi = b$. Denovo uses the LAPACK routine `dgesv` to solve this equation.

3. Step Characteristics

The primary advantage of the SC scheme is that it produces uniformly positive solutions. SC gives positive fluxes as long as the source is positive, and it does not require non-linear flux fix-ups like WDD-FF or TWD. Also, it does not suffer from oscillatory behavior like step differencing. Because of these properties, SC is an ideal choice for calculating adjoint importance maps for use in hybrid Monte Carlo methods. However, if very high accuracy is required, LD and TLD are better choices because they are second-order methods, whereas SC is first-order.

The SC spatial discretization can be derived using the cell-balance or finite element formalism. We will use the slice-balance method originally proposed by Lathrop [**16**]. First, define a minimum step size through the cell defined in Fig. 6.1,

$$d = \min\left(\frac{\Delta_i}{|\mu|}, \frac{\Delta_j}{|\eta|}, \frac{\Delta_k}{|\xi|}\right) . \tag{6.25}$$

We now define a new set of indices, $\{i, j, k\} \rightarrow \{p, q, r\}$, such that p is associated with the index of d,

$$p = \text{index}[s(i; j; k)] , \tag{6.26}$$

and $\{q, r\}$ are associated with the remaining indices. The slice fluxes are defined

$$\begin{aligned} \psi_0^m &= \bar{s} + \mathrm{e}^{-\sigma d}(\psi_{\text{inc}}^m - \bar{s}) , \\ \psi_1^m &= \bar{s} + \frac{1}{\sigma d}(\psi_{\text{inc}}^m - \psi_0^m) , \\ \psi_2^m &= \bar{s} + \frac{2}{\sigma d}(\psi_{\text{inc}}^m - \psi_1^m) , \\ m &= p, q, r , \end{aligned} \tag{6.27}$$

and $\bar{s} = s/\sigma$. For each slice a normalized distance is defined,

$$\delta_m = \left(\frac{|\hat{\mathbf{\Omega}}|_m}{\Delta_m}\right) d , \tag{6.28}$$

such that $\delta_p = 1$ and $\delta_{q,r} \leq 1$. The areas of each slice are

$$A_{pq} = \frac{\delta_p \delta_r}{2} , \quad B_{pq} = \delta_p(1 - \delta_r) , \quad C_{pp} = (1 - \delta_q)(1 - \delta_r) . \tag{6.29}$$

By using these definitions, the outgoing fluxes are

$$\begin{aligned} \psi_p &= A_{qp}\psi_2^q + A_{rp}\psi_2^r + B_{qp}\psi_1^q + B_{rp}\psi_1^r + C_{pp}\psi_0^p , \\ \psi_q &= A_{pq}\psi_2^p + A_{rq}\psi_2^r + B_{pq}\psi_1^p , \\ \psi_r &= A_{pr}\psi_2^p + A_{qr}\psi_2^q + B_{pr}\psi_1^p . \end{aligned} \tag{6.30}$$

Finally, the cell-centered flux is

$$\psi = \bar{s} + \frac{1}{\sigma d}\sum_m \delta_m(\psi_{\text{inc}}^m - \psi_m) , \quad m = p, q, r . \tag{6.31}$$

Because these equations depend inversely on σd, we need to handle cases where $\sigma d \ll 1$, which includes vacuum regions. In Denovo when $\sigma d \leq 0.025$ we expand $\exp(-\sigma d)$ in a $O(7)$ Taylor-series. Using the expansion in Eqs. (6.27) through (6.31) yields formulas for the outgoing and cell-centered flux that do not vary inversely proportional to σd.

CHAPTER 7

Spatial Discretizations of the SP_N Equations

The multigroup SP_N equations are given in § 2. Equation (5.32) gives the fixed-source form and Eq. (5.57) gives the eigenvalue form. In this chapter we will derive spatial discretizations of these equations.

1. Finite Volume Discretization of the SP_N Equations

The general form for the SP_N equations is given in Eq. (5.57); Marshak boundary conditions are defined in Eq. (5.47); reflecting boundary conditions are given by Eq. (5.52). Applying Fick's Law [Eq. (5.51)] to Eq. (5.57) gives

$$\nabla \cdot \mathbb{J}_n + \sum_{m=1}^{4} \mathbb{A}_{nm} \mathbb{U}_m = \mathbb{Q}_n \,, \quad n = 1, 2, 3, 4 \,. \tag{7.1}$$

Because there is no spatial coupling in the fission source, we have represented it in the term $\mathbb{Q}_n$. Thus, in a fixed-source problem $\mathbb{Q}_n$ is defined by Eq. (5.38), and in an eigenvalue problem it is

$$\mathbb{Q}_n = \frac{1}{k} \sum_{m=1}^{4} \mathbb{F}_{nm} \mathbb{U}_{nm} \,. \tag{7.2}$$

The finite-volume discretization is defined on a 3D, orthogonal Cartesian grid with logical dimensions $0 < i, j, k < N_{i,j,k}$, where $N_{i,j,k}$ is the number of computational cells in i, j, or k, respectively as shown in Fig. 7.1. The finite-volume form of the equations is derived by integrating Eq. (7.1) over cell volume and applying the divergence theorem. Integrating over volume yields, with piece-wise constant A_{nm},

$$\int_V \nabla \cdot \mathbb{J}_n \, dV + \sum_{m=1}^{4} g A g_{nm,ijk} \mathbb{U}_{m,ijk} V_{ijk} = \mathbb{Q}_{n,ijk} V_{ijk} \,, \tag{7.3}$$

where

$$u_{n,ijk} = \frac{1}{V_{ijk}} \int_V u_n \, dV \,, \tag{7.4}$$

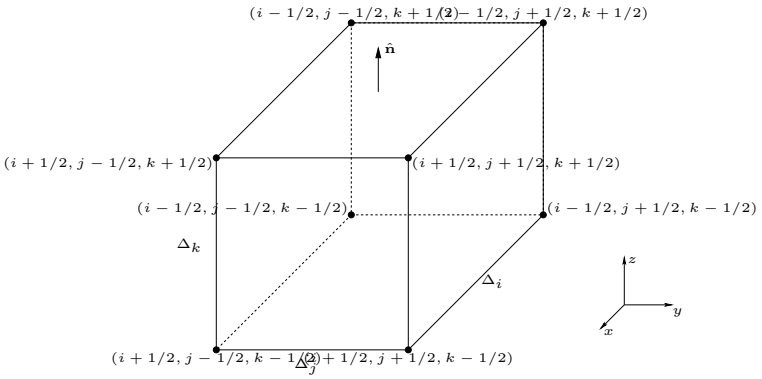


FIGURE 7.1. Three-dimensional, Cartesian mesh cell.

and

$$V_{ijk} = \Delta_i \Delta_j \Delta_k \, . \tag{7.5}$$

The Divergence Theorem gives[1]

$$\int_V \nabla \cdot \mathbb{J}_n \, dV = \oint \hat{\mathbf{n}} \cdot \mathbb{J}_n \, dA = \sum_{f=1}^{6} \hat{\mathbf{n}}_f \cdot \mathbb{J}_{n,f} A_f \, , \tag{7.6}$$

where f is the index over faces such that $f \in \{1, ..., 6\}$ as illustrated in Fig. 7.1. Applying these terms to Eq. (7.3) gives the discrete balance equation for Eq. (7.1) in cell (i, j, k),

$$\begin{aligned}(\mathbb{J}_{n,i+1/2} - \mathbb{J}_{n,i-1/2})\Delta_j \Delta_k + (\mathbb{J}_{n,j+1/2} - \mathbb{J}_{n,j-1/2})\Delta_i \Delta_k + \\ (\mathbb{J}_{n,k+1/2} - \mathbb{J}_{n,k-1/2})\Delta_i \Delta_j + \sum_{m=1}^{4} \mathbb{A}_{nm,ijk} \mathbb{U}_{m,ijk} V_{ijk} = \mathbb{Q}_{n,ijk} V_{ijk} \, .\end{aligned} \tag{7.7}$$

Here, we have written the face-edge currents with suppressed subscripts as follows:

$$\mathbb{J}_{n,i\pm 1/2\, jk} \to \mathbb{J}_{n,i\pm 1/2} \, , \quad \mathbb{J}_{n,i\, j\pm 1/2\, k} \to \mathbb{J}_{n,j\pm 1/2} \, , \quad \mathbb{J}_{n,ij\, k\pm 1/2} \to \mathbb{J}_{n,k\pm 1/2} \, .$$

The same convention will be applied to all face-edge quantities.

Applying second-order differencing to Fick's Law, Eq. (5.51), in each direction for the plus/minus faces of the computational cell gives

$$\begin{aligned}\mathbb{J}_{n,l+1/2} &= -\frac{1}{\Delta_{l+1/2}} \mathbb{D}_{n,l+1/2} (\mathbb{U}_{n,l+1} - \mathbb{U}_{n,l}) \, , \\ \mathbb{J}_{n,l-1/2} &= -\frac{1}{\Delta_{l-1/2}} \mathbb{D}_{n,l-1/2} (\mathbb{U}_{n,l} - \mathbb{U}_{n,l-1}) \, ,\end{aligned} \tag{7.8}$$

for $l = i, j, k$, and $\Delta_{l\pm 1/2} = \frac{1}{2}(\Delta_l + \Delta_{l\pm 1})$. We note here that the true, physical current is the first moment of the angular flux and is not equivalent to $\mathbb{J}_n$. Using Eq. (7.8), the balance equation (7.7) can be written in terms of the unknowns $\mathbb{U}$; however, the cell-edge diffusion coefficients must be defined. To make the method consistent, the moments and their first derivatives must be continuous at inter-cell boundaries. This condition implies that the effective currents, $\mathbb{J}$, are continuous across cell boundaries, i.e.,

$$\begin{aligned}\mathbb{J}^-_{n,l+1/2} &= \mathbb{J}^+_{n,l+1/2} \, , \\ -2\mathbb{D}_{n,l} \frac{\mathbb{U}_{n,l+1/2} - \mathbb{U}_{n,l}}{\Delta_l} &= -2\mathbb{D}_{n,l+1} \frac{\mathbb{U}_{n,l+1} - \mathbb{U}_{n,l+1/2}}{\Delta_{l+1}} \, ,\end{aligned} \tag{7.9}$$

and

$$\begin{aligned}\mathbb{J}^-_{n,l-1/2} &= \mathbb{J}^+_{n,l-1/2} \, , \\ -2\mathbb{D}_{n,l-1} \frac{\mathbb{U}_{n,l-1/2} - \mathbb{U}_{n,l-1}}{\Delta_{l-1}} &= -2\mathbb{D}_{n,l} \frac{\mathbb{U}_{n,l} - \mathbb{U}_{n,l-1/2}}{\Delta_l} \, .\end{aligned} \tag{7.10}$$

Solving for the $(N_g \times N_g)$ cell-edge unknowns, $\{\mathbb{U}_{n,l\pm 1/2}\}$, substituting these values back into $\mathbb{J}^+_{n,l\pm 1/2}$, and setting the resulting equation equal to Eq. (7.8) gives the cell-edge diffusion coefficients that preserve continuity of current at the cell interfaces,

$$\begin{aligned}\mathbb{D}_{n,l+1/2} &= 2\Delta_{l+1/2} \mathbb{D}_{n,l+1} (\Delta_l \mathbb{D}_{n,l+1} + \Delta_{l+1} \mathbb{D}_{n,l})^{-1} \mathbb{D}_{n,l} \, , \\ \mathbb{D}_{n,l-1/2} &= 2\Delta_{l-1/2} \mathbb{D}_{n,l} (\Delta_l \mathbb{D}_{n,l-1} + \Delta_{l-1} \mathbb{D}_{n,l})^{-1} \mathbb{D}_{n,l-1} \, .\end{aligned} \tag{7.11}$$

[1]Note that $\hat{\mathbf{n}} = n_i \mathbf{i} + n_j \mathbf{j} + n_k \mathbf{k}$ is the outward normal whereas the $_n$ subscript indicates the index of the moment equation, $n \in \{1, 2, 3, 4\}$.

Inserting Eq. (7.11) into Eq. (7.8) gives the internal cell-edge currents that can be used in in Eq. (7.7). Grouping unknowns yields

$$
\begin{aligned}
-\mathbb{C}^+_{n,i}\mathbb{U}_{n,i+1\,jk} - \mathbb{C}^-_{n,i}\mathbb{U}_{n,i-1\,jk} - \mathbb{C}^+_{n,j}\mathbb{U}_{n,i\,j+1\,k} - \mathbb{C}^-_{n,j}\mathbb{U}_{n,i\,j-1\,k} - \\
\mathbb{C}^+_{n,k}\mathbb{U}_{n,ij\,k+1} - \mathbb{C}^-_{n,k}\mathbb{U}_{n,ij\,k-1} + \sum_{m=1}^{4}\big[\mathbb{A}_{nm,ijk} + (\mathbb{C}^+_{m,i} + \mathbb{C}^-_{m,i} \\
+ \mathbb{C}^+_{m,j} + \mathbb{C}^-_{m,j} + \mathbb{C}^+_{m,k} + \mathbb{C}^-_{m,k})\delta_{nm}\big]\mathbb{U}_{m,ijk} = \mathbb{Q}_{n,ijk}\,, \\
n = 1,2,3,4\,, \quad (7.12)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbb{C}^+_{n,l} &= \frac{2}{\Delta_l}\mathbb{D}_{n,l+1}(\Delta_l\mathbb{D}_{n,l+1} + \Delta_{l+1}\mathbb{D}_{n,l})^{-1}\mathbb{D}_{n,l}\,, \\
\mathbb{C}^-_{n,l} &= \frac{2}{\Delta_l}\mathbb{D}_{n,l}(\Delta_l\mathbb{D}_{n,l-1} + \Delta_{l-1}\mathbb{D}_{n,l})^{-1}\mathbb{D}_{n,l-1}\,.
\end{aligned} \quad (7.13)
$$

The fission source is

$$
\mathbb{Q}_{n,ijk} = \frac{1}{k}\sum_{m=1}^{4}\mathbb{F}_{nm,ijk}\mathbb{U}_{m,ijk}\,. \quad (7.14)
$$

The Marshak boundary conditions given in Eq. (5.47) couple all of the moments at the problem boundaries. In finite-volume discretizations of standard diffusion operators, these unknowns can be algebraically eliminated from the system of equations. However, the moment coupling in SP_N prevents this simplification and imposes the introduction of cell-edge unknowns. The cell-edge currents that are inserted into Eq. (7.7) on low and high boundaries are

$$
\mathbb{J}_{n,1/2} = -\frac{2}{\Delta_1}\mathbb{D}_{n,1}(\mathbb{U}_{n,1} - \mathbb{U}_{n,1/2})\,, \quad (7.15)
$$

$$
\mathbb{J}_{n,L+1/2} = -\frac{2}{\Delta_L}\mathbb{D}_{n,L}(\mathbb{U}_{n,L+1/2} - \mathbb{U}_{n,L})\,. \quad (7.16)
$$

In order to close the system of equations for the added unknowns $\{\mathbb{U}_{n,1/2}, \mathbb{U}_{n,L}\}$, we use Eq. (5.47) with cell-edge currents defined by Eqs. (7.15) and (7.16) on the low and high boundaries, respectively. The resulting equations close the system

$$
\sum_{m=1}^{4}\Big(\mathbb{B}_{nm} + \frac{2}{\Delta_1}\mathbb{D}_{n,1}\delta_{nm}\Big)\mathbb{U}_{m,1/2} - \frac{2}{\Delta_1}\mathbb{D}_{n,1}\mathbb{U}_{n,1} = 0\,, \quad (7.17)
$$

$$
\sum_{m=1}^{4}\Big(\mathbb{B}_{nm} + \frac{2}{\Delta_L}\mathbb{D}_{n,L}\delta_{nm}\Big)\mathbb{U}_{m,L+1/2} - \frac{2}{\Delta_L}\mathbb{D}_{n,L}\mathbb{U}_{n,L} = 0\,. \quad (7.18)
$$

Likewise, reflecting boundary conditions are imposed with

$$
\mathbb{J}_{n,1/2} = 0\,, \quad (7.19)
$$

$$
\mathbb{J}_{n,L+1/2} = 0\,, \quad (7.20)
$$

which get used in Eq. (7.7) at reflecting boundaries. No additional unknowns are required at reflecting boundary faces.

To review, Eqs. (7.15) and (7.16) define the cell-edge currents that are inserted into the balance equation (7.7) on low and high boundary faces that have vacuum boundaries. Equations (7.17) and (7.18) provide the additional equations needed to close the system for the added unknowns on those faces. On reflecting faces no additional equations are required and the cell-edge net currents are zero.

Equation (7.12), with the appropriate boundary conditions defined by Eqs. (7.15)–(7.18), can be written in operator form as a generalized eigenvalue problem,

$$
\mathbf{A}\mathbf{u} = \frac{1}{k}\mathbf{B}\mathbf{u}\,. \quad (7.21)
$$

Here, we explicitly form the matrices $\mathbf{A}$ and $\mathbf{B}$. Forming the matrix once at the beginning of a solve reduces the runtime because the $(N_g \times N_g)$ matrix inverses required to calculate the $\mathbb{C}_n^{\pm}$ need only be performed once per solve. Furthermore, any branching logic needed at problem boundaries is codified in the matrix. Explicit formulation of the matrix system also enables the use of algebraic preconditioners. Finally, adjoint calculations are easily defined by simply taking the transpose of formulated matrices. These advantages are at least partially offset by the increased memory requirements associated with storing the matrix.

The choice to construct the SP_N equations as a monolithic system should be contrasted with approaches generally taken in the literature, which involve solving the SP_N equations using a Gauss–Seidel approach over the moment equations and possibly energy [**10**, **12**, **20**]. Indeed, it was observed in Ref. [**20**] through both Fourier analysis and numerical experiments that the monoenergetic form of the SP_N equations considered in this study (denoted the "composite" formulation by Zhang, Ragusa, and Morel) experience significant degradation in iterative convergence as the SP_N order is increased. Despite this fact, this method was observed to be the favored approach for problems with high scattering ratios and relatively low SP_N orders. Considering that Gauss–Seidel over energy is known to exhibit poor iterative performance for problems with significant upscattering [**21**, **22**], use of a Gauss–Seidel approach over *both* moments and energy is certain to result in slow convergence. Thus, avoiding the use of Gauss–Seidel iterations altogether in favor of solution approaches targeted at the entire matrix at once is an attractive option.

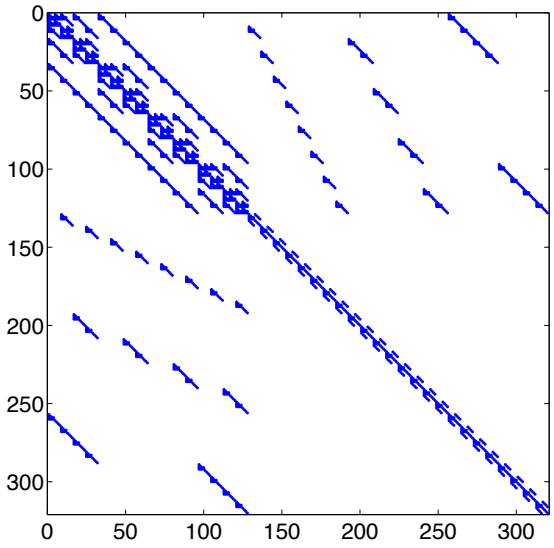
The multigroup SP_N equations have dimension $N_g \times N_m \times N_c$, where $N_m = (N+1)/2$ is the number of moment equations; N_c is the number of spatial cells; and, as mentioned previously, N_g is the number of energy groups. The solution vector $\mathbf{u}$ can be ordered in multiple ways; however, the ordering that minimizes the bandwidth of the matrix is to order $\mathbf{u}$ in groups-moments-cells,

$$\mathbf{u} = \begin{pmatrix} u_0 & u_1 & \dots & u_{m-1} & u_m & u_{m+1} & \dots & u_M \end{pmatrix}^T , \tag{7.22}$$

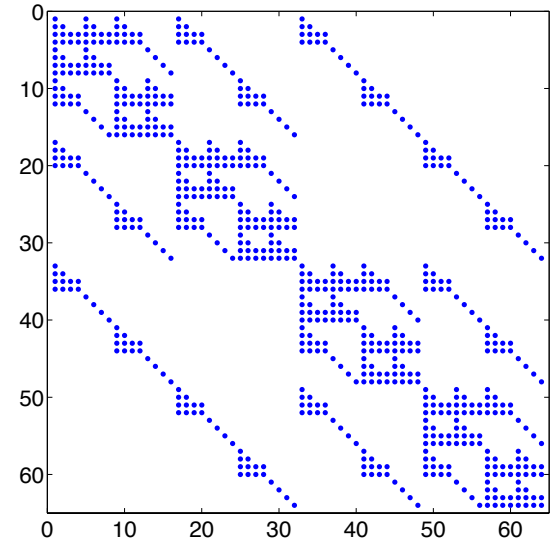
with

$$m = g + N_g(n + cN_m) , \tag{7.23}$$

where g is the group, n is the moment equation, and c is the cell. Consider an example SP_3 matrix that results from a $2 \times 2 \times 4$ grid with 4 groups. With all reflecting boundary conditions, the total number of unknowns is 128. Alternatively, vacuum boundary conditions must be coupled over all equations as indicated by Eq. (5.47); thus the size of the matrix will be augmented by $N_b \times N_g \times N_m$ unknowns, where N_b is the number of boundary cells over all faces. The sparsity plot for a $2 \times 2 \times 4$ grid with vacuum boundary conditions on four faces is shown in Fig. 7.2. One feature of the application of the SP_N equations is that, for many problems of interest, scattering moments above P_1 are not required to attain sufficient accuracy. Thus, all $\mathbf{\Sigma}_n$ matrices with $n > 1$ will simply be diagonal matrices. Also, there is relatively little coupling from low to high energy groups in most physical regimes, which yields $\mathbf{\Sigma}_n$ matrices that are predominately lower-triangular. A sparsity plot of a representative $\mathbf{\Sigma}_n$ matrix with 56 energy groups is shown in Fig. 7.3. It should be noted that while the monoenergetic SP_N equations are symmetric, the multigroup equations are nonsymmetric and therefore require the use of both eigensolvers and linear solvers intended for use with nonsymmetric systems.



(A) Full matrix.



(B) Close-up of the leading 64×64 block of the full matrix.

FIGURE 7.2. SP_3 matrix sparsity pattern for a $2 \times 2 \times 4$ spatial grid and 4 energy groups.

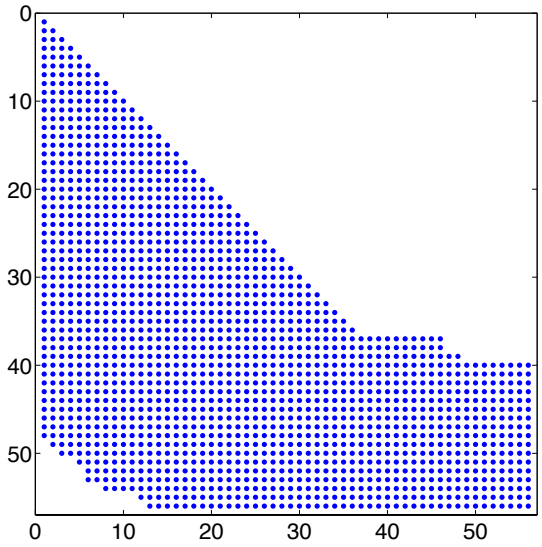


FIGURE 7.3. Sparsity of 56 group $\mathbf{\Sigma}_n$ matrix.

CHAPTER 8

Transport Solution Methods

1. Eigenvalue Solvers for the SP_N Equations

Although we are currently considering the solution of the SP_N form of the k-eigenvalue problem, the underlying principles are the same as with other transport discretizations. Therefore, it is expected that trends observed for other formulations (e.g., discrete ordinates or even diffusion) will still hold for SP_N problems. Several eigenvalue solvers have been proposed in the literature for the k-eigenvalue problem. Many approaches are based on converting the generalized eigenvalue problem of Eq. (7.21) to the equivalent standard eigenvalue problem

$$\mathbf{A}^{-1}\mathbf{B}\mathbf{u} = k\mathbf{u}\,. \tag{8.1}$$

Such a conversion is possible because the matrix $\mathbf{A}$ is guaranteed to be nonsingular [**23**]. In practice, factorizations of $\mathbf{A}$ will be too dense to store explicitly, but the matrix-vector product $\mathbf{y} = \mathbf{A}^{-1}\mathbf{B}\mathbf{v}$ can be computed by first performing the product $\mathbf{z} = \mathbf{B}\mathbf{v}$, followed by solving the linear system $\mathbf{A}\mathbf{y} = \mathbf{z}$. The simplest and perhaps most widely used approach is to use power iteration (PI) applied to the operator $\mathbf{A}^{-1}\mathbf{B}$. Convergence to the dominant eigenvalue is guaranteed [**24**], but rates of convergence (dictated by the dominance ratio $\rho \equiv \frac{k_2}{k_1}$, where k_1 and k_2 are the largest and second largest eigenvalues, respectively) may be prohibitively slow for many problems of interest. The rate of convergence of PI can be improved by first applying a shift to Eq. (7.21) before inverting, i.e.,

$$\left(\mathbf{A} - \mu\mathbf{B}\right)^{-1}\mathbf{B}\mathbf{u} = \bar{k}\mathbf{u}\,, \tag{8.2}$$

where μ is a fixed approximation to the inverse of the dominant eigenvalue [**25**]. It can easily be shown that the eigenvalue of the shifted system, $\bar{k}$, is related to the eigenvalue of the original system by

$$k = \frac{\bar{k}}{1 + \mu\bar{k}} \tag{8.3}$$

and that the dominance ratio is given by $\rho_{\text{shift}} = \left(\frac{k_2}{k_1}\right)\left(\frac{k_1 - 1/\mu}{k_2 - 1/\mu}\right)$, which can be significantly smaller than the dominance ratio of the original problem for $\mu \approx \frac{1}{k_1}$. Shifted power iteration faces two primary difficulties. First, improper selection of the shift will cause the method to converge to an incorrect eigenvalue. Second, solving linear systems involving the shifted matrix $(\mathbf{A} - \mu\mathbf{B})$ may be significantly more difficult than linear systems involving the unshifted matrix. This increased difficulty arises due to a clustering of the spectrum of the shifted matrix around 0 when μ is close to a true eigenvalue.

A natural extension of the shifted PI is to use the current eigenvalue estimate as a shift rather than using a fixed value. This results in the Rayleigh quotient iteration (RQI) algorithm:

$$\mathbf{u}^{(m+1)} = \left(\mathbf{A} - \lambda^{(m)}\mathbf{B}\right)^{-1}\mathbf{B}\mathbf{u}^{(m)} \tag{8.4}$$

$$\lambda^{(m+1)} = \frac{\langle \mathbf{u}^{(m+1)}, \mathbf{A}\mathbf{u}^{(m+1)} \rangle}{\langle \mathbf{u}^{(m+1)}, \mathbf{B}\mathbf{u}^{(m+1)} \rangle}\,, \tag{8.5}$$

where m is the iteration index and $\langle \cdot, \cdot \rangle$ indicates an inner product. The primary advantage of RQI is that convergence is quadratic and thus very few iterations will generally be required. The difficulties associated with shifted PI, however, are made even worse. Convergence to the dominant eigenvalue is not guaranteed, regardless of the choice of initial shift, and the linear systems that must be solved are not only more

difficult than the unshifted systems, but increase in difficulty as the algorithm progresses because $(\mathbf{A}-\lambda\mathbf{B})$ approaches a singular matrix as λ approaches the true eigenvalue. RQI has only recently begun to attract attention for radiation transport problems [**26**–**28**].

The methods discussed so far are all fixed-point methods, i.e., the next estimate of the solution depends only on the estimate immediately preceding it. An alternative to fixed-point iterations is subspace eigenvalue solvers in which information from several previous vectors is used to generate the next approximate solution. The vast majority of subspace solvers are built on two basic principles: extracting an approximate solution from a given subspace and adding an additional vector (or vectors) to the current subspace. In the solution extraction phase, an estimate of the desired eigenvector is obtained as a linear combination of the subspace basis vectors. The process is almost invariably achieved through a Rayleigh–Ritz procedure, solving the projected eigenvalue problem

$$\mathbf{V^T A V y} = \lambda \mathbf{V^T B V y}\,, \tag{8.6}$$

or for a standard eigenvalue problem [i.e., for solving Eq. (8.1)]

$$\mathbf{V^T A^{-1} B V y} = k\mathbf{y}\,, \tag{8.7}$$

where $\mathbf{V}$ contains a set of (typically orthogonal) basis vectors for the current subspace. For an appropriate selection of $\mathbf{V}$, the eigenvalues of the projected problem will closely approximate the eigenvalues of the original system, and the vectors $\mathbf{Vy}$ will approximate the corresponding eigenvectors. The approximate eigenvalues and eigenvectors obtained from the Rayleigh–Ritz procedure are generally referred to as Ritz values and Ritz vectors, respectively. In the case of symmetric matrices, it can be shown that the Ritz values and vectors satisfy certain optimality conditions. In the nonsymmetric case, no such optimality conditions apply, although the Rayleigh–Ritz procedure is still the basis for many eigensolvers [**29**].

The method of subspace expansion is what distinguishes the majority of subspace eigensolvers. In the Arnoldi method [**30**], the subspace is taken to be the Krylov subspace corresponding to the operator $\mathbf{A}^{-1}\mathbf{B}$. Thus, each iteration requires the operator action $\mathbf{y} = \mathbf{A}^{-1}\mathbf{Bx}$, just as with PI. Unlike PI, however, convergence of Arnoldi's method is not dictated by the dominance ratio, and so the number of iterations required to converge is likely to be significantly smaller. The Arnoldi method has gained attention in the transport literature in recent years and has become the staple of some production codes [**31**, **32**].

Another subspace eigensolver is the Davidson method [**33**]. The central idea behind subspace expansion in the Davidson method is that at iteration m, given an approximate eigenvalue, $\lambda^{(m)}$, and corresponding eigenvector, $\mathbf{u}^{(m)}$, one should seek a correction, $\mathbf{t}^{(m)}$, such that the eigenvalue correction equation given by

$$\mathbf{A}(\mathbf{u}^{(m)} + \mathbf{t}^{(m)}) = \lambda^{(m)}\mathbf{B}(\mathbf{u}^{(m)} + \mathbf{t}^{(m)}) \tag{8.8}$$

is satisfied. Rearranging this equation yields

$$(\mathbf{A} - \lambda^{(m)}\mathbf{B})\mathbf{t}^{(m)} = -(\mathbf{A} - \lambda^{(m)}\mathbf{B})\mathbf{u}^{(m)} \equiv -\mathbf{r}^{(m)}\,, \tag{8.9}$$

where $\mathbf{r}^{(m)}$ is the residual of the eigenvalue problem. This equation implies that a linear system involving the matrix $(\mathbf{A} - \lambda^{(m)}\mathbf{B})$ must be solved at each iteration. This is likely to incur a significant computational expense, suggesting the use of a preconditioner, $\mathbf{M}$, that approximates $(\mathbf{A} - \lambda^{(m)}\mathbf{B})$, leading to the Davidson correction equation

$$\mathbf{M}\mathbf{t}^{(m)} = -\mathbf{r}^{(m)}\,. \tag{8.10}$$

Although the original Davidson method was targeted at the symmetric standard eigenvalue problem, later work extended the idea to the generalized eigenvalue problem [**34**] and to nonsymmetric matrices [**35**]. The Davidson method has one extremely appealing feature for the k-eigenvalue problem: because it solves the generalized eigenvalue problem directly, it is not necessary to solve any linear system involving the full problem operator; only a preconditioner approximating the solution of a linear system is required. Despite this attractive feature, use of the Davidson method for transport problems has only very recently appeared in the literature [**27**, **36**, **37**].

Although it will not be considered further in this study, another subspace solver that has garnered much attention in the mathematics community in recent years is the Jacobi-Davidson method [**38**]. The

TABLE 8.1. Classification of eigensolvers by linear system solution required

	Solver Type	
Linear System Matrix	Fixed Point	Subspace
$\mathbf{M}$	–	Davidson
$\mathbf{A}$	Power Iteration	Arnoldi
$\mathbf{A}-\mu\mathbf{B}$	Shifted Power Iteration	Shift-and-Invert Arnoldi
$\mathbf{A}-\lambda^{(m)}\mathbf{B}$	Rayleigh Quotient It.	Jacobi-Davidson*

*The Jacobi-Davidson correction equation includes additional projection operators.

Jacobi–Davidson correction equation is given by

$$\left(\mathbf{I}-\mathbf{u}\mathbf{u}^T\right)\left(\mathbf{A}-\lambda^{(m)}\mathbf{B}\right)\left(\mathbf{I}-\mathbf{u}\mathbf{u}^T\right)\mathbf{t}^{(m)}=-\mathbf{r}^{(m)}\,, \tag{8.11}$$

where the projection operator $\left(\mathbf{I}-\mathbf{u}\mathbf{u}^T\right)$ forces the update to be orthogonal to the current solution estimate and prevents stagnation of the method.

A classification of the eigenvalue solvers discussed here is given in Table 8.1 based on the linear system that each method is required to solve. This list is not intended to be exhaustive, as several other subspace eigenvalue solvers appear in the mathematics literature. Additionally, eigensolvers based on optimization strategies or general nonlinear solvers exist (the latter case was studied in Refs. [**39**, **40**] for the k-eigenvalue problem).

The idea of preconditioning for eigensolvers involving a conversion to a standard eigenvalue problem is straightforward: a preconditioner is applied to accelerate the convergence of the solution of the relevant linear system of equations only and therefore does not directly influence the rate of convergence of the eigensolver (though clearly it influences the efficiency of the overall approach). Preconditioning for eigenvalue solvers operating directly on a generalized eigenvalue problem (e.g., Davidson-style methods) is not understood as well. Preconditioning in these cases has a direct impact on the rate of convergence of the eigensolver. Davidson's original method focused solely on the use of diagonal preconditioning (a natural choice because the matrices under consideration were strongly diagonally dominant), though subsequent studies considered more general preconditioners [**41**, **42**]. Generally speaking, the preconditioner, $\mathbf{M}$, should approximate the matrix $(\mathbf{A}-\lambda^{(m)}\mathbf{B})$. Care must be taken, however, because stagnation can occur if $\mathbf{M}$ too closely approximates $(\mathbf{A}-\lambda^{(m)}\mathbf{B})$. The reason for this can easily be seen from Eq. (8.9), which admits $\mathbf{t}^{(m)}=\mathbf{u}^{(m)}$ as a solution. Thus, the vector proposed for addition to the current subspace is already contained in the subspace, and no further progress toward a solution can be made. One possible remedy for this stagnation, due to Olsen [**43**], is to force the subspace expansion to be orthogonal to the current iterate. The downside to this approach is that two applications of the preconditioner are required at each iteration, rather than only a single application in the standard Davidson method.

One preconditioning approach that obviates the stagnation issue is to have $\mathbf{M}$ approximate $(\mathbf{A}-\mu\mathbf{B})$ for a fixed value of μ. In fact, if the smallest magnitude eigenvalue is being sought, it may be sufficient to have $\mathbf{M}$ approximate $\mathbf{A}$. In the limiting case where $\mathbf{M}=\mathbf{A}$, the subspace constructed by the Davidson method is the Krylov subspace corresponding to $\mathbf{A}^{-1}\mathbf{B}$ and is thus equivalent (in exact arithmetic) to the Arnoldi method. Using the Arnoldi method in this case, however, requires $\mathbf{A}^{-1}$ to be applied to high accuracy to maintain the structure of the subspace. For problems where direct factorization of $\mathbf{A}$ is not practical, this can impose a significant computational burden. The Davidson method, on the other hand, explicitly computes the residual at each iteration and thus is not constrained by the accuracy to which the linear solves are performed. In this respect, the Davidson method with $\mathbf{M}\approx\mathbf{A}$ can be viewed as an inexact Arnoldi approach. There is a cost associated with the freedom to perform inexact solves: storage of additional basis vectors and additional orthogonalization work must be done because the subspace has no structure for the subspace for the Davidson solver to exploit.

Part 2

Stochastic Methods

CHAPTER 9

The Monte Carlo Method

The distinguishing feature of Monte Carlo methods versus deterministic transport techniques is the ability to treat stochastic events continuously in many variables including energy, space, and angle. This allows the complete history of the particle track to be simulated. Many such calculations will statistically converge to a meaningful result allowing the user to determine several quantities which cannot be calculated directly from deterministic approaches [**2**].

The essence of Monte Carlo consists of simulating a finite number of particle histories using a pseudo random number generator. Random numbers are used to sample probability distributions for particle events such as scattering angles, energy transfer, collisions and so on. If the cross sections describing such events are known to a high degree of accuracy, the transport can be simulated exactly for an infinite number of histories. The desired result is determined by calculating the expectation value of some quantity, $\hat{x}$, in the region of interest. This quantity could be flux, dose, current, or any number of variables. The estimate of x then has the form

$$\hat{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \,, \tag{9.1}$$

where x_n is the contribution of each history to x [**2**].

Because Monte Carlo methods rely on the result of stochastic events to continue the history, most practical three dimensional geometries may be modeled easily. The main drawback of Monte Carlo with regard to deterministic approaches is the calculation of quantities in highly attenuating material. Because Monte Carlo tallies are dependent on events occurring within the region of interest, regions in which few events occur yield poor statistical results unless a large number of particle histories are run. As an example, consider a photon shielding problem in which the area of interest is located behind a thick lead shield. Because of the high attenuating properties of lead very few particles will penetrate the shield and reach the area of interest. A statistically poor answer will result because of the few events which are tallied.

In this chapter the general Monte Carlo techniques which have been utilized in Shift are described. A comprehensive treatment of general Monte Carlo techniques for many particles and conditions may be found in the literature [**2**, **44**–**47**].

1. Probability Distribution Functions

Events along a particle track are randomly sampled according to the probability of such an event occurring. Consider a random variable X that has a value which is defined by some probability distribution. For discrete distributions the probability mass function (PMF) is defined as the probability that the expectation value of the random variable X is x in a single outcome. The PMF has the following properties,

$$0 \leq p_X(x) \leq 1 \,, \tag{9.2}$$

and

$$\sum_x p_X(x) = 1 \,, \tag{9.3}$$

for all values of x [**46**].

The cumulative probability distribution (CPD) is defined as the probability that X will have a value less then x. Accordingly, the CPD is given by

$$\begin{aligned} P\{X \leq x\} &= F_X(x) \\ &= \sum_{x' \leq x} p_X(x') \,, \end{aligned} \tag{9.4}$$

where $0 \leq F_X \leq 1$. Alternatively, the CPD may be written

$$P\{a \leq X \leq b\} = F_X(b) - F_X(a) \,, \tag{9.5}$$

where P is the probability that X will have a value between a and b and $F_X(b)$ and $F_X(a)$ are defined by Eq. (9.4).

Now consider the case where the random variable X is defined by a continuous distribution. The probability density function (PDF) is

$$f(x)dx = P\{x \leq X \leq x + dx\} \,, \tag{9.6}$$

where $f(x)$ is the PDF in the limit $dx \to 0$. Analogous to the discrete case, the PDF has the following properties,

$$f(x) \geq 0 \text{ for all } x \,, \tag{9.7}$$

and

$$\int_{x^-}^{x^+} f(x)dx = 1 \,, \tag{9.8}$$

where the values of x belong to the set $[x^-, x^+]$ [**2**].

The CPD, $F(x)$, has been defined in Eq. (9.4). For a continuous distribution $F(x)$ is

$$F(x) = \int_{-\infty}^{x} f(x')dx' \,, \tag{9.9}$$

or inversely,

$$\frac{dF(x)}{dx} = f(x) \,. \tag{9.10}$$

Closer analysis of Eq. (9.9) reveals the following limits associated with $F(x)$ [**2**],

$$\lim_{x \to \infty} F(x) \equiv F(\infty) = 1 \,, \tag{9.11}$$

$$\lim_{x \to -\infty} F(x) \equiv F(-\infty) = 0 \,. \tag{9.12}$$

The probability defined over the range $[a, b]$ as given by Eq. (9.5) holds for both continuous and discrete distributions.

The discussion of probability distribution functions has so far been limited to functions involving one variable. The definitions given by Eqs. (9.4), (9.6), and (9.9) may easily be extended to multiple variables. The joint PDF for two random variables is defined,

$$f(x, y)dxdy = P\{x \leq x' \leq x + dx, y \leq y' \leq +dy\} \,. \tag{9.13}$$

The joint PDF obeys the following normalization,

$$\int_{x^-}^{x^+} dx \int_{y^-}^{y^+} dy f(x, y) = 1 \,. \tag{9.14}$$

The marginal PDF in x or y may be found by integrating the joint PDF over the variable of interest,

$$g(x) = \int_{y^-}^{y^+} dy f(x, y) \,, \tag{9.15}$$

where the joint PDF in y, $g(y)$, is defined by integrating over dx. As in the single variable case, the joint PDF has an associated joint CPD,

$$P\{x' \leq x, y' \leq y\} = F(x, y)\,. \tag{9.16}$$

In analogy to Eq. (9.9) the joint CPD may be written [**2**],

$$F(x, y) = \int_{-\infty}^{x} dx' \int_{-\infty}^{y} dy' f(x', y')\,. \tag{9.17}$$

The probability that x will have a value between x and $x + dx$ for a given value of y is the conditional PDF and is given by

$$f(x \mid y) = \frac{f(x, y)}{g(y)}\,, \tag{9.18}$$

where $g(y)$ is the marginal PDF in y as given by Eq. (9.15). The conditional PDF $f(y \mid x)$ is defined similarly [**2**],

$$f(y \mid x) = \frac{f(x, y)}{g(y)}\,. \tag{9.19}$$

The conditional PDF may also be defined for a separate condition, A, where the probability of A is given by

$$P[A] = \int \int_{A} f(x, y) dx dy\,. \tag{9.20}$$

Given $P[A]$, the probability that x and y will have a value between x and $x + dx$ and y and $y + dy$ is given by [**46**],

$$f(x, y \mid A) = \frac{f(x, y)}{P[A]}\,. \tag{9.21}$$

Sampling a function y where $y = y(x)$ and x is a random variable requires the PDF for y as a function of x. Accordingly, we define $g(y)dy$ as the probability that y is between y and $y + dy$ and $f(x)dx$ as the probability that x has a value between x and $x + dx$. The objective is to find the PDF for y so that y may be sampled as a random variable. Given the preceding conditions on x and y, the PDFs $f(x)$ and $g(y)$ must satisfy

$$|g(y)dy| = |f(x)dx|, \tag{9.22}$$

which leads immediately to [**2**],

$$g(y) = f(x)\frac{dx}{dy}\,. \tag{9.23}$$

For a given PDF, $f(x)$, and associated random variable, x, the PDF of a function, $y = y(x)$, may be found using the transformation given by (9.23).

2. Distribution Sampling

Having defined a random variable and its associated PDF and CPD we proceed to the relevance of these functions to the Monte Carlo process. Random number generators provide sequences of numbers uniformly distributed between 0 and 1. Thus, random numbers may be used to sample uniformly distributed functions in an unbiased manner. Because the CPD is uniformly distributed between 0 and 1 we have

$$F(x) = \xi\,, \tag{9.24}$$

where ξ is a random number. The objective is to sample the distribution of x so the required operation becomes,

$$x = F(\xi)^{-1}\,. \tag{9.25}$$

The inversion of CPDs representing physical phenomenon is a primary process in Monte Carlo transport. There are numerous techniques for sampling CPDs which are described in detail in References **2**, **44**, **46**, **48** and **47**. A brief overview of techniques pertinent to Exnihilo follows.

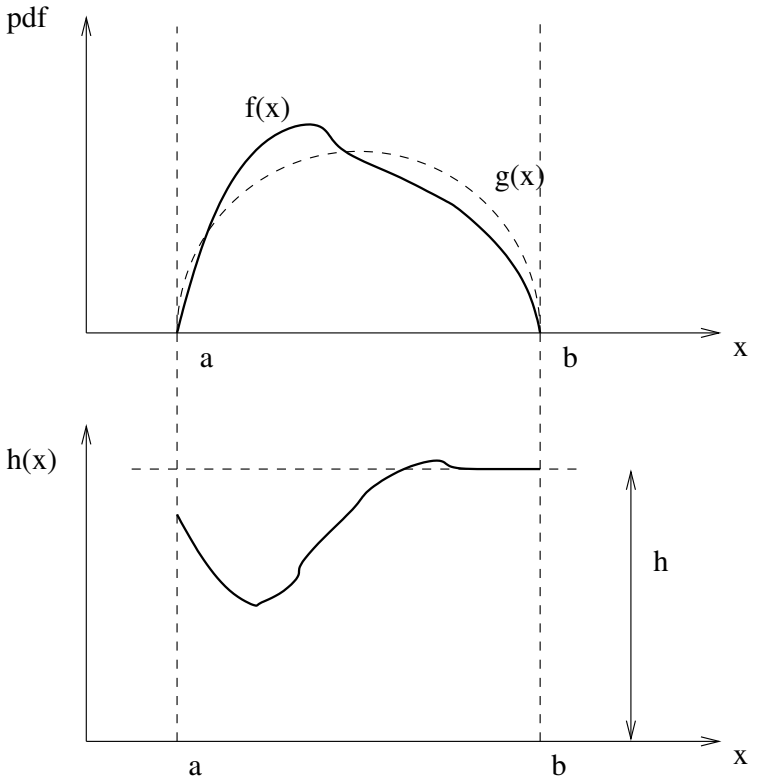


FIGURE 9.1. The probability density function $f(x)$ along with a more simple PDF, $g(x)$. The ratio of $g(x)$ and $f(x)$, $h(x)$ is plotted in the lower graph. The upper limit of $h(x)$ is h_{max}.

2.1. Cumulative Distribution Method. If the PDF is defined by a simple expression then the CPD may be found through direct integration using Eq. (9.9). As an example consider a particle born in an infinite homogeneous medium characterized by the cross section σ. The PDF for a particle interacting in a thin slab of width dx is

$$f(x)dx = \sigma\, e^{-\sigma x} dx\,, \tag{9.26}$$

which is the product of the probability of the particle traveling a distance x and the probability of an interaction occurring within some distance dx [**2**]. Utilization of Eq. (9.9) gives

$$F(x) = \int_0^x \sigma\, e^{-\sigma x} dx\,, \tag{9.27}$$

which, after integration yields,

$$F(x) = 1 - e^{-\sigma x}\,. \tag{9.28}$$

Substituting ξ for $F(x)$, the expression for the random variable is

$$x = -\frac{\ln(1-\xi)}{\sigma} = \frac{\ln(\xi)}{\sigma}\,, \tag{9.29}$$

where $1 - \xi \equiv \xi$ because both distributions are uniformly distributed between 0 and 1. Utilization of Eq. (9.29) allows for the unbiased sampling of x.

2.2. Rejection Technique. The cumulative distribution method is impractical in many cases because the PDF cannot be integrated easily. An alternative method is the *rejection technique* [**45**]. Consider an analytically complex function $f(x)$ as shown in Figure 9.1. A simple PDF, $g(x)$, is defined over the same limits. This function has a trivial CPD, $G(x)$, which may be easily inverted to give $G^{-1}(\xi)$. The desired CPD is defined

$$F(x) = \int_a^x h(x')g(x')dx' \,, \tag{9.30}$$

where

$$h(x) \equiv \frac{f(x)}{g(x)} \,. \tag{9.31}$$

If $g(x) \sim f(x)$ then $h(x)$ will be near unity. The function $g(x)$ is constrained by some maximum bounding value, h_{max}, such that $h(x) \leq h_{max}$ in $[a, b]$. In practice h_{max} is the maximum value of $h(x)$ in the interval $[a, b]$.

The rejection technique is applied as follows [**45**]:

1.: Select a random number ξ_i.
2.: Obtain an initial value of x using $x_i = G^{-1}(\xi_i)$.
3.: Select a second random number ξ_{i+1}.
4a.: If $\xi_{i+1} > h(x_i)/h_{max}$ then reject the value x_i and return to step 1.
4b.: If $\xi_{i+1} < h(x_i)/h_{max}$ then x_i is accepted as the appropriate value of x.

3. Monte Carlo Errors

The expectation value for x is defined

$$E[g(x)] = \int g(x)f(x)dx \,, \tag{9.32}$$

where $f(x)$ is the PDF for x. For discrete probabilities,

$$E[g(x)] = \sum_x g(x)f(x) \,. \tag{9.33}$$

For a given sample, x_n, the expectation value is

$$E[x] = E[x_n] \,. \tag{9.34}$$

And, the *true mean* is equal to the expectation value of x, $\bar{x} = E[x]$. Now, we need to show that the expectation value of the sample mean ($\hat{x}$) is equal to $\bar{x}$. The sample mean is (from Eq. (9.1))

$$\hat{x} = \frac{1}{N}\sum x_n \,. \tag{9.1}$$

Then,

$$\begin{aligned} E[\hat{x}] &= E\big[\frac{1}{N}\sum x_n\big] \\ &= \frac{1}{N}\sum E[x_n] \\ &= \frac{1}{N}NE[x] \\ &= E[x] = \bar{x} \,. \end{aligned} \tag{9.35}$$

3.1. Variance and Error. Likewise, we can define the variance as the expected value of the function defining the second moment about the mean,

$$\sigma^2(x) \equiv E[(x-\bar{x})^2]\,, \tag{9.36}$$

Then, by virtue of the above definitions

$$\begin{aligned} E[(x-\bar{x})^2] &= \int (x-\bar{x})^2 f(x)dx \\ &= \int (x^2 - 2\bar{x}x + \bar{x}^2) f(x)dx \\ &= E[x^2] - 2\bar{x}E[x] + \bar{x}^2 \\ &= E[x^2] - E[x]^2 \\ &= \langle x^2\rangle - \langle x\rangle^2\,, \end{aligned} \tag{9.37}$$

where

$$E[x^n] = \langle x^n\rangle = \int x^n f(x)dx\,. \tag{9.38}$$

Obviously, $\langle x\rangle = \bar{x}$.

Now, we need to define the variance of the sample mean, $\hat{x}$. The variance of the sample mean is [**2**]

$$\sigma^2(\hat{x}) = E[(\hat{x}-\bar{x})^2]\,. \tag{9.39}$$

Substituting for $\bar{x}$ and expanding the square term gives

$$\begin{aligned} \sigma^2(\hat{x}) &= E\Big[\big(\frac{1}{N}\sum x_n - \bar{x}\big)^2\Big] \\ &= E\Big[\big(\frac{1}{N}\sum x_n - \frac{1}{N}\sum \bar{x}\big)^2\Big] \\ &= E\Big[\frac{1}{N^2}\big(\sum (x_n-\bar{x})\big)^2\Big]\,. \end{aligned} \tag{9.40}$$

Expanding the square term gives

$$\begin{aligned} \big(\sum (x_n-\bar{x})\big)^2 &= \big(\sum (x_n-\bar{x})\big)\big(\sum (x_n-\bar{x})\big) \\ &= \big((x_0-\bar{x}) + (x_1-\bar{x}) + (x_2-\bar{x}) + \ldots\big)\big((x_0-\bar{x}) + (x_1-\bar{x}) + (x_2-\bar{x}) + \ldots\big) \\ &= \big[(x_0-\bar{x})(x_0-\bar{x}) + (x_1-\bar{x})(x_1-\bar{x}) + (x_2-\bar{x})(x_2-\bar{x}) + \ldots\big] \\ &\quad + \big[(x_0-\bar{x})(x_1-\bar{x}) + (x_0-\bar{x})(x_2-\bar{x}) + (x_1-\bar{x})(x_0-\bar{x}) + \ldots\big] \\ &= \sum_n (x_n-\bar{x})^2 + \sum_n \sum_m (x_n-\bar{x})(x_m-\bar{x})\delta_{nm}\,. \end{aligned} \tag{9.41}$$

Now, plugging back into the variance

$$\begin{aligned} \sigma^2(\hat{x}) &= E\Big[\frac{1}{N^2}\big(\sum_n (x_n-\bar{x})^2 + \sum_n \sum_m (x_n-\bar{x})(x_m-\bar{x})\delta_{nm}\big)\Big] \\ &= \frac{1}{N^2}\sum E\big[(x_n-\bar{x})^2\big] + \frac{1}{N^2}\sum_n \sum_m E\big[(x_n-\bar{x})(x_m-\bar{x})\big]\delta_{nm}\,. \end{aligned} \tag{9.42}$$

First, we evaluate the second term. Expanding gives

$$E\big[(x_n-\bar{x})(x_m-\bar{x})\big] = E[x_m x_n] - \bar{x}E[x_n] - \bar{x}E[x_m] + \bar{x}^2\,. \tag{9.43}$$

Since x_m and x_n are independently sampled from the same PDF ($f(x)$) we have

$$E[x_m] = E[x_n] = \bar{x}\,. \tag{9.44}$$

Also,

$$E[x_m x_n] = \int x_m x_n f(x_m, x_n) dx_m dx_n\,. \tag{9.45}$$

Because the samples are independent we have $f(x_m, x_n) = f(x_m)f(x_n)$, and

$$E[x_m x_n] = \Big(\int x_m f(x_m) dx_m\Big)\Big(\int x_n f(x_n) dx_n\Big) = E[x_m]E[x_n] = \bar{x}^2 \,. \tag{9.46}$$

Substituting these expressions gives

$$\bar{x}^2 - \bar{x}^2 - \bar{x}^2 + \bar{x}^2 = 0 \,. \tag{9.47}$$

Then,

$$\begin{aligned} \sigma^2(\hat{x}) &= \frac{1}{N^2} \sum E\big[(x_n - \bar{x})^2\big] \\ &= \frac{N}{N^2} E\big[(x_n - \bar{x})^2\big] \\ &= \frac{1}{N}\big(E[x_n^2] - 2\bar{x}E[x_n] + \bar{x}^2\big) \,. \end{aligned} \tag{9.48}$$

As stated above,

$$E[x_n^2] = E[x^2] = \langle x^2 \rangle \,, \tag{9.49}$$

$$E[x_n] = E[x] = \langle x \rangle \,. \tag{9.50}$$

Finally, the variance of the sample mean is

$$\sigma^2(\hat{x}) = \frac{\langle x^2 \rangle - \langle x \rangle^2}{N} = \frac{\sigma^2(x)}{N} \,. \tag{9.51}$$

The *error* of the sample mean is thus inversely proportional to the number of samples squared,

$$\sigma(\hat{x}) = \frac{\sigma(x)}{\sqrt{N}} \,, \tag{9.52}$$

where $\sigma(x)$ is the error from the mean for each individual sample.

3.2. Sample Variance. Having defined the true variance (error) of the mean and sample mean, we must define the *sample variance* (S^2) that is an unbiased estimator of the true variance,

$$E[S^2] = \sigma^2(x) \,. \tag{9.53}$$

Defining the sample variance as follows,

$$S^2 = C \sum_n (x_n - \hat{x})^2 \,, \tag{9.54}$$

where C is a normalization that ensures that the sample variance is an unexpected estimator of the true variance. We take the expected value of the sample variance as

$$E[S^2] = CE\big[\textstyle\sum (x_n - \hat{x})^2\big] \,. \tag{9.55}$$

The operand can be decomposed as follows,

$$(x_n - \hat{x}) = (x_n - \bar{x}) - (\hat{x} - \bar{x}) \,, \tag{9.56}$$

yielding

$$\begin{aligned} E[S^2] &= CE\Big[\sum \big((x_n - \bar{x}) - (\hat{x} - \bar{x})\big)^2\Big] \\ &= CE\Big[\sum \big((x_n - \bar{x})^2 - 2(x_n - \bar{x})(\hat{x} - \bar{x}) + (\hat{x} - \bar{x})^2\big)^2\Big] \\ &= CE\Big[\sum (x_n - \bar{x})^2 - \sum 2(x_n - \bar{x})(\hat{x} - \bar{x}) + N(\hat{x} - \bar{x})^2\Big] \,. \end{aligned} \tag{9.57}$$

First, we expand the middle term of the operand and substitute for $\bar{x}$,

$$\begin{aligned}\sum 2(x_n-\bar{x})(\hat{x}-\bar{x}) &= 2\sum_n\Big((x_n-\bar{x})(\frac{1}{N}\sum_m x_m-\bar{x})\Big)\\ &= 2\sum_n\Big(\frac{1}{N}x_n\sum_m x_m - x_n\bar{x} - \frac{1}{N}\bar{x}\sum_m x_m + \bar{x}^2\Big)\\ &= 2\Big(\frac{1}{N}\sum_n x_n\sum_m x_m - \bar{x}\sum_n x_n - \bar{x}\sum_m x_m + N\bar{x}^2\Big)\\ &= 2\Big(N\hat{x}^2 - N\bar{x}\hat{x} - N\bar{x}\hat{x} + N\bar{x}^2\Big)\\ &= 2N(\hat{x}-\bar{x})^2\,.\end{aligned} \tag{9.58}$$

Plugging this into the expectation value of the sample variance gives

$$E[S^2] = CE\big[\sum(x_n-\bar{x})^2 - N(\hat{x}-\bar{x})^2\big]\,. \tag{9.59}$$

Above we showed that

$$E\big[(\hat{x}-\bar{x})^2\big] = \frac{\sigma^2(x)}{N}\,. \tag{9.60}$$

Then,

$$E[S^2] = C\Big(E\big[\sum(x_n-\bar{x})^2\big] - \sigma^2(x)\Big)\,. \tag{9.61}$$

Expanding gives

$$\begin{aligned}E[S^2] &= C\Big(\sum\big(E[x_n^2] - 2\bar{x}E[x_n] - \bar{x}^2\big) - \sigma^2(x)\Big)\\ &= C\Big(N\big(\langle x^2\rangle - \langle x\rangle\big) - \sigma^2(x)\Big)\\ &= C(N\sigma^2(x) - \sigma^2(x))\\ &= C(N-1)\sigma^2(x)\,.\end{aligned} \tag{9.62}$$

In order to satisfy the condition $E[S^2]=\sigma^2(x)$ the following must be true

$$C(N-1) = 1 \tag{9.63}$$

which implies

$$C = \frac{1}{N-1}\,. \tag{9.64}$$

Finally, we have the sample variance defined

$$S^2 = \frac{1}{N-1}\sum_n(x_n-\hat{x})^2\,. \tag{9.65}$$

A more convenient form for calculating S^2 in practice is to expand the operand,

$$\begin{aligned}S^2 &= \frac{1}{N-1}\sum_n(x_n^2 - 2\hat{x}x_n - \hat{x}^2)\\ &= \frac{1}{N-1}(N\widehat{x^2} - N\hat{x}^2)\\ &= \frac{N}{N-1}(\widehat{x^2} - \hat{x}^2)\,,\end{aligned} \tag{9.66}$$

where $\hat{x}$ is given by Eq. (9.1) and

$$\widehat{x^2} = \frac{1}{N}\sum_{n=1}^{N}x_n^2\,. \tag{9.67}$$

Having showed that S^2 is an unbiased estimate of the true variance, we can use Eq. (9.52) to calculate the error of the sample mean

$$\sigma(\bar{x}) = \frac{\sigma(x)}{\sqrt{N}} \to \frac{S}{\sqrt{N}} \, . \tag{9.68}$$

CHAPTER 10

Parallel Monte Carlo Methods

Monte Carlo methods require various parallel algorithms to communicate the information a particle carries with it from one domain to another domain (processor).

1. Multiple Set Overlapping Domain Decomposition Algorithm

The Multiple Set Overlapping Domain (MSOD) algorithm is a method that divides a given problem geometry into structured Cartesian "blocks" based upon x, y, and z grids [**49**]. An overlap fraction specifies how much of each block is duplicated between adjacent neighbors. Depending on the type of transport problem, this overlap fraction can reduce the amount of communication needed between processors during the transport solve. The MSOD method replicates this block-decomposed geometry into "sets" based upon the number of processors.

Mathematically this means the geometry is decomposed into N_s sets, each of which contains N_b blocks. Using this nomenclature, the number of domains (processes) in a problem is thus

$$N_\mathrm{d} = N_\mathrm{s} \times N_\mathrm{b} \, . \tag{10.1}$$

The number of particles per set is then determined by

$$N_\mathrm{p,s} = \frac{N_\mathrm{p}}{N_\mathrm{s}} \, , \tag{10.2}$$

where N_p is the number of requested particle histories to run in the Monte Carlo simulation. An example of an MSOD problem is illustrated in Fig. 10.1. MSOD naturally reduces to full domain replication or full domain decomposition by defining $N_\mathrm{b} = 1$ or $N_\mathrm{b} = N_\mathrm{d}$, respectively.

What the heck is a boundary mesh?

There are two principal parallel communication channels in the MSOD algorithm: *instra-set* (block-to-block communication within a set) and *intra-block* (set-to-set communication on the same block). There is no communication of particles across sets; however, when $N_\mathrm{b} > 1$, particle histories need to be communicated across blocks. This block-to-block communication is handled by a modified domain-decomposition algorithm which is detailed in section 3.

Several parameters tallied during a Monte Carlo simulation require communication. These include tallies and fission sites. The fission site rebalance methodology is discussed in section 2. Intra-block communication is required to accumulate tally site contributions in overlapping region to the "parent" block. If overlapping domains are not present, tallies may still require intra-block communication if nonaligned tally meshes are present. When performing a k-eigenvalue Monte Carlo calculation, the k-eigenvalue estimators are global and therefore require a full global reduction across all sets and blocks to determine contributions to this estimator.

For mesh tallies in a Monte Carlo simulation, a two-step processed is used to handle communication:

(1) relay off-block tally contributions to parent blocks, and
(2) perform an intra-set reduction on each block.

To best understand this procedure, an example of this process is illustrated in Fig. 10.2. In this example, the decomposition per set is such that each tally mesh cell is owned by a block. A source mesh is defined on each block that includes overlapped tally mesh cells. These overlapped cells can be the result of either a defined overlap region or noncoincident boundary and tally meshes. The tallies defined on each block are scored on the source mesh. At the end of the particle transport calculation a global scatter is performed

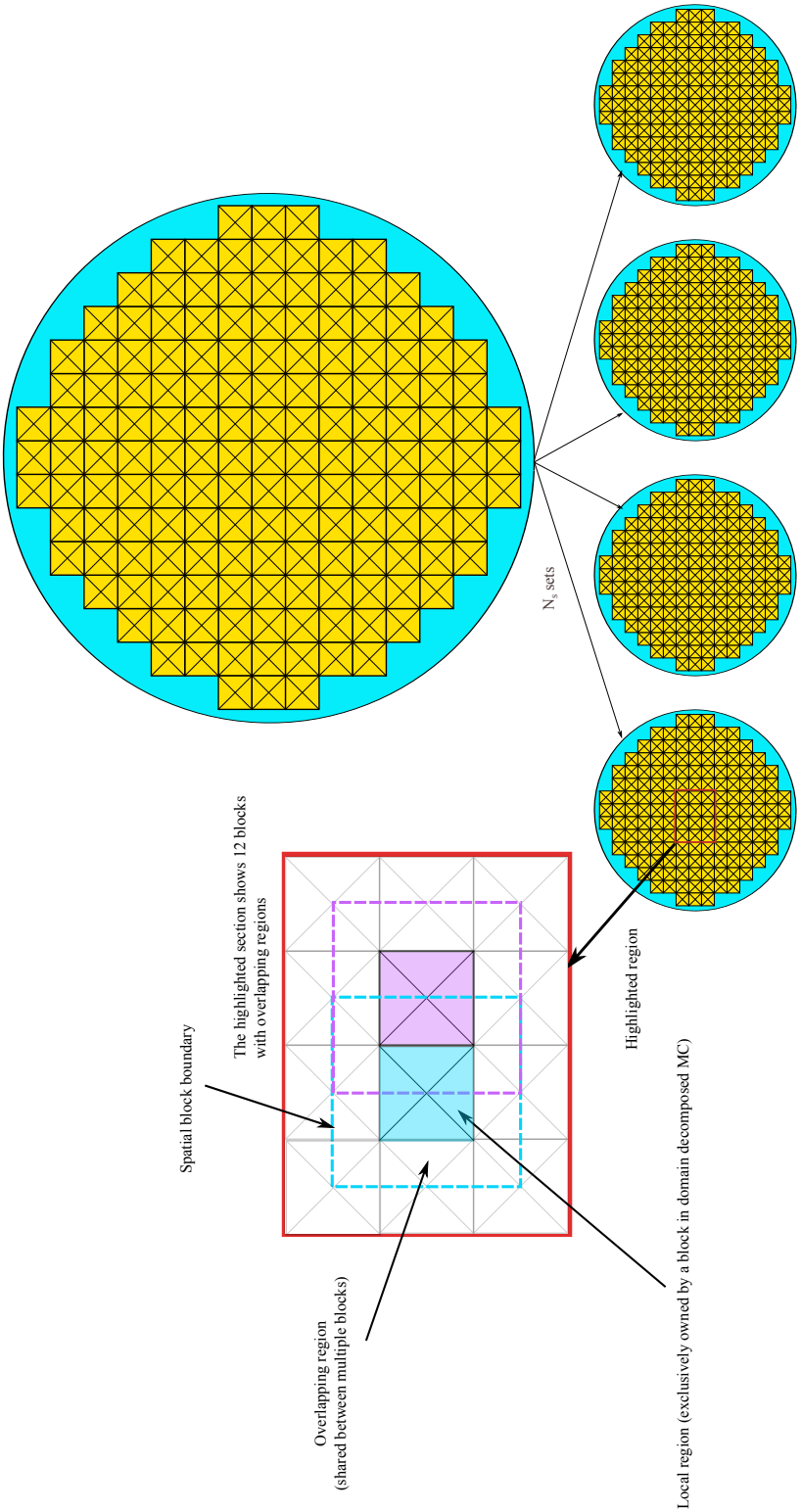


FIGURE 10.1. Example of MSOD geometry. In this example, $N_s = 4$ and the entire geometry is represented within each set.

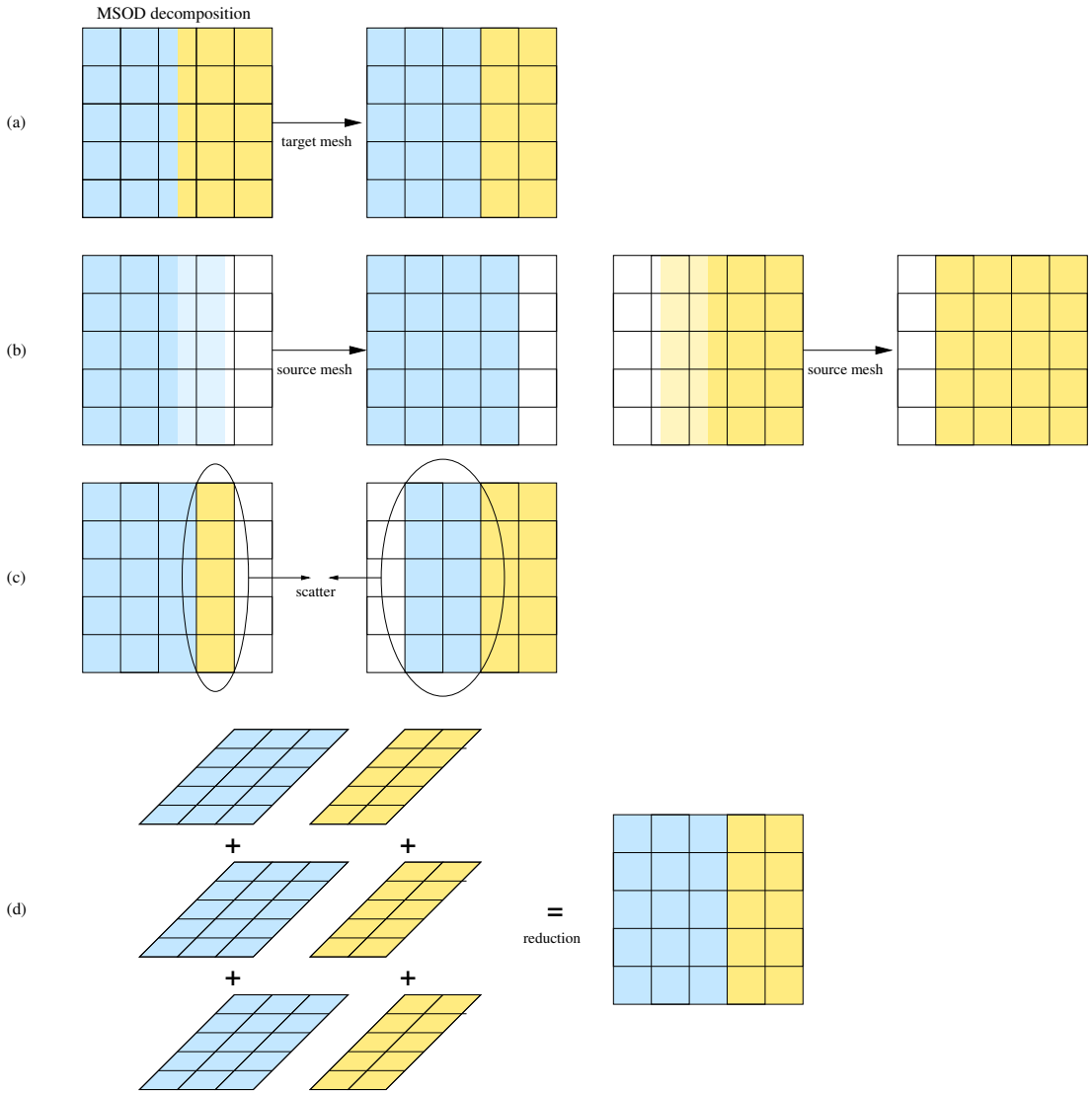


FIGURE 10.2. Illustration of a mesh tally reduction with a 5×5 tally mesh and a 2×1 boundary mesh yielding 2 blocks per set. In (a) a target mesh is defined for each block. In (b) a source mesh is defined for each block. The boundary mesh overlap regions are indicated by the lightly shaded areas. In (c) a global scatter is performed from the source mesh to the target mesh. In (d) an intra-set reduction is performed to get a final, global tally on each block.

in which tally results are accumulated on each block's target mesh which only requires nearest-neighbor communication. Finally, an intra-set global reduction, dimensioned by the number of sets, accumulates the global tally results on each block.

2. Fission Site Rebalance

This section gives the details of the fission source rebalance method including the communication pattern. Fission source sites are the sites sampled during a k-eigenvalue Monte Carlo calculation at which fission occurs. The sites for cycle $l+1$ are sampled during the random walk process at each collision site in cycle l

according to the equation

$$n_{\mathrm{f}} = \left[\frac{1}{k_l} \frac{\nu\sigma_{\mathrm{f}}}{\sigma_{\mathrm{t}}} w \right] , \tag{10.3}$$

where $[\cdot]$ is the nearest integer notation, w is the current particle weight, k_l is the current estimate of the k-eigenvalue estimator, and n_{f} is the number of fission neutrons that will be sampled at this location in cycle $l+1$. The resulting fission sites are stored in a fission bank, and these sites constitute the source for the $l+1$ cycle. This sampling statistically preserves the requested total weight of particles, M, from cycle to cycle; however, poor estimates of k_l can cause large deviations from this value. To account for this statistical deviation, particles in cycle $l+1$ are born with a starting weight of M/N_l, where N_l is the number of fission sites sampled in cycle l. This scheme preserves the total weight, M, in each cycle.

In a parallel context, the sampling scheme outlined above can create load imbalance across sets between subsequent cycles. When a set statistically samples more fission sites than adjoining sets, the fission site population on that set will increase as the iteration progresses. This increase occurs because the fission site distribution attempts to create M fission sites globally. Thus, after several cycles, all of the particles will be born on a single set. To alleviate this pathology, the fission bank must be rebalanced between cycles. An iterative version of the fission bank rebalance algorithm described in Ref. [**50**] is used to redistribute the fission sites across sets. The basic algorithm requires modification to handle two specific cases in order to scale well:

(1) when sets are out of balance such that nearest neighbors do not have enough sites to send or
(2) when there are multiple blocks per set (MSOD).

To alleviate the first condition, the original algorithm is wrapped in an iteration scheme until full fission bank balance is achieved across all sets. To handle multiple blocks, an additional constraint must be applied that limits set-to-set transfers based on similar block populations of fission sites.

The algorithm to perform fission site rebalance is outlined below. The objective is to place the same number of fission sites on each set using

$$n_i = \begin{cases} \lfloor \frac{N_l}{N_{\mathrm{s}}} \rfloor + 1 & N_l \bmod N_{\mathrm{s}} > 0 \,, \\ \lfloor \frac{N_l}{N_{\mathrm{s}}} \rfloor & N_l \bmod N_{\mathrm{s}} = 0 \,, \end{cases} \tag{10.4}$$

where n_i is the desired number of fission sites on set i. If the global number of fission sites are monotonically ordered over the range $[0, N_l]$, the desired bounds $(\hat{a}_i, \hat{b}_i)$ on each set can be defined by the recurrence relation

$$\begin{aligned} \hat{a}_0 &= 0 \,, \\ \hat{b}_i &= \hat{a}_i + n_i - 1 \,, \\ \hat{a}_{i+1} &= \hat{b}_i + 1 \,, \quad i = 0, \ldots, N_{\mathrm{s}} \,. \end{aligned} \tag{10.5}$$

The actual bounds of the unbalanced fission bank on each set is given by (a_i, b_i) as illustrated in Fig. 10.3(a).

The goal of this algorithm is to align the actual fission bank boundaries on each set with the target boundaries. Thus, on each set Alg. 1 is applied, in which the actual fission bank boundaries are compared to the target boundaries and any surplus is communicated to the left-neighbor set $(i-1)$ or right-neighbor set $(i+1)$ as shown in Fig. 10.3(b). As long as fission sites received from the left and right neighbors are prepended and appended, respectively, to the local fission bank, the global ordering of the fission bank will not change; see Fig. 10.3(c). Thus, for fully domain-replicated problems ($N_{\mathrm{b}} = 1$), reproducibility can be preserved regardless of the size of N_{s}. Furthermore, since each set is only required to transfer sites with its left and right neighbors, the communication cost is local.

When the number of blocks per set is greater than one, the fission site rebalance must take into account the block structure within a set. Communication between sets is constrained to equivalent blocks, as shown in Fig 10.4. The number of sites that get passed between sets is determined by Alg. 1; furthermore, communication is constrained between equivalent blocks. The degree of freedom in Alg. 1 is the number of sites to pull from each block. For reactor problems, we desire equal numbers of particles on each block within a set to achieve ideal load balancing. This condition may not hold for other types of problems that have large

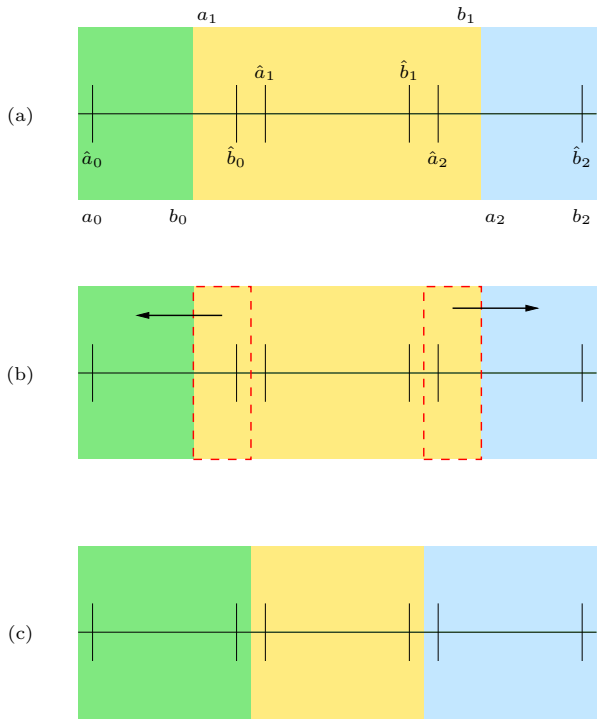


FIGURE 10.3. Rebalance boundaries across sets.

Algorithm 1 Fission bank rebalance algorithm across sets.

s_i = number of fission sites on set i
on set i calculate (a_i, b_i) and $(\hat{a}_i, \hat{b}_i)$
while $s_i \neq n_i$ **do**
    if $a_i < \hat{a}_i$ **then**
        $l_{\text{out}} = \min \hat{a}_i - a_i, s_i$
        send l_{out} to set $i - 1$
    end if
    if $b_i > \hat{b}_i$ **then**
        $r_{\text{out}} = \min b_i - \hat{b}_i, s_i$
        send r_{out} to set $i + 1$
    end if
    $s_i \leftarrow (s_i - l_{\text{out}} - r_{\text{out}})$
    receive l_{in} from $i - 1$
    receive r_{in} from $i + 1$
    $s_i \leftarrow (s_i + l_{\text{in}} + r_{\text{in}})$
end while

nonfissionable regions; however, the blocks for a reactor problem can generally be defined such that each block has a fission source of similar magnitude.

To preserve load balance, Shift calculates the average number of fission sites per block is in each set, $\bar{n}_{\text{b}}$. The number of surplus sites is calculated on each block, $s_{\text{b}} = n_{\text{b}} - \bar{n}_{\text{b}}$, where n_{b} is the number of sites on the block. Fission sites are communicated from this surplus by pulling sites uniformly from each block. If there are not enough surplus sites, regular sites are chosen uniformly from each block.

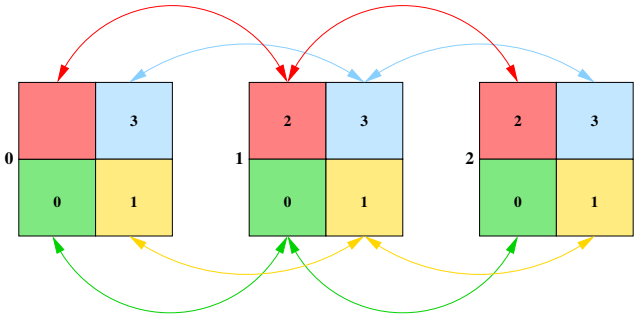


FIGURE 10.4. Illustration of block-to-block communication between sets during fission bank rebalance. Communication is constrained to equivalent blocks in adjacent sets.

3. Domain-Decomposed Solving Algorithms

As mentioned previously, to handle the block-to-block particle communication when performing a Monte Carlo transport solve, a modified domain-decomposition algorithm is used based upon Refs. [**51**] and [**52**]. This algorithm is robust and can handle the communication of particles that split between blocks. All sums in this algorithm, including global sums, are performed asynchronously. This asynchronicity removes the race conditions present in the original algorithm. Algorithm 2 gives the details of the domain-decomposed solving algorithm. This algorithm is broken into parts that are detailed in Algs. 3 - 7.

There are several parameters free to vary in this algorithm that are determined by user input:

- *numRequested* - total number of particle histories to run on a domain which is determined by the total number of particles histories in the problem divided by the number of domains,
- pbuffer size - number of particles to keep in the particle communication buffer before communicating,
- $checkfreq$ - frequency to check for incoming particles on the domain (this is an integer frequency of number of particles run, e.g. check every particle, $checkfreq = 1$, and check every other particle, $checkfreq = 2$.

These parameters affect the walltime of the problem including the communication cost.

Algorithm 2 Domain-decomposed solving algorithm.

```
Get list of neighboring processors
for each neighbor do
    Post a nonblocking receive with max particle buffer size
    Allocate the particle buffer
end for
numRun = numCompleted = 0
localCompletedSum = localOnlyCompleted = 0
localPlusChildCompleted = 0
localOnlyCreated = localCreatedSum = numRequested
localPlusChildCreated = numRequested
complete_flag = False
Set intraset communicator and binary tree proc pattern
for each child do
    Post nonblocking receive for all local plus child tallies
end for
Post nonblocking receive for control from parent
while !complete_flag do
    if source NOT empty then
        TRANSPORT_SOURCE_PARTICLE                          ▷ see Alg. 3
    end if
    if bank NOT empty then
        TRANSPORT_BANK_PARTICLE                            ▷ see Alg. 4
    end if
    if pbuffer NOT full AND pbuffer NOT empty then
        Send pbuffers to neighbors
    end if
    if numRun == checkfreq then
        Check for incoming particles
        PROCESS_MESSAGES                                   ▷ see Alg. 6
        numRun = 0
    end if
    if no active local particles then
        PROCESS_MESSAGES                                   ▷ see Alg. 6
        numRun = 0
        CONTROL_TERMINATION                                ▷ see Alg. 7
    end if
end while
Cancel all outstanding nonblocking receives
Check lost particles and output warning
```

Algorithm 3 Transporting algorithm for a source particle on a block.

```
TRANSPORT_PARTICLE                                         ▷ see Alg. 5
increment particle_history
if numRun mod checkfreq == 0 then
    if incoming particles then
        while bank NOT empty do
            TRANSPORT_BANK_PARTICLE                        ▷ see Alg. 4
        end while
    end if
end if
```

Algorithm 4 Transporting algorithm for a banked particle on a block.

```
Get particle from bank
TRANSPORT_PARTICLE                                   ▷ See Alg. 5
if numRun mod checkfreq == 0 then
    Add incoming particles to bank
end if
```

Algorithm 5 Transporting algorithm for a particle on a block.

```
Perform transport through domain
if particle NOT killed then
    increment numRun
end if
localOnlyCreated += numSecondaryParticles
localPlusChildCreated += numSecondaryParticles
if particle hit boundary then
    Send particle to neighbor block if still in problem
end if
increment numCompleted
increment localOnlyCompleted
increment localPlusChildCompleted
localCreatedSum = localOnlyCreated
localCompletedSum = localOnlyCompleted
```

Algorithm 6 Domain-decomposed algorithm for processing messages.

```
1: for all children do
2:     if numDone received from child then
3:         localPlusChildCompleted += numDone
4:         Post a new receive from this child (DONE)
5:     end if
6:     if numCreated received from child then
7:         localPlusChildCreated += numCreated
8:         Post a new receive from this child (CREATED)
9:     end if
10: end for
11: localCompletedSum = localOnlyCompleted
12: localCreatedSum = localOnlyCreated
13: if !MASTER then
14:     if control message received from parent then
15:         Send control message to children (CONTROL)
16:         done_recv = False
17:         created_recv = False
18:         for all children do
19:             if numDoneGlob received from child then
20:                 localCompletedSum += numDoneGlob
21:                 Post a new receive from this child (DONEGLOB)
22:                 done_recv = True
23:             end if
24:             if numCreatedGlob received from child then
25:                 localCreatedSum += numCreatedGlob
26:                 Post a new receive from this child (CREATEDGLOB)
27:                 created_recv = True
28:             end if
29:         end for
30:         if (done_recv) OR (num_children == 0) then
31:             Send localCompletedSum to parent (DONEGLOB)
32:         end if
33:         if (created_recv) OR (num_children == 0) then
34:             Send localCreatedSum to parent (CREATEDGLOB)
35:         end if
36:         localCompletedSum = localOnlyCompleted
37:         localCreatedSum = localOnlyCreated
38:         Repost control message receive from parent
39:     end if
40: end if
```

Algorithm 7 Domain-decomposed algorithm for control termination.

```
1: if MASTER then
2:     if localPlusChildCompleted == localPlusChildCreated! = 0 then
3:         Send control message to children (CONTROL)
4:         localCompletedSum = localOnlyCompleted
5:         localCreatedSum = localOnlyCreated
6:         for all children do
7:             if numDoneGlob received from child then
8:                 localCompletedSum += numDoneGlob
9:                 Post a new recieve from this child (DONEGLOB)
10:            end if
11:            if numCreatedGlob received from child then
12:                localCreatedSum += numCreatedGlob
13:                Post a new receive from this child (CREATEDGLOB)
14:            end if
15:        end for
16:        if localCompletedSum == localCreatedSum then
17:            if lPCCompleted == localCompletedSum
18:                AND lPCCreated == localCreatedSum then
19:                if lPCCompleted >= numRequested then
20:                    complete_flag = True
21:                end if
22:            end if
23:            p_child = 0
24:            while (!complete_flag) AND (p_child < numChildren) do
25:                if numDone received from child then
26:                    localPlusChildCompleted += numDone
27:                    Post a new receive from this child (DONE)
28:                    if (lPCCompleted == localCompletedSum)
29:                          AND (lPCCreated == localCreatedSum) then
30:                        if lPCCompleted >= numRequested then
31:                            complete_flag = True
32:                        end if
33:                    end if
34:                end if
35:                if numCreated received from child then
36:                    localPlusChildCreated += numCreated
37:                    Post a new receive from this child (CREATED)
```

Algorithm 7 DD algorithm for control termination (continued).

```
38:                     if (lPCCompleted == localCompletedSum)
39:                           AND (lPCCreated == localCreatedSum) then
40:                        if lPCCompleted >= numRequested then
41:                           complete_flag = True
42:                        end if
43:                     end if
44:                 end if
45:             end while
46:             if complete_flag then
47:                 Send complete flag to children (COMPLETE)
48:             else
49:                 localCompletedSum = localOnlyCompleted
50:                 localCreatedSum = localOnlyCreated
51:             end if
52:         end if
53:     end if
54: else
55:     if localPlusChildCompleted > 0 then
56:         Send to parent (DONE)
57:         localPlusChildCompleted = 0
58:     end if
59:     if localPlusChildCreated > 0 then
60:         Send to parent (CREATED)
61:         localPlusChildCreated = 0
62:     end if
63:     if complete received from parent then
64:         complete_flag = True
65:         Send complete message to children (COMPLETE)
66:     end if
67: end if
```

Part 3

Application-Specific Methods

CHAPTER 11

Methods for Reactor Analysis

This chapter describes supporting numerical technologies that are required to execute the transport methods described in Parts 1-2 for the purposes of performing reactor analysis. In the following sections we describe algorithms for mesh generation of lightwater reactor models (§ 1) and cross section generation (§ 2).

1. Mesh Generation

All of the deterministic transport methods described in Part 1 are defined for regular Cartesian grids. A necessary condition for k-eigenvalue problems is the exact (to numerical precision) conservation of fissionable material in the problem. Therefore, each Cartesian mesh cell must be mixed to preserve the exact fuel volume. Consider a PWR pin-cell illustrated in Fig. 11.1 Assume, as shown in the figure, that we wish to radially mesh the pin using an 8×8 grid. The meshing algorithm uses symmetry to only mesh 1/8th (for $N \times N$ meshes) or 1/4th (for $N \times M$ meshes) of the cells. As shown in the figure, there are two cell types: clean and mixed. In this example the resulting meshing produces 9 unique mixtures:

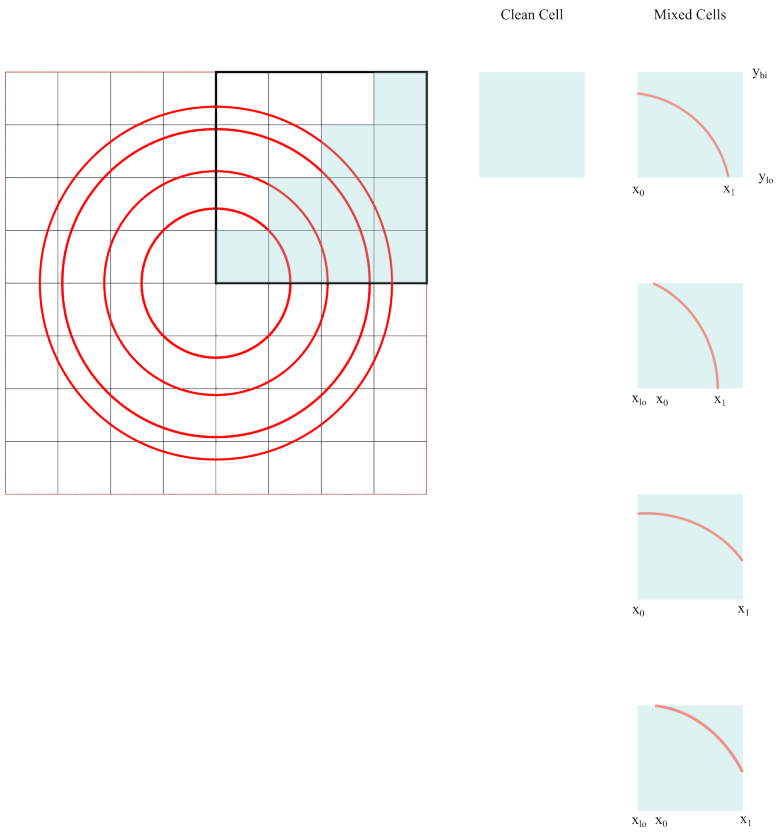


FIGURE 11.1. PWR pin-cell meshed with an 8×8 Cartesian grid.

1	1	7	4	4	7	1	1
1	8	6	3	3	6	8	1
7	6	5	2	2	5	6	7
4	3	2	0	0	2	3	4
4	3	2	0	0	2	3	4
7	6	5	2	2	5	6	7
1	8	6	3	3	6	8	1
1	1	7	4	4	7	1	1

Mixtures 0 and 1 are clean fuel and moderator, respectively. The additional cells are all mixtures of multiple materials. For example, if we apply region labels of 0–3 to each cylindrical shell and label 4 to the moderator, the mixture table corresponding to this mesh is

	Region					
Mixture	0	1	2	3	4	Cell type
0	*	0.0	0.0	0.0	0.0	clean
1	0.0	0.0	0.0	0.0	*	clean
2	*	*	*	0.0	0.0	mixed
3	0.0	*	*	*	0.0	mixed
4	0.0	0.0	0.0	*	*	mixed
5	0.0	*	*	0.0	0.0	mixed
6	0.0	0.0	*	*	*	mixed
7	0.0	0.0	0.0	*	*	mixed
8	0.0	0.0	*	*	*	mixed

Here, a star indicates a non-zero volume fraction for that region in the cell. Thus, mixture 3 consists of volume contributions from regions 1, 2, and 3.

The volume subtended by a shell in a cylindrical shell segment can be calculated by integrating

$$f(x) = \sqrt{r^2 - x^2} \tag{11.1}$$

Additionally, we must add any component of the volume in the cell outside of the shell segment, and we must subtract the component outside the cell. Thus, we have

$$V_{\text{shell}} = \int_{x_0}^{x_1} f(x)dx + (y_{\text{hi}} - y_{\text{lo}})(x_0 - x_{\text{lo}}) - y_{\text{lo}}(x_1 - x_0) \tag{11.2}$$

where the integral is

$$\frac{1}{2}\left[x\sqrt{r^2 - x^2} + r^2 \tan^{-1}\left(\frac{x}{\sqrt{r^2 - x^2}}\right)\right]\Bigg|_{x_0}^{x_1} \tag{11.3}$$

As illustrated in the figure, *xlo, xhi, ylo, yhi* are the (x, y) low and high boundaries of the mesh cell. There are 4-basic types of shell-cell intersections, each of which is illustrated in the figure. This algorithm is only valid for $N \times N$ and $N \times M$ meshes in which $N, M \in \{2k; \forall k \in \mathcal{Z}\}$.

2. Cross Section Generation

Cross sections can be generated for reactor problems using many means. The details of cross section generation form a field unto itself and is thus beyond the scope of this text. The algorithms that we use for cross section generation are described in the SCALE code system documentation in Ref. [**53**]. For the purposes of performing transport calculations, each Cartesian mesh cell requires a macroscopic cross section. There are two methods for doing this: (a) generate cross sections for a homogenized pincell, and (b) use volume weighting to mix the cross sections. In the former case, every mesh cell discretizing a pin-cell gets the same cross section because the cross section generation procedure has already homogenized the cell. SCALE has routines that can perform this homogenization.

In the second case the volume fractions of each mesh cell must be calculated using the methods described in § 1. Once the volume fractions in each mesh cell are known, the cross section for mesh cell c can be constructed from the macroscopic cross sections in each region r via

$$\sigma_c = \sum_r \sigma_r V_r \ , \quad c = 0, 1, 2, \ldots N_c - 1 \ , \tag{11.4}$$

where N_c is the number of mesh cells discretizing a pin-cell (16 in the example shown in Fig. 11.1).

Bibliography

[1] T. Evans, A. Stafford, R. Slaybaugh, and K. Clarno, "DENOVO: A new three-dimensional parallel discrete ordinates code in SCALE," *Nuclear Technology*, vol. 171, pp. 171–200, 2010.

[2] E. Lewis and J. W.F. Miller, *Computational Methods of Neutron Transport*. LaGrange Park, IL: American Nuclear Society, Inc., 1993.

[3] J. Duderstadt and W. Martin, *Transport Theory*. Wiley-Interscience Publications, Books on Demand, 1979.

[4] J. Shultis and R. Faw, *Radiation shielding*. Prentice Hall PTR, 1996.

[5] G. Arfken, *Mathematical Methods for Physicists*. New York: Academic Press, Inc., third ed., 1985.

[6] E. Gelbard, "Application of the spherical harmonics to reactor problems," Tech. Rep. WAPD-BT-20, Westinghouse Atomic Power Department, Sept 1960.

[7] M. Frank, A. Klar, E. W. Larsen, and S. Yasuda, "Time-dependent simplified PN approximation to the equations of radiative transfer," *Journal of Computational Physics*, vol. 226, pp. 2289–2305, Oct. 2007.

[8] E. W. Larsen, G. Thömmes, A. Klar, M. Sead, and T. Götz, "Simplified PN Approximations to the Equations of Radiative Heat Transfer and Applications," *Journal of Computational Physics*, vol. 183, pp. 652–675, Dec. 2002.

[9] E. W. Larsen, J. E. Morel, and J. M. McGhee, "Asymptotic Derivation of the Multigroup P1 and Simplified PN Equations with Anisotropic Scattering," *Nuclear Science and Engineering*, vol. 123, pp. 328–342, 1996.

[10] P. S. Brantley and E. W. Larsen, "The Simplified P3 Approximation," *Nuclear Science and Engineering*, vol. 134, pp. 1–21, 2000.

[11] A. D. Klose and E. W. Larsen, "Light transport in biological tissue based on the simplified spherical harmonics equations," *Journal of Computational Physics*, vol. 220, pp. 441–470, Dec. 2006.

[12] J. E. Morel, J. M. McGhee, and E. W. Larsen, "A Three-Dimentional Time-Dependent Unstructured Tetrahedral-Mesh SPN Method," *Nuclear Science and Engineering*, vol. 123, pp. 1–9, 1996.

[13] J. A. Josef and J. E. Morel, "Simplified spherical harmonic method for coupled electron-photon transport calculations," *Physical Review E*, vol. 57, pp. 6161–6171, May 1998.

[14] W. Rhodes and J. W.W. Engle, "A new weighted difference formulation for discrete ordinates calculations," *Transactions of the American Nuclear Society*, vol. 27, p. 776, 1977.

[15] M. Adams, "Discontinuous finite element transport solutions in thick diffusive problems," *Nuclear Science and Engineering*, vol. 137, pp. 298—333, 2001.

[16] K. Lathrop, "Spatial differencing of the transport equation: Positivity vs accuracy," *Journal of Computational Physics*, vol. 4, pp. 475–498, 1969.

[17] M. Adams, T. Wareing, and W. Walters, "Characteristic methods in thick diffusive problems," *Nuclear Science and Engineering*, vol. 130, pp. 18–46, 1998.

[18] W. Rhodes and R. Childs, "TORT: A three-dimensional discrete ordinates neutron/photon transport code," *Nuclear Science and Engineering*, vol. 107, no. 4, pp. 397–398, 1991.

[19] J. Reddy, *An Introduction to the Finite Element Method*. McGraw-Hill, Inc., second ed., 1993.

[20] Y. Zhang, J. Ragusa, and J. Morel, "Iterative performance of various formulations of the SP_N equations," *Journal of Computational Physics*, vol. 252, pp. 558–572, 2013.

[21] B. Adams and J. Morel, "A two-grid acceleration scheme for the multigroup s_n equations with neutron upscattering," *Nuclear Science and Engineering*, vol. 115, pp. 253–264, 1993.

[22] T. Evans, K. Clarno, and J. Morel, "A transport acceleration scheme for multigroup discrete ordinates with upscattering," *Nuclear Science and Engineering*, vol. 165, no. 3, pp. 292–304, 2010.

[23] V. Faber and T. Manteuffel, "A look at transport theory from the point of view of linear algebra," in *Lecture Notes in Pure and Applied Mathematics* (P. Nelson *et al.*, eds.), vol. 115, (New York), pp. 31–61, Marcel Dekker, 1988.

[24] G. Golub and C. van Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 3 ed., 1996.

[25] E. J. Allen and R. M. Berry, "The inverse power method for calculation of multiplication factors," *Annals of Nuclear Energy*, vol. 29, pp. 929–935, 2002.

[26] F. Scheben and I. G. Graham, "Iterative methods for neutron transport eigenvalue problems," *SIAM Journal on Scientific Computing*, vol. 33, pp. 2785–2804, 2011.

[27] S. Hamilton, *Numerical Solution of the k-Eigenvalue Problem*. PhD thesis, Emory University, Atlanta, GA, 2011.

[28] R. N. Slaybaugh, T. M. Evans, G. G. Davidson, and P. P. H. Wilson, "Rayleigh quotient iteration in 3D, deterministic neutron transport," in *Advances in Reactor Physics Linking Research, Industry, and Education (PHYSOR 2012)*, (Knoxville, TN), 2012.

[29] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*. Manchester, UK: Manchester University Press, 1992.

[30] W. E. Arnoldi, "The principle of minimized iterations in the solution of matrix eigenvalue problems," *Quarterly of Applied Mathematics*, vol. 9, pp. 17–29, 1951.

[31] J. S. Warsa, T. A. Wareing, J. E. Morel, J. M. McGhee, and R. B. Lehoucq, "Krylov subspace iterations for deterministic k-eigenvalue calculation," *Nuclear Science and Engineering*, vol. 147, pp. 26–42, 2004.

[32] G. G. Davidson, T. M. Evans, J. J. Jarrell, S. P. Hamilton, T. M. Pandya, and R. N. Slaybaugh, "Massively parallel, three-dimensional transport solutions for the k-eigenvalue problem," *Nuclear Science and Engineering*, 2013. , forthcoming.

[33] E. R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *Journal of Computational Physics*, vol. 17, pp. 87–94, 1975.

[34] R. B. Morgan and D. S. Scott, "Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 817–825, 1986.

[35] M. Sadkane, "Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems," *Numerische Mathematik*, vol. 64, pp. 195–211, 1993.

[36] S. Hamilton and M. Benzi, "A Davidson method for the k-eigenvalue problem," *Transactions of the American Nuclear Society*, vol. 105, pp. 432–434, 2011.

[37] S. Subramanian, S. van Criekingen, V. Heuveline, F. Nataf, and P. Havé, "The Davidson method as an alternative to power iterations for criticality calculations," *Annals of Nuclear Energy*, vol. 38, no. 12, pp. 2818–2823, 2011.

[38] G. L. G. Sleijpen and H. A. van der Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 2, pp. 401–425, 1996.

[39] D. Knoll, H. Park, and C. Newman, "Acceleration of k-eigenvalue/criticality calculations using the Jacobian-free Newton-Krylov method," *Nuclear Science and Engineering*, vol. 167, no. 2, pp. 133–140, 2011.

[40] D. Gill and Y. Azmy, "Newton's method for solving k-eigenvalue problems in neutron diffusion theory," *Nuclear Science and Engineering*, vol. 167, no. 2, pp. 141–153, 2011.

[41] R. B. Morgan, "Davidson's method and preconditioning for generalized eigenvalue problems," *Journal of Computational Physics*, vol. 89, pp. 241–245, 1990.

[42] M. Crouzeix, B. Philippe, and M. Sadkane, "The Davidson method," *SIAM Journal on Scientific Computing*, vol. 15, pp. 62–76, 1994.

[43] J. Olsen, P. Jorgensen, and J. Simons, "Passing the one-billion limit in full configuration-interaction (FCI) calculations," *Chemical Physics Letters*, vol. 169, pp. 463–472, 1990.

[44] L. Carter and E. Cashwell, "Particle transport simulation with the Monte Carlo method," Tech. Rep. TID-26607, Los Alamos National Laboratory, 1975.

[45] A. Chilton, J. Shultis, and R. Faw, *Principles of Radiation Shielding*. New York: Prentice Hall, 1984.

[46] N. Schaeffer, "Reactor shielding for nuclear engineers," Tech. Rep. TID-25951, U.S. Atomic Energy Commission, 1973.

[47] J. Turner, H. Wright, and R. Hamm, "A Monte Carlo primer for health physicists," *Health Physics*, vol. 48, pp. 717–733, 1985.

[48] X-5 Monte Carlo Team, "MCNP - a general Monte Carlo n-particle transport code, version 5," Tech. Rep. LA-UR-03-1987, Los Alamos National Laboratory, 2008.

[49] J. C. Wagner, S. W. Mosher, T. M. Evans, D. E. Peplow, and J. A. Turner, "Hybrid and parallel domain-decomposition methods development to enable Monte Carlo for reactor analyses," *Progress in Nuclear Science Technology*, vol. 2, pp. 815–820, 2011.

[50] P. Romano and B. Forget, "Parallel fission bank algorithms in Monte Carlo criticality calculations," *Nuclear Science and Engineering*, vol. 170, pp. 125–135, 2012.

[51] T. A. Brunner, T. J. Urbatsch, T. M. Evans, and N. A. Gentile, "Comparison of four parallel algorithms for domain decomposed implicit Monte Carlo," *Journal of Computational Physics*, vol. 212, pp. 527–539, 2006.

[52] T. A. Brunner and P. S. Brantley, "An efficient, robust, domain-decomposition algorithm for particle Monte Carlo," *Journal of Computational Physics*, vol. 228, pp. 3882–3890, February 2009.

[53] "SCALE: A comprehensive modeling and simulation suite for nuclear safety analysis and design," Tech. Rep. ORNL/TM-2005/39, Version 6.1, Oak Ridge National Laboratory, Oak Ridge, TN, 2011.

APPENDIX A

Spherical Harmonics Expansions

Any function of $\mathbf{\Omega}$ can be expanded in Spherical Harmonics using

$$f(\mathbf{\Omega}) = \sum_{l=0}^{N} \sum_{m=-l}^{l} Y_{lm}(\mathbf{\Omega}) f_{lm} \,. \tag{A.1}$$

The $Y_l m$ are the Spherical Harmonics, and they constitute a complete, orthonormal set. They are defined

$$Y_{lm}(\theta,\varphi) = (-1)^m \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\, P_{lm}(\cos\theta) \mathrm{e}^{im\varphi} \,, \tag{A.2}$$

where the P_{lm} are the associated Legendre Polynomials. We note that some treatments prefer to include the Condon-Shortley phase factor, $(-1)^m$, in the definition of P_{lm}, whereas here it is applied to the definition of Y_{lm}. The orthogonality of Spherical Harmonics is

$$\int_{4\pi} Y^*_{l'm'} Y_{lm}\, d\mathbf{\Omega} = \delta_{ll'}\delta_{mm'} \,. \tag{A.3}$$

Using this property the moments of f are calculated

$$f_{lm} = \int_{4\pi} Y^*_{lm}(\mathbf{\Omega}) f(\mathbf{\Omega})\, d\mathbf{\Omega} \,. \tag{A.4}$$

If $f \in \mathscr{R}$ then the expansion given in Eq. (A.1) will be inappropriate because it is complex. We seek a real expansion of f. First, we rewrite Eq.(A.1) by explicitly breaking it into $\pm$ components of m,

$$f = \sum_{l=0}^{N} \sum_{m=0}^{l} \left[C_{lm} P_{lm} f_{lm} \mathrm{e}^{im\varphi} + C_{l-m} P_{l-m} \mathrm{e}^{-im\varphi} \right] , \tag{A.5}$$

where

$$C_{lm} = (-1)^m \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} \tag{A.6}$$

Noting that the imaginary terms vanish when $m = 0$, we define the complex form of f_{lm} as follows,

$$f_{lm} = \begin{cases} \alpha_{lm} & m = 0 \,, \\ \alpha_{lm} + i\beta_{lm} & m > 0 \,. \end{cases} \tag{A.7}$$

Expanding $\exp(im\varphi) = \cos m\varphi + i \sin m\varphi$, we write

$$f = \sum_{l=0}^{N} \sum_{m=0}^{l} \left[C_{lm} P_{lm} (\alpha_{lm} + i\beta_{lm})(\cos m\varphi + i \sin m\varphi) + C_{l-m} P_{l-m} (\alpha_{l-m} + i\beta_{l-m})(\cos m\varphi - i \sin m\varphi) \right] . \tag{A.8}$$

Grouping terms into real and imaginary parts gives

$$\begin{aligned} f = \sum_{l=0}^{N} \sum_{m=0}^{l} \Big[& (C_{lm} P_{lm} \alpha_{lm} + C_{l-m} P_{l-m} \alpha_{l-m}) \cos m\varphi + (C_{l-m} P_{l-m} \beta_{l-m} - C_{lm} P_{lm} \beta_{lm}) \sin m\varphi \\ & + i \big[(C_{lm} P_{lm} \beta_{lm} + C_{l-m} P_{l-m} \beta_{l-m}) \cos m\varphi + (C_{lm} P_{lm} \alpha_{lm} - C_{l-m} P_{l-m} \alpha_{l-m}) \sin m\varphi \big] \Big] \,. \end{aligned} \tag{A.9}$$

Because $f \in \mathscr{R}$, the imaginary parts must vanish, which gives the following condition:

$$C_{l-m}P_{l-m}\beta_{l-m} = -C_{lm}P_{lm}\beta_{lm}\,, \tag{A.10}$$

$$C_{l-m}P_{l-m}\alpha_{l-m} = C_{lm}P_{lm}\alpha_{lm}\,. \tag{A.11}$$

Substituting these expressions into Eq. (A.9), yields

$$f(\mathbf{\Omega}) = \sum_{l=0}^{N}\sum_{m=0}^{l} C_{lm}P_{lm}(2\alpha_{lm}\cos m\varphi - 2\beta_{lm}\sin m\varphi)\,. \tag{A.12}$$

The new basis for the real expansion of f consists of the following functions

$$\hat{Y}^e_{lm} = C_{lm}P_{lm}\cos m\varphi\,, \tag{A.13}$$

$$\hat{Y}^o_{lm} = C_{lm}P_{lm}\sin m\varphi\,, \tag{A.14}$$

and

$$\begin{aligned} Y_{lm} &= C_{lm}P_{lm}\cos m\varphi + iC_{lm}P_{lm}\sin m\varphi \\ &= \hat{Y}^e_{lm} + i\hat{Y}^o_{lm}\,. \end{aligned} \tag{A.15}$$

This basis needs to constitute an orthonormal set, applying orthogonality to the even harmonic gives

$$\begin{aligned} \int_{4\pi} \hat{Y}^e_{lm}\hat{Y}^e_{l'm'}\,d\mathbf{\Omega} &= C^2_{lm}\frac{2\pi}{2l+1}\frac{(l+m)!}{(l-m)!}\delta_{ll'}(1+\delta_{m0})\delta_{mm'} \\ &= \frac{(1+\delta_{m0})}{2}\delta_{mm'}\delta_{ll'}\,, \end{aligned} \tag{A.16}$$

where we have used orthogonality relations for cos and P_{lm} listed in Chap. B. In order for $\hat{Y}^e_{lm}$ to be orthonormal, we must have

$$\mathcal{C}^2\frac{(1+\delta_{m0})}{2}\delta_{mm'}\delta_{ll'} = 1\,, \tag{A.17}$$

which yields the following conditions on $\mathcal{C}$,

$$\mathcal{C} = \begin{cases} 1 & m = 0\,, \\ \sqrt{2} & m > 0\,. \end{cases} \tag{A.18}$$

Defining $\mathcal{C} = \sqrt{2-\delta_{m0}}$ satisfies both constraints. The odd harmonics are developed similarly with the exception that they vanish when $m = 0$, ie.

$$\int_{4\pi} \hat{Y}^o_{lm}\hat{Y}^o_{l'm'}\,d\mathbf{\Omega} = \frac{(1-\delta_{m0})}{2}\delta_{ll'}\delta_{mm'}\,. \tag{A.19}$$

Applying $\mathcal{C}$ to $\hat{Y}^e$ and $\hat{Y}^o$, the real basis for f is

$$Y^e_{lm} = D_{lm}P_{lm}\cos m\varphi\,, \tag{A.20}$$

$$Y^o_{lm} = D_{lm}P_{lm}\sin m\varphi\,, \tag{A.21}$$

where

$$D_{lm} = (-1)^m\sqrt{(2-\delta_{m0})\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\,. \tag{A.22}$$

The orthogonality conditions of Y^e and Y^o are now correctly defined

$$\int_{4\pi} Y^e_{lm}Y^e_{l'm'}\,d\mathbf{\Omega} = \frac{(2-\delta_{m0})(1+\delta_{m0})}{2}\delta_{mm'}\delta_{ll'} = \delta_{mm'}\delta_{ll'}\,, \tag{A.23}$$

$$\int_{4\pi} Y^o_{lm}Y^o_{l'm'}\,d\mathbf{\Omega} = \frac{(2-\delta_{m0})(1-\delta_{m0})}{2}\delta_{mm'}\delta_{ll'} = (1-\delta_{m0})\delta_{mm'}\delta_{ll'}\,. \tag{A.24}$$

Defining

$$a_{lm} = \sqrt{2 - \delta_{m0}}\, \alpha_{lm} \,, \tag{A.25}$$

$$b_{lm} = \sqrt{2}(\delta_{m0} - 1)\beta_{lm} \,, \tag{A.26}$$

and applying the set in Eqs. (A.20) and (A.21) to Eq. (A.12) yields

$$\boxed{f(\mathbf{\Omega}) = \sum_{l=0}^{N} \Big[Y_{l0}^{e} a_{l0} + \sum_{m=1}^{l} (Y_{lm}^{e} a_{lm} + Y_{lm}^{o} b_{lm}) \Big] \,.} \tag{A.27}$$

Applying the orthogonality conditions, Eqs. (A.23) and (A.24), to Eq. (A.27) defines the moments, a_{lm} and b_{lm},

$$a_{lm} = \int_{4\pi} Y_{lm}^{e}(\mathbf{\Omega}) f(\mathbf{\Omega})\, d\mathbf{\Omega} \,, \quad m \geq 0 \,, \tag{A.28}$$

$$b_{lm} = \int_{4\pi} Y_{lm}^{o}(\mathbf{\Omega}) f(\mathbf{\Omega})\, d\mathbf{\Omega} \,, \quad m > 0 \,. \tag{A.29}$$

Equation (A.27) defines the Spherical Harmonics expansion of a real-valued function $f(\mathbf{\Omega})$. The orthonormal Spherical Harmonics used in Eq. (A.27) are defined in Eqs. (A.20) and (A.21). The moments of f are defined in Eqs. (A.28) and (A.29).

APPENDIX B

Mathematical Properties and Identities

Useful orthogonality properties that are used in the text are:

$$\int_0^{2\pi} \cos m'x \cos mx \, dx = \pi(1+\delta_{m0})\delta_{mm'} \,, \qquad \text{cosine} \tag{B.1}$$

$$\int_0^{2\pi} \sin m'x \sin mx \, dx = \pi(1-\delta_{m0})\delta_{mm'} \,, \qquad \text{sine} \tag{B.2}$$

$$\int_{-1}^{1} P_n(x)P_m(x) \, dx = \frac{2}{2n+1}\delta_{nm} \,, \qquad \text{Legendre polynomials} \tag{B.3}$$

$$\int_{-1}^{1} P_{lm}(x)P_{l'm}(x) \, dx = \frac{2}{2l+1}\frac{(l+m)!}{(l-m)!}\delta_{ll'} \,. \qquad \text{associated Legendre polynomials} \tag{B.4}$$

The relationship between m and $-m$ associated Legendre polynomials is

$$P_{l-m} = (-1)^m \frac{(l-m)!}{(l+m)!} P_{lm} \,. \tag{B.5}$$

In Eqs. (A.13) and (A.14) the complex components of the Spherical Harmonics are given that satisfy

$$Y_{lm} = \hat{Y}^e_{lm} + i\hat{Y}^o_{lm} \,. \tag{B.6}$$

Using Eq. (B.5) the relationship between the positive and negative m-components of Y_{lm} can be found:

$$\begin{aligned} \hat{Y}^e_{l-m} &= (-1)^m C_{l-m} \frac{(l-m)!}{(l+m)!} P_{lm} \cos m\varphi \\ &= \sqrt{\frac{2l+1}{4\pi}\frac{(l+m)!}{(l-m)!}}\frac{(l-m)!}{(l+m)!} P_{lm} \cos m\varphi \\ &= (-1)^{-m}\hat{Y}^e_{lm} \equiv (-1)^m \hat{Y}^e_{lm} \,. \end{aligned} \tag{B.7}$$

Similarly,

$$\hat{Y}^o_{l-m} = -(-1)^m \hat{Y}^o_{lm} \,. \tag{B.8}$$