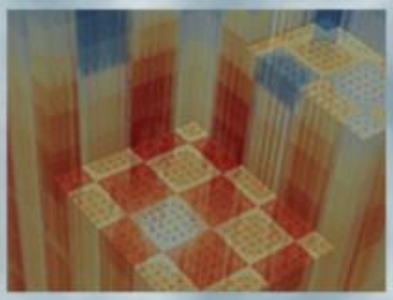
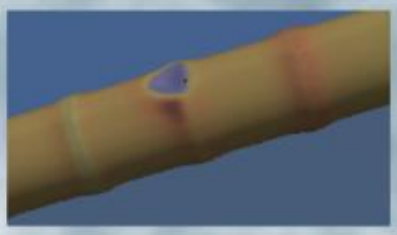
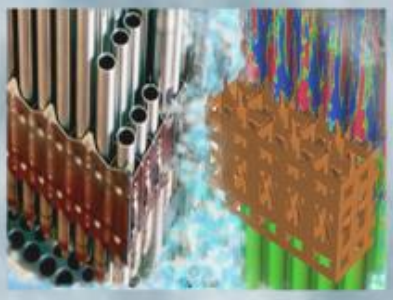
CASL-U-2019-1874-001

Consortium for Advanced Simulation of LWRs







MPACT Theory Manual

Version 4.1

Edited by Edward Larsen[1],

Benjamin Collins[2], Brendan Kochunas[1], and Shane Stimpson[2]

[1]University of Michigan

[2]Oak Ridge National Laboratory

11/1/2019

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

Revision Log

Revision	Date	Affected Pages	Revision Description
0	7/24/2019	All	Initial Release
1	11/1/2019	10, 15–16, 18, 32, 38, 43, 49, 54, 56–57, 59, 60, 63, 66, 76–78, 87, 89–91, 94, 95, 97, 100, 102, 104–105, 115–119, 122, 129–130, 137	Typographical Corrections

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

MPACT Theory Manual

Approvals:

Digitally signed by Brendan Kochunas
Date: 2019.11.11 20:49:40 -05'00'

11/11/2019

Brendan Kochunas, MPACT Product Software Manager **Date**

Digitally signed by Yuxuan Liu
Date: 2019.11.11 16:28:23 -05'00'

11/11/2019

Yuxuan Liu, Independent Reviewer **Date**

Contributors (in alphabetical order)

- Dr. Kevin Clarno
- Dr. Benjamin Collins
- Dr. Luke Cornejo

- Zackary Dodson
- Prof. Thomas J. Downar
- Andrew Fitzgerald
- Dr. Jess Gehin
- Dr. Cole Gentry

- Andrew Godfrey
- Dr. Aaron Graham
- Shane Henderson

- Daniel Jabaay
- Dr. Michael Jarrett
- Dr. Blake Kelley
- Dr. Kang Seog Kim
- Prof. Brendan Kochunas
- Prof. Edward Larsen
- Dr. Yuxuan Liu
- Dr. Zhouyu Liu
- Prof. William R. Martin
- Dr. Adam Nelson
- Dr. Scott Palmtag
- Dr. Michael Rose
- Dr. Thomas Saller
- Qicang Shen
- Dr. Shane Stimpson
- Dr. Travis Trahan
- Dr. Erik Walker

- Dr. Jipu Wang
- Dr. Will Wieselquist
- Xinyan Yang

- Dr. Ben C. Yee
- Dr. Mitchell T. H. Young
- Dr. Ang Zhu

Acknowledgments:

The development of MPACT has been supported by the University of Michigan, Oak Ridge National Laboratory, and the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under US Department of Energy Contract No. DE-AC05-00OR22725.

Abstract

This manual presents the theory underlying the three-dimensional (3D) whole-core, pin-resolved neutron transport calculation methodologies employed in the MPACT code. MPACT's primary goal is to provide accurate sub-pin power distributions in a computationally efficient manner. To accomplish this, MPACT offers several different transport method options. The 2D/1D method, in which 3D problems are decomposed into an axial stack of radial "slices," is currently the most commonly used. In this option, two-dimensional (2D) planar solutions are provided by the method of characteristics (MOC), and axial solutions are provided via one-dimensional (1D) approximate diffusion or P_3 solutions. The radial and axial solutions are coupled by (i) axial and radial transverse leakages, and (ii) a global 3D coarse mesh finite difference (CMFD) solve, which provides both acceleration and stability to the solution iteration scheme.

The subsequent chapters of this manual present a range of topics, including MOC, CMFD, axial nodal transport solvers, 2D/1D, self-shielding, depletion, thermal-hydraulics, and transient methods. The underlying theory of the 2D/1D method is developed, diagrams are included to highlight important algorithmic flow, and important concepts are discussed as appropriate. This manual is intended to be self-sufficient, but references to published articles and other materials are included for further reading.

MPACT is a relatively new code, with new capabilities and many computational methods that did not exist until recently. This manual is an even newer document with chapters written by several different code contributors working under time constraints. For these reasons, the manual is not yet be complete and finalized. Future versions of this manual will address current deficiencies.

Contents

1 Introduction **1**

2 The Linear Boltzmann Transport Equation **4**

2.1 The Boltzmann Equation 4

2.2 The Multigroup Approximation 6

2.3 The Discrete Ordinates (S_N) Approximation 9

3 Summary of Algorithms **12**

4 The 2D/1D Method **16**

4.1 Historical Overview 16

4.2 Preliminaries 17

4.3 The Basic 2D/1D Equations 19

4.4 Discretizations 21

4.5 Iteration Strategy 24

4.6 Appendix A: Relaxation Strategy 26

5 The Method of Characteristics Solution Methodology **32**

5.1 2D Radial Solution Methodology by the MOC 32

5.2 Method of Characteristics Solution of the Boltzmann Transport Equation in 3D 34

5.3 Approximations of the Characteristics Transport Equation . . . 35
5.3.1 Constant Material Properties in a Discrete Region . . . 35
5.3.2 Flat Source Approximation . . . 36
5.3.3 Isotropic Scattering Source Approximation . . . 37
5.4 Discretization of the Characteristics Equations . . . 39
5.5 Iteration Schemes . . . 43
5.5.1 Source Iteration . . . 43
5.5.2 Convergence Criterion for Source Iteration . . . 45
5.5.3 MOC Sweep Procedure . . . 46
5.5.4 Gauss-Seidel and Jacobi Inscatter Sweeping Algorithms . . . 46

6 1D Axial Solution Methodology 51

6.1 One-Dimensional Axial Governing Equations . . . 51
6.1.1 Transport-Based . . . 51
6.1.2 Diffusion-Based Solvers . . . 54
6.2 One-Node vs. Two-Node . . . 54
6.3 Nodal Expansion Method (NEM) . . . 56
6.4 Source Expansion Nodal Method (SENM) . . . 58
6.5 Spherical Harmonics (P_N) . . . 58
6.6 Discrete Ordinates (S_N) . . . 60
6.6.1 Spatial Moments . . . 60
6.6.2 Azimuthal Moments . . . 61
6.7 Transverse Leakage Interpolation . . . 63

7 Coarse Mesh Finite Difference Acceleration 65

7.1 Conventional CMFD . . . 66
7.2 Spatial Domain Decomposed CMFD . . . 68
7.3 Artificially Diffusive CMFD . . . 69
7.4 Optimal CMFD . . . 70
7.5 3D MOC CMFD Acceleration . . . 71
7.6 Subplane CMFD . . . 72
7.6.1 Homogenization . . . 72
7.6.2 Coupling Coefficients . . . 73
7.6.3 Projection . . . 73
7.7 Solving the CMFD Eigenvalue Problem . . . 73
7.7.1 Power Iteration . . . 73
7.7.2 Wielandt Shift . . . 74
7.7.2.1 Traditional Wielandt Shifts . . . 74
7.7.2.2 Space-Dependent Wielandt Shift . . . 75
7.7.2.3 Impact on Linear Solvers . . . 77
7.7.3 Convergence Criterion for CMFD Power Iterations . . . 77
7.8 Red-Black Successive Over-Relaxation CMFD . . . 78
7.8.1 Red-Black Gauss-Seidel . . . 78
7.8.2 Successive Over-Relaxation (SOR) . . . 79
7.9 Summary . . . 81

8 General Cross Section Data Calculation **82**

8.1 The Multigroup Cross Section Library . . . 82
8.2 Macroscopic Cross Section . . . 84

8.3 Transient Data Calculation . . . 85
8.4 Data for Thermal Power Calculation . . . 86

9 Cross Section Resonance Self-Shielding 88

9.1 The Resonance Self-Shielding Treatment . . . 89
9.2 The Subgroup Method, ESSM, and ESSM-X . . . 90
9.3 Resonance Interference and Resonance Scattering . . . 93
9.4 Lumped Parameter MOC for Subgroup . . . 93
9.5 Multigroup and 1-Group Subgroup . . . 98

10 Nuclide Depletion and Decay 101

10.1 Nuclide Transmutation Equation and its Solution . . . 101
10.1.1 Nuclide Transmutation Equation . . . 101
10.1.2 Computing the Matrix Exponential . . . 103
10.1.3 MPACT Point Depletion Solution Algorithm . . . 106
10.2 Coupling of the Neutron Transport and Nuclide Transmutation Equations . . . 109
10.2.1 Predictor-Corrector . . . 109
10.2.2 Substep Method . . . 110
10.2.3 Depletion Time-stepping algorithm . . . 111

11 Transient Methods 113

11.1 Transient Methods within the 2D/1D Framework . . . 113
11.1.1 3D Time-dependent Neutron Transport Equations . . . 113
11.1.2 Precursor Integration and Formulation of the Transient Fixed Source Problem . . . 114
11.1.3 2D MOC Solution of the Transient Fixed Source Problem . . . 116

11.1.4 Transient 1D Nodal Method . . . 117
11.1.5 CMFD Transient Fixed Source Problem . . . 118
11.1.5.1 MGS CMFD Formulation . . . 119
11.1.5.2 MGM CMFD Formulation . . . 120
11.1.6 Iteration Strategy . . . 121
11.2 Transient Multilevel (TML) Method . . . 121
11.2.1 CMFD Adjoint Flux . . . 122
11.2.2 Point Kinetics Equations . . . 123
11.2.3 Transient Multilevel Method . . . 124
11.2.3.1 3D Transport and CMFD Coupling . . . 124
11.2.3.2 3D CMFD and EPKE Coupling . . . 126
11.2.4 Iteration Scheme with TH Feedback . . . 127
11.2.5 First Level TH Coupling for 3D Transport/3D CMFD . . . 127
11.2.5.1 Second Level TH Coupling for 3D CMFD/EPKE . . . 130
11.2.5.2 Overall Flow Chart for TML with TH Feedback . . . 130

12 Simplified Thermal Hydraulic Model 131

12.1 Introduction . . . 131
12.2 Fluid Flow Model . . . 131
12.3 Fuel Temperature Models . . . 132
12.3.1 1D Heat Conduction . . . 133
12.3.1.1 Heat Transfer from Fluid to Clad . . . 133
12.3.1.2 Gap Conductance . . . 133
12.3.1.3 Radial Heat Transfer Equation . . . 134

12.3.1.4 Thermal Properties 135

12.3.2 Fuel Temperature Tables 135

12.4 Discussion 135

13 Miscellaneous Topics **137**

13.1 Module-Based Decomposition Strategy 137

13.2 Rotational Symmetry Boundary Treatment 139

1. Introduction

MPACT is a three-dimensional (3D) full-core neutron transport code capable of calculating subpin power distributions. Calculations are based on the Boltzmann transport equation for the neutron fluxes, for problems in which the detailed geometrical configuration of fuel components such as the pellet and cladding are explicitly retained. The cross section data needed for the neutron transport calculation are obtained directly from a multigroup cross section library, which has traditionally been used by lattice physics codes to generate few-group homogenized cross sections for nodal core simulators. Hence, MPACT assumes neither *a priori* homogenization nor group condensation for the full core spatial solution.

The 3D MPACT transport solution can be obtained using the method of characteristics (MOC) [1], [43], which employs discrete ray tracing within each fuel pin. However, for practical reactor applications, the direct application of MOC to 3D core configurations requires an excessive amount of memory and computing time, due to the very large number of rays.

For practical 3D full-core calculations, MPACT commonly uses an approximate "2D/1D" method that treats the radial (x and y) variables differently from the axial (z) variable. In particular, the radial dependence of the solution is calculated using transport theory, and the axial dependence is calculated using diffusion or P_3 theory. The 2D/1D method requires the core to be divided into a vertical stack of axial slices with a thickness of $\Delta_z \approx 5$–10 cm. Each axial slice is divided radially into coarse spatial cells with boundaries that usually constitute the pin cell boundaries, for which $\Delta_x = \Delta_y \approx 1.5$ cm. Then, each coarse radial cell (pin cell) is divided into 50–100 fine radial cells, which resolve the angular flux in the fuel, cladding, and moderator regions.

The 2D radial transport calculations are formulated using the 2D MOC method on the fine radial spatial cells within each slice. The 1D axial diffusion or P_3 calculations are formulated using the nodal expansion method (NEM) on the coarse radial spatial cells and extend axially through every slice. The fine- and coarse-grid equations are coupled through transverse leakages, and the equations are structured so that upon convergence, the fine-grid scalar fluxes – integrated over a coarse spatial cell – exactly yield the coarse-grid scalar fluxes. The approximate 1D axial calculation is used because most spatial heterogeneity in the reactor core occurs in the radial direction instead of the axial direction. Therefore, the neutron flux itself should have strong spatial variations in the radial direction, but only weak spatial variations axially. Alternatively, a full 3D MOC solution can be performed, if the problem warrants it and the computational resources are available.

To accelerate the iterative convergence of the whole-core transport calculation, MPACT uses the coarse mesh finite difference (CMFD) acceleration method. The basic spatial mesh in the CMFD formulation is the 3D coarse spatial grid upon which the group constants are dynamically homogenized using the estimates of the fine-grid scalar fluxes. The concept of dynamically homogenizing the group constants for each pin cell is the basis of the effectiveness of the CMFD formulation. The intra-cell flux distribution determined from the MOC calculation is used to generate the homogenized cell constants, while the MOC cell surface-averaged currents are used to determine the radial nodal coupling coefficients. An *equivalence formalism* guarantees that, on convergence, the scalar flux obtained with CMFD is identical to the scalar flux obtained by the MOC calculation, integrated over a coarse cell. In addition to the acceleration aspect of the CMFD formulation, CMFD provides the framework for the 3D calculation in which the global 3D neutron balance is maintained through the use of the MOC-generated cell constants, radial coupling coefficients, and the NEM-generated axial coupling coefficients.

An outline of the remainder of this manual follows:

- In Chapter 2, the steady-state linear Boltzmann (neutron transport) equation is introduced, along with its standard multigroup approximation (in energy) and discrete ordinates approximation (in angle).
- Chapter 3 briefly outlines how MPACT uses the algorithms described in the subsequent chapters of the manual to solve eigenvalue, depletion, and transient problems.
- Chapter 4 begins with the history of the 2D/1D method in MPACT. Then the 2D/1D method is described, starting from the underlying Boltzmann transport equation. The unique feature of this method is that the 2D transport equation describes neutron transport in the radial directions x and y, and the diffusion or P_3 equations approximately describe neutron transport in the axial direction z. The chapter concludes with an outline of the basic iteration strategy used by MPACT to solve the 2D/1D equations.
- Chapter 5 presents the method of characteristics (MOC) methodology for discretizing and solving the 2D linear Boltzmann (neutron transport) equation.
- Chapter 6 describes the methods used in MPACT to spatially discretize the 1D axial (typically diffusion or P_3) equations.
- Chapter 7 describes the 3D coarse mesh finite difference (CMFD) method used to accelerate the iterative convergence of the 2D/1D solver.
- Chapter 8 presents the general multigroup cross section calculation in MPACT.
- Chapter 9 discusses MPACT's treatment of energy resonance and self-shielding.
- Chapter 10 presents the MPACT methodology for solving depletion problems.
- Chapter 11 describes methods used in MPACT for solving transient problems.

- Chapter 12 outlines MPACT's simplified thermal hydraulics (TH) model.
- Chapter 13 concludes the manual with brief discussions of some miscellaneous topics.

2. The Linear Boltzmann Transport Equation

2.1 The Boltzmann Equation

This chapter provides a brief introduction to the 3D steady-state Linear Boltzmann Transport Equation, the solution of which determines the neutron (and photon) fluxes in nuclear reactor cores and shields. A detailed derivation of this equation is beyond the scope of this manual; the equation is provided here, and some of its properties are described. For a careful derivation, see Prinja and Larsen (2010) [73].

To characterize a general 3D steady-state neutron transport process, *six* independent variables are required: three components of the 3D spatial position vector:

$$\boldsymbol{x} = (x, y, z), \tag{2.1a}$$

two angular variables (the polar cosine μ and the azimuthal angle ω) to specify the 3D unit vector $\boldsymbol{\Omega}$ denoting the direction of flight:

$$\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z) = (\sqrt{1-\mu^2}\cos\omega, \sqrt{1-\mu^2}\sin\omega, \mu), \tag{2.1b}$$

and the kinetic energy E. These variables ultimately make it possible to specify the population of neutrons (i) at an arbitrary point $\boldsymbol{x}$ in the system, (ii) traveling in an arbitrary direction of flight $\boldsymbol{\Omega}$, and (iii) with an arbitrary energy E.

If the spatial variables x, y, and z are displaced by *incremental* (very small) amounts dx, dy, and dz, the spatial vector $\boldsymbol{x}$ will sweep out an incremental hexahedral volume $dV = dxdydz$. Similarly, if the angular variables μ and ω are displaced by incremental amounts $d\mu$ and $d\omega$, then the unit vector $\boldsymbol{\Omega}$ will sweep out a dimensionless incremental rectangular *element of area* or *solid angle* $d\Omega = d\mu d\omega$ on the unit sphere.

The *macroscopic cross sections* are defined by the following:

$$\Sigma_t(\boldsymbol{x},E)ds = \text{the incremental probability that a neutron at point } \boldsymbol{x}\text{, with energy } E,$$
$$\text{traveling an incremental distance } ds\text{, will experience a collision with a nucleus,} \tag{2.2a}$$
$$\Sigma_s(\boldsymbol{x},E)ds = \text{the incremental probability that a neutron at point } \boldsymbol{x}\text{, with energy } E,$$
$$\text{traveling an incremental distance } ds\text{, will scatter off a nucleus,} \tag{2.2b}$$
$$\Sigma_\gamma(\boldsymbol{x},E)ds = \text{the incremental probability that a neutron at point } \boldsymbol{x}\text{, with energy } E,$$
$$\text{traveling an incremental distance } ds\text{, will be captured by a nucleus,} \tag{2.2c}$$
$$\Sigma_f(\boldsymbol{x},E)ds = \text{the incremental probability that a neutron at point } \boldsymbol{x}\text{, with energy } E,$$
$$\text{traveling an incremental distance } ds\text{, will initiate a fission event with a nucleus.} \tag{2.2d}$$

These cross sections satisfy:

$$\Sigma_t(\boldsymbol{x},E) = \Sigma_s(\boldsymbol{x},E) + \Sigma_\gamma(\boldsymbol{x},E) + \Sigma_f(\boldsymbol{x},E). \tag{2.3}$$

The *macroscopic differential scattering* cross section is defined by:

$$\Sigma_s(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},E'\rightarrow E)dsd\Omega dE = \text{the incremental probability that a neutron at } (\boldsymbol{x},\boldsymbol{\Omega}',E'),$$
$$\text{traveling an incremental distance } ds\text{, will scatter into}$$
$$d\Omega \text{ about } \Omega \text{ and } dE \text{ about } E. \tag{2.4}$$

This cross section satisfies:

$$\int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},E'\rightarrow E)d\Omega dE = \Sigma_s(\boldsymbol{x},E'). \tag{2.5}$$

Also, the *fission spectrum* is defined by

$$\chi(\boldsymbol{x},E)dE = \text{the incremental probability that a fission neutron, emitted}$$
$$\text{at } \boldsymbol{x}\text{, will have energy between } E \text{ and } E+dE. \tag{2.6}$$

The fission spectrum satisfies:

$$\int_0^\infty \chi(\boldsymbol{x},E)dE = 1. \tag{2.7}$$

It is assumed that in all problems, the material cross sections and fission spectrum are specified at each spatial point $\boldsymbol{x}$ in the physical system of interest.

Next, we consider all neutrons that (i) are geometrically located in a volume increment dV about a point $\boldsymbol{x}$, (ii) travel within a solid angle $d\Omega$ about the direction $\boldsymbol{\Omega}$, and (iii) have energies between E and $E+dE$. The *angular neutron density* $N(\boldsymbol{x},\boldsymbol{\Omega},E,t)$, a function of six independent variables, is defined by:

$$N(\boldsymbol{x},\boldsymbol{\Omega},E)dVd\Omega dE = \text{the incremental number of neutrons in}$$
$$dVd\Omega dE \text{ about the phase space point } (\boldsymbol{x},\boldsymbol{\Omega},E). \tag{2.8}$$

The *angular flux* $\psi(\boldsymbol{x},\boldsymbol{\Omega},E)$ is now defined by:

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E) = vN(\boldsymbol{x},\boldsymbol{\Omega},E), \tag{2.9}$$

where $v = \sqrt{2E/m}$ = neutron speed. ψ has the following physical interpretation:

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E)dVd\Omega dE = \text{the incremental rate at which path length is generated by neutrons in } dVd\Omega dE \text{ about } (\boldsymbol{x},\boldsymbol{\Omega},E)\ . \tag{2.10}$$

(The rates at which neutrons interact with matter are directly proportional to ψ.)

In the following linear Boltzmann equation for $\psi(\boldsymbol{x},\boldsymbol{\Omega},E)$, we assume that a convex spatial domain V is given, and that all the material cross sections are known for all points $\boldsymbol{x} \in V$:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\boldsymbol{x},\boldsymbol{\Omega},E) + \Sigma_t(\boldsymbol{x},E)\psi(\boldsymbol{x},\boldsymbol{\Omega},E) = &\int_0^\infty\int_{4\pi}\Sigma_s(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},E'\to E)\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')d\Omega' dE' \\ &+ \frac{\chi(\boldsymbol{x},E)}{4\pi k_{\text{eff}}}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\boldsymbol{x},E')\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')d\Omega' dE', \\ &\boldsymbol{x}\in V, \boldsymbol{\Omega}\in 4\pi, 0<E<\infty. \end{aligned} \tag{2.11a}$$

Eq. (2.11a) must be solved subject to the "vacuum" boundary condition:

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E) = 0, \quad \boldsymbol{x}\in\partial V, \boldsymbol{\Omega}\cdot\boldsymbol{n} < 0, 0<E<\infty. \tag{2.11b}$$

Eqs. (2.11a and b) constitute an *eigenvalue problem* for the *eigenfunction* $\psi(\boldsymbol{x},\boldsymbol{\Omega},E)$ and the *criticality eigenvalue* k_{eff}. The smallest positive eigenvalue k_{eff} is sought so that Eqs. (2.11a and b) have a solution $\psi(\boldsymbol{x},\boldsymbol{\Omega},E) \geq 0$. (Problems of this type are most commonly solved by MPACT.)

For 3D neutron transport equations to be computationally solved by a deterministic method, it is necessary to discretize each of the six independent variables, thereby turning Eqs. (2.11a and b) into a (typically very large) algebraic system of equations. Discretizing the energy, direction, and spatial variables presents different types of problems. The energy variable is discussed first.

2.2 The Multigroup Approximation

The *multigroup* approximation to the neutron transport equation is almost universally used to discretize the continuous energy variable E. The structure of the resulting multigroup transport equations is closely related to that of the original transport equation, the difference being that the energy variable is discrete rather than continuous. (Integrals over E are replaced by sums over *energy groups*.) Several important identities of the original continuous-energy scattering operator are preserved in the multigroup approximation. Here the multigroup transport equations are derived and some of their properties are discussed.

The multigroup approximation requires that a finite number G of energy bins or *groups* be chosen:

$$E_{min} = E_G < E_{G-1} < \cdots < E_g < E_{g-1} < \cdots < E_2 < E_1 = E_{max},$$

with E_{min} sufficiently small that neutrons with energies less than E_{min} are negligible, and with E_{max} sufficiently large such that neutrons with energies greater than E_{max} are negligible. The energy range $E_g \leq E < E_{g-1}$ is the g^{th} *energy group*. It is customary to order the energy groups with the group index g increasing as the energies decrease. Then the slowing-down of fast fission neutrons occurs through energy groups with increasing indices.

For each $1 \leq g \leq G$, we define:

$$\begin{aligned} \psi_g(\boldsymbol{x},\boldsymbol{\Omega}) &= \int_{E_g}^{E_{g-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE \\ &= \text{ angular flux for group } g \text{ ,} \textit{and} \qquad (2.12a) \\ \chi_g(\boldsymbol{x}) &= \int_{E_g}^{E_{g-1}} \chi(\boldsymbol{x},E)dE \\ &= \text{ multigroup fission spectrum for group } g \text{ .} \qquad (2.12b) \end{aligned}$$

By the preceding definitions and Eq. (2.7), the multigroup fission spectrum automatically satisfies

$$\sum_{g=1}^{G} \chi_g(\boldsymbol{x}) = \sum_{g=1}^{G} \int_{E_g}^{E_{g-1}} \chi(\boldsymbol{x},E)dE = \int_{E_{min}}^{E_{max}} \chi(\boldsymbol{x},E)dE = 1. \qquad (2.13)$$

To proceed, we integrate Eq. (2.11a) over the g^{th} energy group, obtaining:

$$\begin{aligned} &\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) + \int_{E_g}^{E_{g-1}} \Sigma_t(\boldsymbol{x},E)\psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE \\ &\quad = \sum_{g'=1}^{G} \int_{E_g}^{E_{g-1}} \int_{E'_g}^{E_{g'-1}} \int_{4\pi} \Sigma_s(\boldsymbol{x},E'\rightarrow E,\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')d\Omega' dE' dE \\ &\qquad + \frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \int_{E'_g}^{E_{g'-1}} \int_{4\pi} \nu\Sigma_f(\boldsymbol{x},E')\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')d\Omega' dE', \end{aligned}$$

or:

$$\begin{aligned} &\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) + \left[\frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\boldsymbol{x},E)\psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE}{\int_{E_g}^{E_{g-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE}\right]\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) \\ &\quad = \sum_{g'=1}^{G} \int_{4\pi} \left[\frac{\int_{E_g}^{E_{g-1}}\int_{E_{g'}}^{E_{g'-1}} \Sigma_s(\boldsymbol{x},E'\rightarrow E,\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'dE}{\int_{E_{g'}}^{E_{g'-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}\right]\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &\qquad + \frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \int_{4\pi} \left[\frac{\int_{E_{g'}}^{E_{g'-1}} \nu\Sigma_f(\boldsymbol{x},E')\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}{\int_{E_{g'}}^{E_{g'-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}\right]\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega', \end{aligned}$$

or:

$$\begin{aligned} \boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) + \hat{\Sigma}_{t,g}(\boldsymbol{x},\boldsymbol{\Omega})\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) &= \sum_{g'=1}^{G} \int_{4\pi} \hat{\Sigma}_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}',\boldsymbol{\Omega})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &\quad + \frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \int_{4\pi} \hat{\nu}\hat{\Sigma}_{f,g}(\boldsymbol{x},\boldsymbol{\Omega}')\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega', \qquad (2.14) \end{aligned}$$

where:

$$\hat{\Sigma}_{t,g}(\boldsymbol{x},\boldsymbol{\Omega}) = \left[\frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\boldsymbol{x},E)\psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE}{\int_{E_g}^{E_{g-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega},E)dE}\right], \tag{2.15a}$$

$$\hat{\Sigma}_{s,g'\to g}(\boldsymbol{x},\boldsymbol{\Omega}',\boldsymbol{\Omega}) = \left[\frac{\int_{E_g}^{E_{g-1}}\int_{E_{g'}}^{E_{g'-1}} \Sigma_s(\boldsymbol{x},E'\to E,\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'dE}{\int_{E_{g'}}^{E_{g'-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}\right], \tag{2.15b}$$

$$\hat{\nu\Sigma}_{f,g}(\boldsymbol{x},\boldsymbol{\Omega}') = \left[\frac{\int_{E_{g'}}^{E_{g'-1}} \nu\Sigma_f(\boldsymbol{x},E')\psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}{\int_{E_{g'}}^{E_{g'-1}} \psi(\boldsymbol{x},\boldsymbol{\Omega}',E')dE'}\right]. \tag{2.15c}$$

Exact boundary conditions can be obtained by integrating Eq. (2.11b) over the energy groups:

$$\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) = 0, \quad \boldsymbol{x}\in\partial V, \boldsymbol{\Omega}\cdot\boldsymbol{n} < 0. \tag{2.16}$$

Eqs. (2.14) through (2.16) are an exact system of equations for the group fluxes. If the hatted coefficients in Eqs. (2.15) were known, then Eqs. (2.14) and (2.16) would, in the absence of spatial and angular discretizations, yield the exact group fluxes. However, by Eqs. (2.15), the hatted coefficients depend on the solution of the continuous-energy problem and are not known.

In the multigroup approximation, an approximation for ψ is specified and introduced into the right sides of Eqs. (2.15). The resulting approximate *multigroup cross sections* are then used in Eq. (2.14).

Specifically, in each of the bracketed terms in Eq. (2.15), we introduce the approximation:

$$\psi(\boldsymbol{x},\boldsymbol{\Omega},E) \approx \Psi(\boldsymbol{x},E)f(\boldsymbol{x},\boldsymbol{\Omega}), \tag{2.17}$$

where $\Psi(\boldsymbol{x},E)$ is a specified *neutron spectrum*. The function $f(\boldsymbol{x},\boldsymbol{\Omega})$ cancels out of each numerator and denominator, and Eqs. (2.15) yield the *multigroup cross sections*:

$$\Sigma_{t,g}(\boldsymbol{x}) = \left[\frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\boldsymbol{x},E)\Psi(\boldsymbol{x},E)dE}{\int_{E_g}^{E_{g-1}} \Psi(\boldsymbol{x},E)dE}\right], \tag{2.18a}$$

$$\Sigma_{s,g'\to g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega}) = \left[\frac{\int_{E_g}^{E_{g-1}}\int_{E_{g'}}^{E_{g'-1}} \Sigma_s(\boldsymbol{x},E'\to E,\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\Psi(\boldsymbol{x},E')dE'dE}{\int_{E_{g'}}^{E_{g'-1}} \Psi(\boldsymbol{x},E')dE'}\right], \tag{2.18b}$$

$$\nu\Sigma_{f,g}(\boldsymbol{x}) = \left[\frac{\int_{E_{g'}}^{E_{g'-1}} \nu\Sigma_f(\boldsymbol{x},E')\Psi(\boldsymbol{x},E')dE'}{\int_{E_{g'}}^{E_{g'-1}} \Psi(\boldsymbol{x},E)dE'}\right]. \tag{2.18c}$$

Using these approximate multigroup cross sections in place of the hatted exact cross sections in Eq. (2.14), we obtain

the *multigroup transport equations*:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi_g(\boldsymbol{x},\boldsymbol{\Omega})+\Sigma_{t,g}(\boldsymbol{x})\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) &= \sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &+\frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\int_{4\pi}\nu\Sigma_{f,g}(\boldsymbol{x})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega', \\ &\boldsymbol{x}\in V,\boldsymbol{\Omega}\in 4\pi, 1\leq g\leq G. \end{aligned} \tag{2.19}$$

The multigroup fluxes $\psi_g(\boldsymbol{x},\boldsymbol{\Omega})$ are obtained by solving Eqs. (2.19) with the *multigroup boundary conditions* (2.16).

To complete the multigroup approximation, the multigroup capture and fission cross sections are defined analogous to Eqs. (2.18):

$$\Sigma_{\gamma,g}(\boldsymbol{x}) = \left[\frac{\int_{E_g}^{E_{g-1}}\Sigma_\gamma(\boldsymbol{x},E)\Psi(\boldsymbol{x},E)dE}{\int_{E_g}^{E_{g-1}}\Psi(\boldsymbol{x},E)dE}\right], \tag{2.20a}$$

$$\Sigma_{f,g}(\boldsymbol{x}) = \left[\frac{\int_{E_g}^{E_{g-1}}\Sigma_f(\boldsymbol{x},E)\Psi(\boldsymbol{x},E)dE}{\int_{E_g}^{E_{g-1}}\Psi(\boldsymbol{x},E)dE}\right]. \tag{2.20b}$$

Then, by Eqs. (2.3), (2.18), and (2.20), the following identities hold for all g and g':

$$\begin{aligned}\Sigma_{t,g}(\boldsymbol{x}) &= \frac{\int_{E_g}^{E_{g-1}}\left[\Sigma_s(\boldsymbol{x},E)+\Sigma_\gamma(\boldsymbol{x},E)+\Sigma_f(\boldsymbol{x},E)\right]\Psi(\boldsymbol{x},E)dE}{\int_{E_g}^{E_{g-1}}\Psi(\boldsymbol{x},E)dE} \\ &= \Sigma_{s,g}(\boldsymbol{x})+\Sigma_{\gamma,g}(\boldsymbol{x})+\Sigma_{f,g}(\boldsymbol{x}),\end{aligned} \tag{2.21a}$$

and:

$$\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g\rightarrow g'}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})d\Omega' = \cdots = \Sigma_{s,g}(\boldsymbol{x}). \tag{2.21b}$$

Eqs. (2.21) hold for any choice of $\Psi(\boldsymbol{x},E)$; they are the multigroup analog of the continuous-energy identities of Eqs. (2.3) and (2.5).

The key element in this derivation is the choice of the neutron spectrum $\Psi(\boldsymbol{x},E)$ in Eqs. (2.17) and (2.18). The determination of this spectrum requires experience and careful understanding of the problem to be solved. A detailed description of this process cannot be given here.

The structure of the multigroup transport equations is similar to that of the continuous-energy transport equation, the key difference being that in the multigroup equations, the energy variable is discrete rather than continuous.

2.3 The Discrete Ordinates (S_N) Approximation

For general 2D and 3D problems, the angular variable $\boldsymbol{\Omega}$ is widely discretized using the *discrete ordinates* or S_N approximation. For Cartesian geometries, this approximation consists of a system of angularly discrete equations with

a mathematical structure that is similar to the original transport equation. The principal difference is that neutrons travel only in a specified finite number of directions on the unit sphere rather than in all directions.

If N denotes the order of an angular quadrature set, then we let M_N denote the number of discrete angles $\mathbf{\Omega}_m$ in the quadrature set, with $1 \leq m \leq M_N$. Associated with each $\mathbf{\Omega}_m$ is an angular weight w_m, which represents the area (on the unit sphere) of the cone whose central axis is $\mathbf{\Omega}_m$. A function $\psi(\mathbf{\Omega})$ is now represented as a discrete quantity:

$$\psi(\mathbf{\Omega}_m) = \psi_m, 1 \leq m \leq M_N,$$

and angular integrals of $\psi(\mathbf{\Omega})$ are represented as quadrature sums:

$$\int_{4\pi} \psi(\mathbf{\Omega}) d\Omega \approx \sum_{m=1}^{M_N} \psi(\mathbf{\Omega}_m) w_m = \sum_{m=1}^{M_N} \psi_m w_m.$$

Historically, the determination of quadrature sets of discrete angles and weights for neutron transport problems has received considerable attention. The general idea has been to choose quadrature sets such that (i) the directions and weights are distributed as symmetrically as possible on the unit sphere, (ii) the angular weights are positive, (iii) the approximation of an integral by a sum is exact when ψ is a low-order spherical harmonic function, and (iv) the total number M_N of directions and weights is as small as possible.

The general details of the procedures by which angular quadrature sets are chosen in practice is beyond the scope of this manual. However, some information about product quadrature sets is included because they play an important role in the 2D/1D method.

In product quadrature sets, the angular variables μ and ω are individually discretized, as shown in Eq. (2.1b): μ on the interval $-1 \leq \mu \leq 1$ (typically using a Gauss-Legendre quadrature set), and ω on the interval $0 \leq \omega < 2\pi$ (typically using a Chebychev quadrature set). Thus, there is a 1D quadrature set for μ of order N_a ($a \sim$ "axial"):

$$\{(\mu_n, u_n) | 1 \leq n \leq N_a\}, \tag{2.22a}$$

where u_n are the angular weights and a 1D quadrature set for ω of order N_r ($r \sim$ "radial"):

$$\{(\omega_m, v_m) | 1 \leq m \leq N_r\}, \tag{2.22b}$$

where v_m are the angular weights. Therefore, Eq. (2.1b) gives the following 3D product quadrature set:

$$\{(\mathbf{\Omega}_{n,m}, w_{n,m}) | 1 \leq n \leq N_a, 1 \leq m \leq N_r\}, \tag{2.22c}$$

where

$$\mathbf{\Omega}_{n,m} = \left(\sqrt{1-\mu_n^2} \cos \omega_m, \sqrt{1-\mu_n^2} \sin \omega_m, \mu_n \right), \tag{2.22d}$$

and

$$w_{n,m} = u_n v_m. \tag{2.22e}$$

Product quadrature sets treat the polar direction differently from the radial directions; this feature conforms well with the underlying methodology of the 2D/1D method.

Now, applying an S_N approximation to the multigroup transport equations (2.19) and (2.16), we obtain:

$$\begin{aligned}\boldsymbol{\Omega}_m \cdot \boldsymbol{\nabla} \psi_{m,g}(\boldsymbol{x}) + \Sigma_{t,g}(\boldsymbol{x})\psi_{m,g}(\boldsymbol{x}) = &\sum_{g'=1}^{G}\sum_{m'=1}^{M_N} \Sigma_{s,g'\to g}(\boldsymbol{x}, \boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{m',g'}(\boldsymbol{x})w_{m'} \\ &+ \frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N} \nu\Sigma_{f,g'}(\boldsymbol{x})\psi_{m',g'}(\boldsymbol{x})w_{m'}, \\ &\boldsymbol{x}\in V, 1\le m\le M_N, 1\le g\le G, \end{aligned} \tag{2.23a}$$

$$\psi_{m,g}(\boldsymbol{x}) = 0, \quad \boldsymbol{x}\in\partial V, \boldsymbol{\Omega}_m\cdot\boldsymbol{n} < 0. \tag{2.23b}$$

Again, these *multigroup discrete ordinates* equations retain the basic structure of the original Boltzmann transport equation. The main difference between Eqs. (2.23) and the original transport Eq. (2.11a) is that in Eqs. (2.23), the energies at which neutrons can exist are discrete (not infinite), and the directions in which neutrons can travel are also discrete (not infinite).

The 2D/1D method approximates the 3D multigroup transport Eq. (2.19) in a way that leads to (i) a 2D "radial" transport equation, coupled to (ii) a 1D "axial" transport equation. The multigroup and discrete ordinates approximations described above can be directly applied to both of these equations. (However, the 1D axial transport equation is typically approximated by a 1D low-order P_N equation; this reduces the computational cost and has only a minor effect on accuracy.) After a brief summary (in Chapter 3) of the algorithms used in MPACT, the 2D/1D method is discussed (in Chapter 4).

3. Summary of Algorithms

This chapter summarizes the algorithms used in MPACT for solving three different types of problems. These algorithms employ different capabilities in MPACT that are described in the subsequent chapters of this manual. For readers who are already knowledgeable about nuclear reactor physics, this chapter provides a brief introduction to the subsequent chapters of the manual, along with a description of how the capabilities in these chapters work together in MPACT. For readers who are not yet knowledgeable, this chapter can be skipped until a later time.

- **Eigenvalue Problems:** Algorithm 3.1 describes the process by which MPACT solves the steady-state eigenvalue problem for k_{eff} and ϕ. In Algorithm 3.1, the user can choose between 2D/1D and 2D, whether or not to use CMFD, Gauss-Siedel, or Jacobi iteration in energy, and whether or not to include thermal feedback. However, it should be noted that the 2D/1D problem cannot be run without CMFD. The primary details for this algorithm can be found in Chapters 4, 5, 7, and 12.
- **Depletion Calculations:** Algorithm 3.2 describes MPACT's depletion calculation. A predictor-corrector algorithm is used to produce improved estimates of the nuclide compositions at each time step. More details are available in Chapter 10.
- **Transient Calculations:** Algorithm 3.3 describes the process by which MPACT performs a transient calculation. A multi-level solver (point kinetics, CMFD, MOC) efficiently solves the fixed-source transient problem while minimizing the number of transport sweeps required. More details are available in Chapter 11, as well as a graphical flow chart of the iteration (see Figure 11.7).

Algorithm 3.1: Algorithm for solving an eigenvalue problem in MPACT.

Update macroscopic cross sections from resonance treatment (subgroup) calculations (Chapter 9).
Guess initial source ($\phi^{(0)}$) and eigenvalue ($k_{\text{eff}}^{(0)}$).
while $\phi^{(n)}$ and $k_{\text{eff}}^{(n)}$ not converged **do**
  if considering feedback, **then**
    Using the current estimate of solution, evaluate feedback equations (Chapter 12).
    if update resonance parameters, **then**
      perform subgroup calculations (Chapter 9).
    end if
    Update the macroscopic cross sections.
  end if
  if using CMFD, **then**
    Generate coarse-grid quantities from the fine-grid fluxes and leakages.
    Solve the CMFD eigenvalue problem iteratively to obtain an updated eigenvalue ($k_{\text{eff}}^{(n+1)}$) and coarse-grid scalar flux. (Chapter 7 for more information).
    Update the fine-grid scalar flux/fission source ($\phi^{(n+1/2)}$).
  end if
  if 2D/1D, **then**
    Obtain the radial leakages from the CMFD solution.
    Solve the 1D axial diffusion equation, Eq. (4.11b), to update the scalar fluxes.
  end if
  Update the axial leakage from CMFD.
  if Gauss-Siedel, **then**
    Obtain $\phi^{(n+1)}$ by performing transport sweep(s) on the 2D radial transport equations, Eq. (4.15), as described in Algorithm 5.3.
  else if Jacobi, **then**
    Obtain $\phi^{(n+1)}$ by performing transport sweep(s) on the 2D radial transport equations, Eq. (4.15), as described in Algorithm 5.4.
  end if
end while

Algorithm 3.2: Depletion algorithm in MPACT, also depicted in Fig. 10.5 of Chapter 10.

Set core power and nuclide compositions for t_0.
for t_i in $t_0, \ldots, t_{final}$ **do**
  for predictor substeps $j = 1, \ldots, J$ **do**
    Perform depletion calculation for all nuclides. (See Fig. 10.1 for details.)
  end for
  Use Algorithm 3.1 to solve steady-state transport problem with updated material compositions.
  for corrector substeps $k = 1, \ldots, K$ **do**
    Perform depletion calculation for all nuclides. (See Fig. 10.1 for details.)
  end for
  Average compositions from predictor and corrector steps. These are the compositions for t_{i+1}.
  if updating resonance parameters, **then**
    Perform subgroup calculation. (See Chapter 9.)
  end if
  Use Algorithm 3.1 to solve steady-state transport problems with updated material compositions.
  Set core power and nuclide compositions for t_{i+1}.
end for

Algorithm 3.3: Algorithm for solving transient problems in MPACT. See Chapter 11 for more details.

Perform initial steady-state forward transport and thermal hydraulics (TH) calculation (obtain converged neutronics/TH steady-state solution)
Perform adjoint CMFD calculation, Eq. (11.33).
Begin transient calculation:
for t_i in $t_{0,MOC}, \ldots, t_{final,MOC}$ **do**
  Perform predictor MOC transient fixed source problem (TFSP) solve (one coarse step of Alg. 5.1).
  for $t_{i,j}$ in $t_{i,1}, \ldots, t_{i,M}$ **do**
    Linearly interpolate CMFD coefficients to CMFD time step j (out of M intermediate steps per MOC step).
    Perform predictor CMFD solve (forward and adjoint), Eqs. (11.32) and (11.33).
    Calculate exact point kinetic equation (EPKE) parameters.
    for $t_{i,j,k}$ in $t_{i,j,1}, \ldots, t_{i,j,N}$ **do**
      Linearly interpolate EPKE parameters to EPKE time step k (out of N fine steps per CMFD step)
      Perform EPKE solve, Eq. (11.36) and Eq. (11.37).
    end for
    Correct CMFD solution with updated information from finer EPKE solve, Eq. (11.55).
  end for
  Correct transport solution with updated information from finer CMFD solve, Eq. (11.50).
  Update pin powers and pass to TH solver
  Perform TH solve
  Update cross sections based on updated densities and temperatures from TH
end for

4. The 2D/1D Method

4.1 Historical Overview

In the reactor physics community, the term *2D/1D* denotes computational methods for solving 3D neutron transport problems in which the *2D radial* (x and y) and *1D axial* (z) derivative terms are discretized differently. The motivation for this is that in light water reactors (LWRs), the geometry of the core is complicated in the radial (x and y) directions, but relatively simple in the axial (z) direction. Thus, the spatial behavior of the neutron flux should be complicated in the radial directions, but simpler in the axial direction. A 3D spatial discretization scheme that makes proper use of the relatively simple axial behavior of the neutron flux could be advantageous.

For realistic 3D neutron transport problems, computational methods that treat the radial and axial variables differently were first proposed and implemented by two groups in Korea during 2002–2007 [15], [11], [17], [16], [18], [48], [59], [13], [63], [10], [12].

- One group (N. Z. Cho, G. S. Lee, C. J. Park, and colleagues) at the Korea Advanced Institute of Science and Technology (KAIST) developed the 2D/1D Fusion method for the CRX code [15], [17], [16], [18], [13], [63]. In this method, the 3D Boltzmann transport equation is solved by discretizing the radial derivative term on a fine 2D radial spatial grid and the axial derivative terms on a coarse radial spatial grid. (Again, a coarse radial grid is usually Cartesian, with one coarse cell consisting of a pin cell. The fine radial grid is not Cartesian; it consists of about 50–100 fine cells per coarse (pin) cell, with boundaries of the fine cells conforming to the circular edges of the fuel and cladding and the outer edges of the coarse cell.) In contrast to the fine and coarse radial spatial grids, there is only one coarse axial grid. Typically, a single coarse 2D radial cell consists of a pin cell, which is about 1.5 cm in diameter. The width of a single axial cell is typically much larger, on the order of 5–10 cm.
- The other group (J. Y. Cho, H. G. Joo, K. S. Kim, and S. Q. Zee and colleagues), at the Korea Atomic Energy Research Institute (KAERI), developed a different "planar MOC solution-based 3D heterogeneous core method" for the DeCART code [11], [48], [10], [12]. The KAERI method also discretized the radial derivative term on the fine spatial grid and the axial derivative term on the coarse radial grid. However, the KAERI method simplified the radial derivative term in a way that (i) is accurate for problems in which the axial leakage can be represented by Fick's Law, and (ii) offers significant computational advantages for parallel- architecture computers. The KAERI 2D/1D method evolved into the 2D/1D method currently implemented in MPACT.

The KAERI method can be motivated by the fact that for 3D problems in which the cross sections depend only on the radial spatial variables (x and y), an exact solution of the transport equation exists in terms of an axial buckling:

$$\psi(x,y,z,\mathbf{\Omega}) = \Psi(x,y,\Omega)e^{iBz}.$$

Since the buckling approximation occurs naturally in diffusion theory, this suggests that for problems in which the transport solution varies weakly in the axial (z) variable (but strongly in the radial (x and y) variables), an approximate 3D transport solution with sufficient accuracy can be obtained from an approximate transport problem – in which transport physics is used to treat the radial variables and diffusion physics is used to treat the axial variable. The KAERI method, implemented in DeCART, accomplished that goal. In an early publication, the KAERI method in DeCART was simply called 2D/1D [48]. The 2D/1D moniker is used for that method and for the other related methods introduced below.

The Deterministic Core Analysis Based on Ray Tracing (DeCART) code was originally developed under an I-NERI project between KAERI and Argonne National Laboratory (ANL). An early version of DeCART was acquired by the University of Michigan (UM), where it was initially used for the US Department of Energy (DOE) Consortium for Advanced Light Water Reactors (CASL) project, which started in 2010. This version had an important deficiency: it failed to converge for small axial cell widths Δ_z. Still, the use of (the UM version of) DeCART demonstrated that for problems in which DeCART converged, it had major computational advantages over other 3D S_N or MOC codes.

Nonetheless, the failure of DeCART to converge for small Δ_z and the lack of a mathematical foundation for the 2D/1D methodology in DeCART were major concerns. For these and other practical reasons, a decision was made in 2012 to develop a new 3D reactor physics code at UM that would employ a more robust 2D/1D methodology. To create this new code, a consistent mathematical foundation for the 2D/1D methodology was developed. The new code was named MPACT; in 2014 it became jointly managed and developed with ORNL.

The remainder of this chapter outlines this basic 2D/1D theory. To simplify the discussion, we consider monoenergetic ($G = 1$) problems with isotropic scattering. However, the extension of the 2D/1D method discussed below to multigroup ($G > 1$) problems with anisotropic scattering is straightforward.

4.2 Preliminaries

We shall now derive the 2D/1D method for the 3D one-group Boltzmann equation with isotropic scattering in a homogeneous medium:

$$\mathbf{\Omega}\cdot\nabla\Psi(\boldsymbol{x},\mathbf{\Omega}) + \Sigma_t\Psi(\boldsymbol{x},\mathbf{\Omega}) = \frac{\Sigma_s}{4\pi}\int_{4\pi}\Psi(\boldsymbol{x},\mathbf{\Omega}')d\Omega' + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}}\int_{4\pi}\Psi(\boldsymbol{x},\mathbf{\Omega}')d\Omega',$$
$$\boldsymbol{x}\in V, \mathbf{\Omega}\in 4\pi. \tag{4.1}$$

The extension of the 2D/1D method to general multigroup, anisotropic scattering problems in heterogeneous media is straightforward. In the subsequent discussion, important steps that must be followed in heterogeneous media are explicitly discussed.

The notation in Eq. (4.1) is the same as in the previous chapters:

$$\begin{aligned} \boldsymbol{x} &= \text{ spatial variable} \\ &= (x,y,z), \\ \boldsymbol{\Omega} &= \text{ direction-of-flight variable} \\ &= (\Omega_x,\Omega_y,\Omega_z) \\ &= (\sqrt{1-\mu^2}\cos\omega, \sqrt{1-\mu^2}\sin\omega, \mu), \end{aligned}$$

where μ is the polar cosine $(-1 \le \mu \le 1)$ and ω is the azimuthal angle $(0 \le \omega < 2\pi)$.

The physical system V is required to be a cylinder, consisting of points

$$\boldsymbol{x} = (x,y,z) \text{ such that: } \begin{cases} (x,y) \in R, \\ 0 \le z \le Z, \end{cases}$$

where R is a 2D region in the (x,y)-plane:

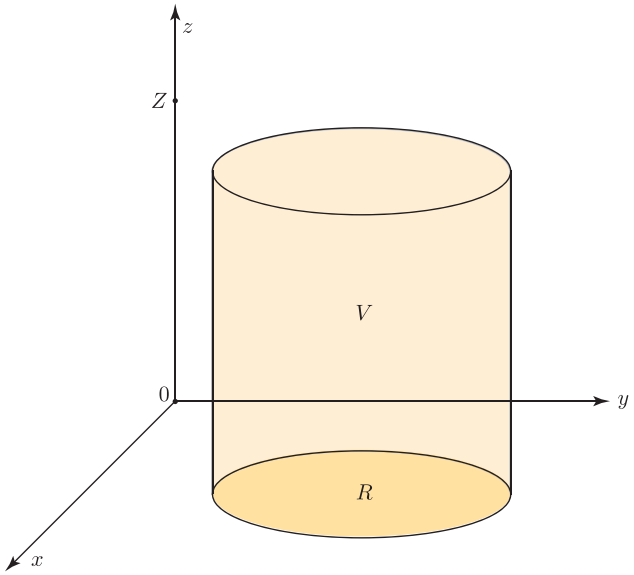


Figure 4.1. The 3D System V.

In this discussion, the boundary conditions on Ψ are assumed to be vacuum:

$$\Psi(\boldsymbol{x},\boldsymbol{\Omega}) = 0, \boldsymbol{x} \in \partial V, \boldsymbol{\Omega}\cdot\boldsymbol{n} < 0, \tag{4.2}$$

where $\boldsymbol{n}$ is the unit outer normal vector on ∂V. However, other boundary conditions (e.g., reflecting) are permitted. The x and y variables as *radial*, and the z variable as *axial*.

4.3 The Basic 2D/1D Equations

The 2D/1D method is based on a system of equations that approximate Eq. (4.1). In this approximate system, neutrons experience the correct 2D transport physics in the radial directions of x and y, but they experience 1D P_1 or P_3 physics in the axial direction z. The 2D/1D method in MPACT is not simple, but it attempts to be as simple as possible, to achieve the necessary accuracy and efficiency.

To begin, a 2D radial transport equation is formulated from Eq. (4.1) with an "isotropized" axial leakage term:

$$\Omega_x \frac{\partial \psi}{\partial x}(\boldsymbol{x},\boldsymbol{\Omega}) + \Omega_y \frac{\partial \psi}{\partial y}(\boldsymbol{x},\boldsymbol{\Omega}) + \Sigma_t \psi(\boldsymbol{x},\boldsymbol{\Omega}) = \frac{\Sigma_s}{4\pi}\int_{4\pi} \psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}}\int_{4\pi} \psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' - \frac{1}{4\pi}\left[\frac{\partial J_z}{\partial z}(\boldsymbol{x})\right]. \tag{4.3}$$

Here the isotropic function $J_z(\boldsymbol{x})$ is an approximation to the axial current:

$$J_z(\boldsymbol{x}) \approx \int_{4\pi} \Omega_z \psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega. \tag{4.4}$$

The exact definition of J_z is given below in Eqs. (4.12) and (4.14). The boundary condition for Eq. (4.3) consists of a vacuum boundary condition on the cylindrical portion of ∂V:

$$\psi(\boldsymbol{x},\boldsymbol{\Omega}) = 0 \text{ for } (x,y) \in \partial R, 0 \le z \le Z. \tag{4.5}$$

Operating on the solution of Eq. (4.3) by $\int_{4\pi}\Omega_x(\cdot)d\Omega$ and $\int_{4\pi}\Omega_y(\cdot)d\Omega$ yields the following radial currents:

$$J_x(\boldsymbol{x}) = \int_{4\pi} \Omega_x \psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega, \tag{4.6a}$$

$$J_y(\boldsymbol{x}) = \int_{4\pi} \Omega_y \psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega. \tag{4.6b}$$

Also, a 1D axial transport equation is formulated from Eq. (4.1), with isotropized radial leakage terms described by J_x and J_y:

$$\mu \frac{\partial \hat{\psi}}{\partial z}(\boldsymbol{x},\boldsymbol{\Omega}) + \Sigma_t \hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}) = \frac{\Sigma_s}{4\pi}\int_{4\pi} \hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}}\int_{4\pi} \hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' - \frac{1}{4\pi}\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x}) + \frac{\partial J_y}{\partial y}(\boldsymbol{x})\right]. \tag{4.7}$$

Integrating this equation over the azimuthal angle ω and defining the azimuthally integrated flux,

$$\hat{\psi}(\boldsymbol{x},\mu) \equiv \int_0^{2\pi} \hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega})d\omega, \tag{4.8}$$

results in

$$\mu\frac{\partial\hat{\psi}}{\partial z}(\boldsymbol{x},\mu)+\Sigma_t\hat{\psi}(\boldsymbol{x},\mu)=\frac{\Sigma_s}{2}\int_{-1}^{1}\hat{\psi}(\boldsymbol{x},\mu')d\mu'+\frac{\nu\Sigma_f}{2k_{\text{eff}}}\int_{-1}^{1}\hat{\psi}(\boldsymbol{x},\mu')d\mu'-\frac{1}{2}\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x})+\frac{\partial J_y}{\partial y}(\boldsymbol{x})\right]. \tag{4.9}$$

The boundary conditions for this equation are vacuum, on the top and bottom of V:

$$\hat{\psi}(\boldsymbol{x},\mu)=0 \text{ for } \begin{cases} z=Z & ,-1\le\mu<0, \\ z=0 & ,0<\mu\le 1. \end{cases} \tag{4.10}$$

In MPACT, the axial transport Eqs. (4.9) and (4.10) are not directly solved. Instead, the simpler 1D P_1 (or P_3) approximation to these equations is formulated and solved, and the radial current J_z is expressed in terms of the approximate P_1 (or P_3) solution. These approximations are defined next.

If we define the axial scalar flux:

$$\hat{\phi}(\boldsymbol{x})\equiv\int_{-1}^{1}\psi(\boldsymbol{x},\mu)d\mu, \tag{4.11a}$$

then the standard P_1 approximation to Eqs. (4.9) and (4.10) is given by the 1D axial diffusion equation:

$$-\frac{\partial}{\partial z}\frac{1}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial z}(\boldsymbol{x})+\Sigma_a\hat{\phi}(\boldsymbol{x})=\frac{\nu\Sigma_f}{k_{\text{eff}}}\hat{\phi}(\boldsymbol{x})-\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x})+\frac{\partial J_y}{\partial y}(\boldsymbol{x})\right], \tag{4.11b}$$

with the boundary conditions:

$$\begin{aligned}\hat{\phi}(x,y,Z)+\frac{2}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial x}(x,y,Z)=0,(x,y)\in R,\\ \hat{\phi}(x,y,0)-\frac{2}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial x}(x,y,0)=0,(x,y)\in R.\end{aligned} \tag{4.11c}$$

After Eqs. (4.11) are solved, the axial current J_z is defined by:

$$J_z(\boldsymbol{x})=-\frac{1}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial z}(\boldsymbol{x}). \tag{4.12}$$

If, in addition to Eq. (4.11a) we also define

$$\hat{\phi}_2(\boldsymbol{x})\equiv\int_{-1}^{1}\mu^2\psi(\boldsymbol{x},\mu)d\mu, \tag{4.13a}$$

then the standard P_3 approximation to Eqs. (4.9) and (4.10) is given by the coupled 1D axial diffusion equations:

$$\begin{aligned}-\frac{\partial}{\partial z}\frac{1}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(\boldsymbol{x})+2\hat{\phi}_2(\boldsymbol{x})\right]+\Sigma_a\hat{\phi}(\boldsymbol{x})=\frac{\nu\Sigma_f}{k_{\text{eff}}}\hat{\phi}(\boldsymbol{x})-\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x})+\frac{\partial J_y}{\partial y}(\boldsymbol{x})\right],\\ -\frac{\partial}{\partial z}\frac{9}{35\Sigma_t}\frac{\partial\hat{\phi}_2}{\partial z}(\boldsymbol{x})+\Sigma_t\hat{\phi}_2(\boldsymbol{x})=\frac{2}{5}\left[\Sigma_a\hat{\phi}(\boldsymbol{x})-\frac{\nu\Sigma_f}{k_{\text{eff}}}\hat{\phi}(\boldsymbol{x})\right],\end{aligned} \tag{4.13b}$$

with boundary conditions at $z=Z$:

$$\begin{aligned}\hat{\phi}(x,y,Z)+\frac{5}{4}\hat{\phi}_2(x,y,Z)+\frac{2}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(x,y,Z)+2\hat{\phi}_2(x,y,Z)\right]=0,\\ -\frac{1}{5}\hat{\phi}(x,y,Z)+\hat{\phi}_2(x,y,Z)+\frac{24}{35\Sigma_t}\frac{\partial\hat{\phi}_2}{\partial z}(x,y,Z)=0,\end{aligned} \tag{4.13c}$$

and at $z = 0$:

$$\begin{aligned} \hat{\phi}(x,y,0) + \frac{5}{4}\hat{\phi}_2(x,y,0) - \frac{2}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(x,y,0) + 2\hat{\phi}_2(x,y,0)\right] = 0, \\ -\frac{1}{5}\hat{\phi}(x,y,0) + \hat{\phi}_2(x,y,0) - \frac{24}{35\Sigma_t}\frac{\partial \hat{\phi}_2}{\partial z}(x,y,0) = 0. \end{aligned} \tag{4.13d}$$

After Eqs. (4.13) are solved, the axial current J_z is defined by

$$J_z(\boldsymbol{x}) = -\frac{1}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(\boldsymbol{x}) + 2\hat{\phi}_2(\boldsymbol{x})\right]. \tag{4.14}$$

The equations underlying the two 2D/1D methods currently used in MPACT can now be stated.

- The 2D/1D method with the P_1 axial leakage solver is defined by Eqs. (4.3), (4.6), (4.11b), and (4.12). These constitute a total of five equations for the five functions $\psi(\boldsymbol{x},\boldsymbol{\Omega})$, $J_x(\boldsymbol{x})$, $J_y(\boldsymbol{x})$, $\hat{\phi}(\boldsymbol{x})$, and $J_z(\boldsymbol{x})$. The boundary conditions for the 2D radial transport equation and the 1D axial diffusion equation are Eqs. (4.5) and (4.11c).
- The 2D/1D method with the P_3 axial leakage solver is defined by Eqs. (4.3), (4.6), (4.13a), and (4.14). These constitute a total of six equations for the six functions $\psi(\boldsymbol{x},\boldsymbol{\Omega})$, $J_x(\boldsymbol{x})$, $J_y(\boldsymbol{x})$, $\hat{\phi}(\boldsymbol{x})$, $\hat{\phi}_2(\boldsymbol{x})$, and $J_z(\boldsymbol{x})$. The boundary conditions for the 2D radial transport equation and the two 1D axial diffusion equations are Eqs. (4.5), (4.13b), and (4.13c).

The 2D/1D equations constitute a 2D radial transport equation, coupled to one or two 1D axial diffusion equations. The 2D radial transport and 1D axial diffusion equations are coupled through the isotropized axial and radial current terms. If the underlying problem is 2D with no axial variation, then the axial currents are zero, and the 2D/1D equations reduce to the physically correct 2D radial transport equation. If the underlying problem is 1D with no radial variation, then the radial currents are zero, and the 2D/1D equations reduce to the 1D axial P_1 or P_3 equations. Thus, the 2D/1D equations describe a process in which transport physics governs the radial variables x and y, and P_1 or P_3 physics governs the axial variable z.

4.4 Discretizations

In the 2D/1D equations, the energy variable is discretized using the conventional multigroup approximation. Details of this discretization are independent of the 2D/1D methodology and were discussed in Chapter 2. The key point is that in the derivation of the underlying multigroup 2D/1D equations from the multigroup transport equation, 2D/1D equations for each energy group are obtained that closely resemble the one-group 2D/1D equations described above. The only extra feature is that additional source terms appear due to the neutrons that scatter into a given group g from other groups g'.

The direction-of-flight variable $\boldsymbol{\Omega}$ is discretized using modular discrete ordinates quadrature sets that optimize the efficiency of the 2D/1D method for the types of pin cell geometries that MPACT is designed to simulate. The details

of how the modular quadrature sets are designed are discussed in Chapter 5. In this report, the *order* of a quadrature set is denoted by N, and the *number* of direction vectors in the quadrature set is denoted by N_q. Thus, a quadrature set of order N consists of a set of N_q discrete 2D vectors $\mathbf{\Omega}_n = (\Omega_{x,n}, \Omega_{y,n})$ and weights w_n, with $1 \leq n \leq N_q$. The modular quadrature sets are used to angularly discretize the 2D radial transport Eq. (4.3) and to construct the radial currents in Eqs. (4.6).

The following is a discussion of the discretization of the spatial variables in the 2D/1D equations. Only the 2D/1D equations with the P_1 axial leakage solver are considered. (The P_3 axial leakage solver is similar; it involves two 1D axial diffusion equations rather than one.)

To begin, the system V is discretized in the axial variable z into K slices:

$$0 = z_{1/2} < z_{3/2} < \cdots < z_{k-1/2} < z_{k+1/2} < \cdots < z_{K+1/2} = Z.$$

The 2D radial transport Eq. (4.3) and the 1D axial diffusion Eq. (4.11b) are now integrated axially over the k^{th} slice $z_{k-1/2} < z < z_{k+1/2}$. The integration of Eq. (4.3) produces a 2D radial transport equation for the axially integrated ψ:

$$\begin{aligned}\Omega_x \frac{\partial \psi_{n,k}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{n,k}}{\partial y}(x,y) + \Sigma_t \psi_{n,k}(x,y) = \frac{\Sigma_s}{4\pi} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m \\ + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m - \frac{1}{4\pi \Delta z_k} \left[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y) \right],\end{aligned} \tag{4.15}$$

where

$$\psi_{n,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} \psi(x,y,z,\mathbf{\Omega}_n) dz, \tag{4.16}$$

is the axially integrated ψ, and

$$J_{z,k\pm1/2}(x,y) = J_z(x,y,z_{k\pm1/2}), \tag{4.17}$$

are the net axial currents on the top and bottom of slice k.

The axial integration of the diffusion Eq. (4.11b) produces a 1D balance equation:

$$\begin{aligned}\frac{1}{\Delta z_k} \left[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y) \right] + \Sigma_a \hat{\phi}_k(x,y) \\ = \frac{\nu\Sigma_f}{k_{\text{eff}}} \hat{\phi}_k(x,y) - \left[\frac{\partial J_{x,k}}{\partial x}(x,y) + \frac{\partial J_{y,k}}{\partial y}(x,y) \right],\end{aligned} \tag{4.18}$$

where

$$\hat{\phi}_k(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} \hat{\phi}(x,y,z) dz, \tag{4.19}$$

is the axially-integrated $\hat{\phi}$, and

$$J_{x,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} J_x(x,y,z) dz, \tag{4.20a}$$

$$J_{y,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} J_y(x,y,z) dz, \tag{4.20b}$$

are the axially-integrated radial currents.

Operating on Eq. (4.15) by:

$$\sum_{n=1}^{N_q}(\cdot)w_n,$$

we obtain:

$$\frac{\partial J_{x,k}}{\partial x}(x,y)+\frac{\partial J_{y,k}}{\partial y}(x,y)+\Sigma_a\phi_k(x,y)=\frac{\nu\Sigma_f}{k_{\text{eff}}}\phi_k(x,y)-\frac{1}{\Delta z_k}\left[J_{z,k+1/2}(x,y)-J_{z,k-1/2}(x,y)\right], \tag{4.21}$$

where

$$\phi_k(x,y)=\sum_{m=1}^{N_q}\psi_{m,k}(x,y)w_m. \tag{4.22}$$

Since the leakage terms in Eqs. (4.18) and (4.21) are identical, it follows that

$$\phi_k(x,y)=\hat{\phi}_k(x,y). \tag{4.23}$$

Thus, the scalar fluxes from the 2D radial transport equation and the 1D axial diffusion equation are identical. Subsequent discretizations in x and y (discussed below) are constrained to preserve this important feature.

Equation (4.15) is the 2D radial transport equation for $\psi_{n,k}(x,y)$. When the axial current terms $J_{z,k\pm1/2}(x,y)$ are known, Eq. (4.15) completely determines $\psi_{n,k}(x,y)$. However, Eq. (4.18) alone cannot determine both $\hat{\phi}_k(x,y)$ and $J_{z,k\pm1/2}(x,y)$; at least one extra equation is needed. In the 2D/1D method described here, this extra equation comes from a standard nodal diffusion expansion of Eq. (4.11b). This method, which has been employed in neutron diffusion codes for many years, requires no modification for use in the 2D/1D method. In fact, any axial discretization method for Eq. (4.11b) that employs the balance Eq. (4.18) could be implemented in MPACT without changing the overall structure of the code.

The discretizations of Eqs. (4.15) and (4.18) are described next in the radial spatial variables x and y. First, the 2D region R is divided into a Cartesian coarse grid (typically, one coarse cell = one pin cell) and an unstructured fine grid is used to discretize a pin cell; typically, about 50 fine cells per coarse cell are used).

Second, the axial diffusion Eq. (4.18) is integrated radially over each coarse cell. Using radially coarse-cell flux-weighted cross sections (the origin of these cross sections is described in the next paragraph), the equation becomes one in which the unknowns are averaged radially (i.e., are constant) over each coarse cell. More precisely, the solution of the fully discrete axial diffusion equation depends on k (the slice index) and j (the coarse cell index), but not on i (the fine cell index). The radial current terms in this equation are obtained from the 2D radial transport calculation, and the axial current terms calculated in this equation are used in the 2D radial transport calculation.

Third, the 2D radial transport Eq. (4.15) is discretized on the fine radial grid for each coarse cell $V_{j,k}$ using the MOC. However, the axial current terms in this equation are obtained from the axial diffusion calculation, and these are constant over the coarse cell. Thus, in coarse cell $V_{j,k}$, the axial current terms depend on j and k, but not on i. The MOC method for Eq. (4.15) utilizes balance equations obtained by integrating Eq. (4.15) over each fine cell $v_{i,j,k}$.

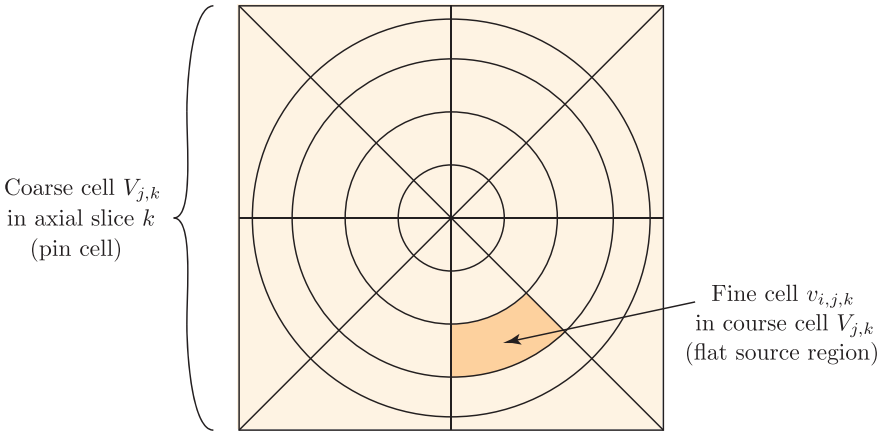


Figure 4.2. Fine and coarse spatial cells.

Summing these balance equations over all i which is over all fine cells in a single coarse cell, the balance equation for the coarse cell is obtained. Operating on this coarse cell balance equation by

$$\sum_{n=1}^{N_q} (\cdot) w_n,$$

the angularly integrated balance equation is obtained for coarse cell $V_{j,k}$. Upon convergence, this balance equation for $V_{j,k}$ becomes identical to the balance equation for the same cell, which is obtained by integrating the axial diffusion Eq. (4.18) over the cell. The flux-weighted coarse-cell homogenized cross sections used in the axial diffusion calculation are obtained from the 2D radial transport calculations.

This completes the discussion of the discretization of the 2D/1D equations. The important features of the spatial discretizations are (i) the use of coarse and fine radial grids, (ii) the discretization of the 2D radial transport equation on the fine grid, (iii) the discretization of the 1D axial diffusion equation on the coarse grid, and (iv) the use of balance equations with common leakage terms to ensure that the coarse mesh scalar fluxes from the 2D radial transport and the 1D axial diffusion equations are identical.

4.5 Iteration Strategy

The iterative 2D/1D process used in MPACT is described in this section. The iteration strategy used to solve the discretized 2D/1D equations involves (i) radial sweeps of the 2D radial transport equation, (ii) axial sweeps of the 1D axial diffusion equation, and (iii) solving a low-order coarse-mesh 3D CMFD diffusion equation described in Chapter 4.

Figure 4.3 outlines an iteration, providing detailed explanations of each step in the subsequent text. (The equation numbers in the figure and the discussion do not refer to the fully discretized equations solved in MPACT. Instead, they refer to the closest version of the fully discrete equations that are printed in this document.)

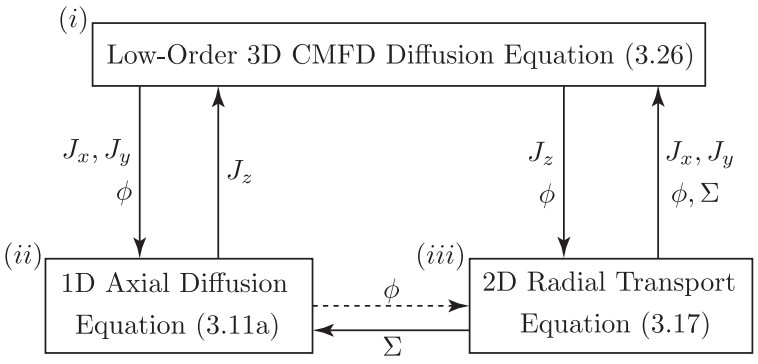


Figure 4.3. MPACT 2D/1D iteration strategy.

(i) First, the $\hat{D}$ terms are calculated from Eq. (7.7) using coarse grid scalar flux and current estimates obtained from the previous 1D axial diffusion and 2D radial transport calculations. (At the beginning of the first iteration, the $\hat{D}$ terms are estimated to be zero.) Then, using a global 3D low-order CMFD diffusion calculation, the solution of Eq. (7.2) is obtained. In this calculation, the homogenized flux-weighted cross sections are obtained from the previous 2D radial transport calculation, or they are volume-averaged for the first iteration. The 3D CMFD calculation results in: new coarse grid cell-averaged scalar flux estimates, new coarse-grid edge current estimates, and a new eigenvalue estimate.

(ii) Next, a solve of the 1D axial diffusion equations is performed, using the coarse-grid edge current estimates from step (i). This solve can consist of local calculations that involve no axial sweeping; in this case, relaxation is needed to stabilize the iterations for small Δz. Alternatively, the solve can consist of several axial sweeps to partly converge the discrete 1D equations. (If this sweeping is performed, relaxation is not likely needed.) The end result consists of new coarse-cell scalar fluxes, and new estimates of the axial currents.

(iii) Next, the most recent available coarse-grid scalar flux estimates are used to update (renormalize) the estimates of the fine-grid scalar fluxes. Then, using the latest axial current information, a specified number (typically, three) of 2D transport sweeps on each axial slice is performed to improve the estimates of the fine grid scalar fluxes. At the end of these sweeps, new coarse grid radial current and scalar flux estimates are available. Also, new estimates of the coarse-grid flux-weighted cross sections are available; these homogenized cross sections are used in subsequent coarse-grid calculations.

(iv) If the fine-grid scalar fluxes and eigenvalue are sufficiently converged, the iteration process is terminated. Otherwise, the computer returns to step (i), and another iteration is initiated.

4.6 Appendix A: Relaxation Strategy

The early 2D/1D method (in the DeCART code) was simpler and less stable than the 2D/1D method in MPACT described previously in this chapter. The old 2D/1D method was based on Eq. (4.15):

$$\Omega_x \frac{\partial \psi_{n,k}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{n,k}}{\partial y}(x,y) + \Sigma_t \psi_{n,k}(x,y) = \frac{\Sigma_s}{4\pi} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m - \frac{1}{4\pi\Delta_k} \left[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y) \right], \tag{4.24}$$

but not on the axial diffusion Eq. (4.11b). In its place, the simpler Fick's Law approximation was used:

$$J_{z,k+1/2}(x,y) = -D_{k+1/2}(x,y) \frac{\phi_{k+1}(x,y) - \phi_k(x,y)}{\Delta_{k+1/2}}, \tag{4.25a}$$

where

$$D_{k+1/2}(x,y) = \frac{\Delta_{k+1} + \Delta_k}{3\Sigma_{t,k+1}(x,y)\Delta_{k+1} + 3\Sigma_{t,k}(x,y)\Delta_k}, \tag{4.25b}$$

$$\Delta_{k+1/2} = \frac{1}{2}(\Delta_{k+1} + \Delta_k). \tag{4.25c}$$

An obvious iteration method for solving these equations consists of the following steps: (i) for estimated axial leakages, perform independent sweeps on each axial slice, (ii) use the results of these sweeps to update the axial leakages, and (iii) return to step (i) or stop if the solution is converged.

A deficiency of this method was its instability for problems with thin axial slices. To understand and prevent this instability, a Fourier analysis of the iteration method was performed [51], [50]. It was found that by properly under-relaxing the iterations, unconditional stability can be guaranteed. These Fourier analysis results were confirmed in numerical simulations. Later, it was demonstrated (again, with Fourier analysis and numerical testing) that the CMFD-accelerated iterations could also be stabilized using under-relaxation [52].

The following section provides some details and results of these Fourier analyses. This discussion is included because the 2D/1D method described earlier – which *is* commonly used in MPACT – has been implemented in ways that can, in certain circumstances, require under-relaxations. The 2D/1D method based on Eqs. (4.24) and (4.25) is *not* commonly used in MPACT.

The Fourier analysis is based on the following fixed-source version of Eqs. (4.24) and (4.25):

$$\left(\Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Sigma_{t,k} \right) \psi_k(x,y,\mathbf{\Omega}) = \frac{1}{4\pi} \left\{ \Sigma_{s,k}\phi_k(x,y) + Q_k(x,y) + \frac{1}{\Delta_k} \left[\frac{D_{k+1/2}}{\Delta_{k+1/2}} \left(\phi_{k+1}(x,y) - \phi_k(x,y) \right) - \frac{D_{k-1/2}}{\Delta_{k-1/2}} \left(\phi_k(x,y) - \phi_{k-1}(x,y) \right) \right] \right\}, \tag{4.26a}$$

$$\phi_k(x,y) = \int \psi_k(x,y,\mathbf{\Omega}') d\Omega'. \tag{4.26b}$$

To solve this equation, a simple source iteration scheme that can be made stable for all $\Delta_z > 0$ is considered. While the right side of the above equation depends only on the scalar fluxes ϕ in adjacent axial slices, a simple 2D sweep is considered on each slice to update the scalar flux:

$$\left(\Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Sigma_{t,k}\right) \psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega}) = \frac{1}{4\pi}\left\{\Sigma_{s,k}\phi_k^{(\ell)}(x,y) + Q_k(x,y) + \frac{1}{\Delta_k}\left[\frac{D_{k+1/2}}{\Delta_{k+1/2}}\left(\phi_{k+1}^{(\ell)}(x,y) - \phi_k^{(\ell)}(x,y)\right) - \frac{D_{k-1/2}}{\Delta_{k-1/2}}\left(\phi_k^{(\ell)}(x,y) - \phi_{k-1}^{(\ell)}(x,y)\right)\right]\right\}, \tag{4.27a}$$

$$\phi_k^{(\ell+1/2)}(x,y) = \int \psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega}')d\Omega', \tag{4.27b}$$

followed by a (nonstandard) relaxation step to define the end-of-iteration scalar flux:

$$\phi_k^{(\ell+1)}(x,y) = \theta\phi_k^{(\ell+1/2)}(x,y) + (1-\theta)\phi_k^{(\ell)}(x,y). \tag{4.27c}$$

In the Fourier analysis of this method, Eqs. (4.27) are not treated with any angular or radial spatial discretizations. While these choices affect the accuracy of the discrete solution, they do not affect the iterative performance in converging to this solution. The relaxation parameter θ in Eq. (4.27c) is to be determined; if $\theta = 1$, the method defined by Eqs. (4.27) is basically source iteration. In each iteration, the numerical solutions in slice k are directly coupled only to the numerical solutions in the neighboring slices $k+1$ and $k-1$. Therefore, many iterations may be required for the numerical fluxes in all the axial slices $1 \le k \le K$ to sufficiently communicate. For an infinite, homogeneous medium with uniform $\Delta_k = \Delta_z$, the iterative performance of the method shown above can be assessed by a Fourier analysis. Referring to Kelley and Larsen's work [51, 52] for details, for $\theta = 1$ (the standard source iteration method), the growth factor ω is bounded from above and below by

$$\omega_{min} \le \omega \le \omega_{max}, \tag{4.28}$$

where

$$\omega_{max} = c = \frac{\Sigma_s}{\Sigma_t} = \text{ scattering ratio}, \tag{4.29a}$$

which is attained for "flat" radial and axial modes. Also,

$$\omega_{min} = \begin{cases} 0 & , \quad \Sigma_t\Delta_z \ge \frac{2}{\sqrt{3c}}, (\text{ "large" } \Delta_z), \\ c - \frac{4}{3(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z < \frac{2}{\sqrt{3c}}, (\text{ "small" } \Delta_z), \end{cases} \tag{4.29b}$$

which is attained (i) for large Δ_z by radially oscillatory modes, and (ii) for small Δ_z by radially flat modes and axially oscillatory modes. The error of iterate n satisfies

$$\text{Error of Iterate } n \approx (\text{constant})\ \rho^n,$$

where

$$\rho = \max(|\omega_{min}|, |\omega_{max}|) = \text{spectral radius}.$$

Therefore, the method is unstable when $\rho \ge 1$, is stable but slowly converging when $\rho < 1$ but $\rho \approx 1$, and is stable and rapidly converging when $\rho \ll 1$.

Using Eq. (4.27c), the Fourier analysis for $0 \leq \theta \leq 1$ yields

$$\theta\omega_{min} + 1 - \theta \leq \omega \leq \theta\omega_{max} + 1 - \theta. \tag{4.30}$$

For $\theta = 1$ (the source iteration method originally in DeCART), Eqs. (4.29) and (4.30) give

$$\rho = |\omega|_{max} = \begin{cases} c & , \quad \sqrt{\frac{2}{3c}} < \Sigma_t\Delta_z, \\ \frac{4}{3(\Sigma_t\Delta_z)^2} - c & , \quad \Sigma_t\Delta_z \leq \sqrt{\frac{2}{3c}}. \end{cases} \tag{4.31}$$

This method is stable for

$$\Sigma_t\Delta_z > \frac{2}{\sqrt{3(1+c)}},$$

but for small Δ_z it becomes unstable, similar to the original method in DeCART.

The optimum value of θ in Eq. (4.30) is the value for which the left and right sides are equal in magnitude but opposite in sign:

$$\theta_{opt}\omega_{min} + 1 - \theta_{opt} = -\left[\theta_{opt}\omega_{max} + 1 - \theta_{opt}\right]. \tag{4.32a}$$

Thus,

$$\theta_{opt} = \frac{2}{2 - (\omega_{max} + \omega_{min})}, \tag{4.32b}$$

and then

$$\begin{aligned} \rho = |\omega|_{max} &= \theta_{opt}\omega_{max} + 1 - \theta_{opt} \\ &= \frac{\omega_{max} - \omega_{min}}{2 - (\omega_{max} + \omega_{min})}. \end{aligned} \tag{4.32c}$$

Combining Eqs. (4.29) and (4.32) results in:

$$\theta_{opt} = \begin{cases} \frac{2}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t\Delta_z, \\ \frac{3(\Sigma_t\Delta_z)^2}{2+3(1-c)(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z \leq \frac{2}{\sqrt{3c}}, \end{cases} \tag{4.33a}$$

$$\rho = \begin{cases} \frac{c}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t\Delta_z, \\ \frac{2}{2+3(1-c)(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z \leq \frac{2}{\sqrt{3c}}. \end{cases} \tag{4.33b}$$

Equation (4.33b) shows that the iterative method defined by Eqs. (4.27) with θ defined by Eq. (4.33a) is stable for all scattering ratios $0 \leq c \leq 1$ and all axial grids $\Delta_z > 0$. Like standard source iteration applied to the S_N equations, this method becomes slowly converging as $c \to 1$. It also becomes slowly converging as $\Delta_z \to 0$. However, like source iteration, it does not become unstable. The DeCART-like method with $\theta = 1$ is stable only for sufficiently large axial grids.

CMFD acceleration can be applied to the source iteration method presented above to effectively couple the axial planes to more than their nearest neighbor by solving an axial diffusion equation (embedded within the CMFD method) and thereby more rapidly converging the transport solution. However, before applying a Fourier analysis, it is necessary to linearize this method. Previous work shows that the linearized form of CMFD is equivalent to diffusion synthetic acceleration (DSA) [61]. The DSA equivalent of this CMFD system yields an update term $\delta\phi_k^{(\ell+1/2)}$ for the scalar flux. (Kelley and Larsen [52] provide an explicit definition of this correction term.) A CMFD iteration then consists of

(i) an axial sweep, as shown in Eq. (4.27a),
(ii) calculation of $\phi_k^{\ell+1/2}$, Eq. (4.27b),
(iii) the CMFD calculation of $\delta\phi_k^{(\ell+1/2)}$ [52], and
(iv) the relaxed update equation

$$\phi_k^{(\ell+1)}(x,y) = \theta\left(\phi_k^{(\ell+1/2)}(x,y) + \delta\phi_k^{(\ell+1/2)}\right) + (1-\theta)\phi_k^{(\ell)}(x,y). \tag{4.34}$$

These equations consitute a complete linearized CMFD iteration scheme. Because steps (i) and (ii) are unchanged, so are their corresponding equations in the Fourier analysis. The growth factor for the CMFD-accelerated planar syntehsis scheme with $\theta = 1$ (no relaxation) may be written in the following form:

$$\omega_1(\lambda_r,\zeta) = 1 - \frac{1-\zeta I_0(\lambda_r)}{1-\zeta\left(1+\frac{\lambda_r^2}{3}\right)^{-1}}, \tag{4.35a}$$

$$\zeta = c - \frac{\Lambda_z^2}{3}, \tag{4.35b}$$

$$\Lambda_z = \frac{\sin(\lambda_z \Sigma_t \Delta_z/2)}{\Sigma_t \Delta_z/2}. \tag{4.35c}$$

The maximum and minimum values of $\omega_1(\lambda_r,\zeta)$, for $0 \le \lambda_r < \infty$ and $0 \le \Lambda_z \le (\Sigma_t\Delta_z/2)^{-1}$ must be determined to be used in Eq. (4.32b) to obtain the optimum relaxation factor. Here, λ_r is the spatial frequency in the radial direction. Unfortunately, extremum values of ω cannot be determined explicitly given the form of Eqs. (4.35). This function is evaluated numerically, and these extrema can be seen in Figure 4.4. The extremum of the growth factor decreases monotonically with the parameter ζ, and the maximum value of ω (occurring for $\zeta = c = 1$) is given by $\omega \approx 0.2247$ (the standard result from traditional CMFD).

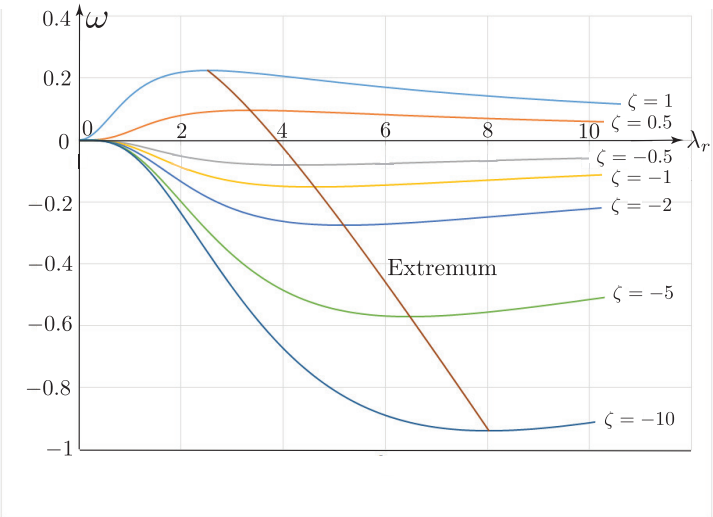


Figure 4.4. Growth factor ω for various radial frequencies λ_r for a sample of parametrized values ζ.

Figure 4.4 shows that when $\zeta > 0$, the maximum value of $\omega_1(\lambda_r,\zeta)$ is positive, and when $\zeta < 0$, the minimum value of $\omega_1(\lambda_r,\zeta)$ is negative. For each ζ, let $\lambda_{ext}(\zeta)$ denote the value of λ_r at which these extrema occur. An approximate

relationship between λ_{ext} and ζ, obtained by curve fitting, is given by:

$$\begin{aligned} \lambda_{ext} &\approx S + M(F-\zeta)^P, \\ S = 1.93801895412889, &\quad M = 1.88037759461481, \\ F = 1.07821249297909, &\quad P = 0.487975837139675. \end{aligned} \tag{4.36}$$

Now that λ_{ext} is known, ω_{min} can be calculated from Eqs. (4.35), and thus θ_{opt} can be calculated from Eq. (4.32b).

Traditional (unrelaxed) CMFD occurs for $\theta = 1$; this method is conditionally stable based on the axial plane thickness. This method is stable if $\zeta \gtrapprox -10.947$, or

$$\Sigma_t \Delta_z \gtrapprox \frac{2}{\sqrt{3(c+10.947)}}, \tag{4.37}$$

but otherwise it is unstable. However, as is the case with source iteration, the optimum relaxation factor yields optimal convergence and unconditional stability.

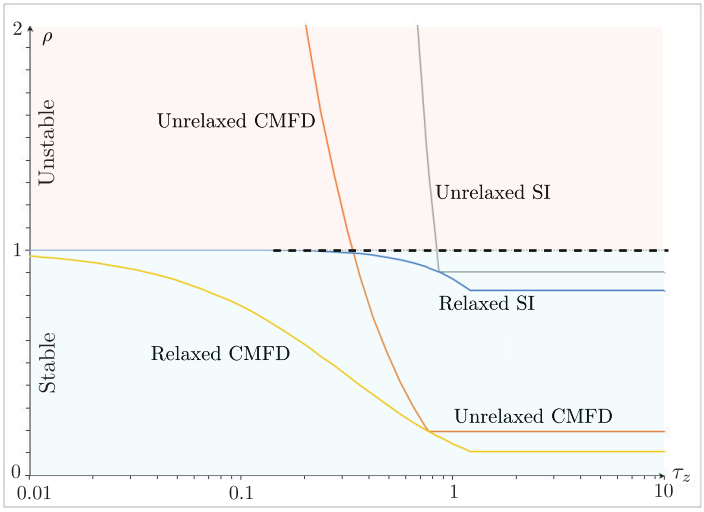


Figure 4.5. Spectral radius ρ for SI and CMFD vs. optical thickness $\tau_z = \Sigma_t \Delta_z$, with c =0.9.

Figure 4.5 illustrates that the region of conditional stability for the unrelaxed CMFD method is much larger than that of unrelaxed source iteration. This improvement comes from the improved communication between the axial slices. At each iterate, the slices receive information from the whole system, rather than just from neighboring slices. The figure also shows that (i) the optimally relaxed SI scheme is unconditionally stable, (ii) the optimally relaxed CMFD method is unconditionally stable, and (iii) the relaxed CMFD method is significantly more efficient that the relaxed SI method. When the axial slice thicknesses tend to 0, the optimally relaxed SI and CMFD methods both become increasingly slow in converging. However, the CMFD method always converges much faster than SI, and for realistic problems, the axial slice thicknesses are not small enough to be a practical concern.

The analysis above applies specifically to the 2D/1D Eqs. (4.26), which employ a finite-differenced Fick's Law to express the axial current on each axial slice edge in terms of the scalar fluxes in the slices above and below the edge.

This simple axial solver is available in MPACT, but more sophisticated nodal diffusion and nodal P_3 solvers are more commonly used. When these nodal solvers are used instead of the finite-difference method described above, the resulting discrete methods are stabilized, and there is no need for relaxation.

However, nodal diffusion solvers introduce complexity because they effectively couple each axial slice to more than just its nearest neighbors. In MPACT, the full 1D axial solves are typically compromised by limiting the coupling between axial slices to the sufficiently near neighbors. Although this reduces the algebra and the amount of data that must be moved between processors, it has a negative effect on stability. However, it has been found that when the under-relaxations described above are included, the iterations in MPACT become stable. Whether these under-relaxation parameters are optimal for more complicated iteration schemes containing incomplete axial solves is an open question. However, they do rigorously stabilize the iterations when the simplest axial solver in Eq. (4.27a) is used.

5. The Method of Characteristics Solution Methodology

This chapter provides a detailed derivation of the method of characteristics (MOC) equations and introduces the concepts and algorithms used in MPACT MOC solvers. A detailed derivation, which highlights important approximations at each step, is provided and is followed by a description of the algorithm for the iterative solution of these equations. Then the techniques required to discretize a problem are described; these techniques are common to any multi-dimensional MOC transport solver. The descriptions in this chapter are given for both 2D and 3D implementations.

5.1 2D Radial Solution Methodology by the MOC

The derivation of the MOC solution to the Boltzmann neutron transport equation starts with the steady-state, axially integrated, multigroup discrete ordinates equations shown below. This combines the multigroup and discrete ordinates approximations of Eq. (2.23) with Eq. (4.15) and does not assume isotropic scattering.

$$\begin{aligned}
&\Omega_x \frac{\partial \psi_{k,m,g}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{k,m,g}}{\partial y}(x,y) + \Sigma_{t,k,g}(x,y)\psi_{k,m,g}(x,y) \\
&\qquad = \sum_{g'=1}^{G}\sum_{m'=1}^{M_N} \Sigma_{s,k,g'\to g}(x,y,\boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{k,m',g'}(x,y)w_{m'} \\
&\qquad\qquad + \frac{\chi_{k,g}(x,y)}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N} \nu\Sigma_{f,k,g'}(x,y)\psi_{k,m',g'}(x,y)w_{m'} \\
&\qquad\qquad - \frac{1}{4\pi\Delta z_k}\left[J_{z,k+1/2,g}(x,y) - J_{z,k-1/2,g}(x,y)\right], \\
&x \in X, y \in Y, k \in z_{k-1/2} \le z \le z_{k+1/2}, \\
&1 \le m \le M_N, 1 \le g \le G,
\end{aligned} \tag{5.1a}$$

$$\psi_{k,m,g}(x,y) = 0, \quad (x,y) \in \partial V, \boldsymbol{\Omega}_m \cdot \boldsymbol{n} < 0. \tag{5.1b}$$

Next, the variable q is introduced to represent the right-hand side of Eq. (5.1) and simplify the notation going forward.

$$q_{k,m,g}(x,y) = \sum_{g'=1}^{G}\sum_{m'=1}^{M_N} \Sigma_{s,k,g'\rightarrow g}(x,y,\boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{k,m',g'}(x,y)w_{m'} + \frac{\chi_{k,g}(x,y)}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\nu\Sigma_{f,k,g'}(x,y)\psi_{m',g'}(x,y)w_{m'} - \frac{1}{4\pi\Delta z_k}\left[J_{z,k+1/2,g}(x,y) - J_{z,k-1/2,g}(x,y)\right], \quad (5.2)$$

yielding

$$\Omega_x\frac{\partial\psi_{k,m,g}}{\partial x}(x,y) + \Omega_y\frac{\partial\psi_{k,m,g}}{\partial y}(x,y) + \Sigma_{t,k,g}(x,y)\psi_{k,m,g}(x,y) = q_{k,m,g}(x,y). \quad (5.3)$$

From here the MOC is applied to Eq. (5.3). This involves a transformation of the dependentvariables to the *characteristic direction*, allowing one to rewrite the partial derivatives in Eq. (5.3) into a single total derivative. The variable transformation is given by:

$$\boldsymbol{r} = \boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r, \begin{cases} x(s) = x_0 + s\Omega_x^r\sqrt{1-\mu_m^2}, \\ y(s) = y_0 + s\Omega_y^r\sqrt{1-\mu_m^2}. \end{cases} \quad (5.4a)$$

Here $r = (x,y)$ and $\boldsymbol{\Omega}_m^r = (\Omega_x^r, \Omega_y^r) = (\cos\omega_m, \sin\omega_m)$. This leads to the characteristics form of Eq. (5.3).

$$\frac{d\psi_{k,m,g}}{ds}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r) + \frac{\Sigma_{t,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)}{\sqrt{1-\mu_m^2}}\psi_{m,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r) = \frac{q_{m,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)}{\sqrt{1-\mu_m^2}}. \quad (5.5)$$

which can be solved analytically using the integrating factor

$$\exp\left(-\int_0^s \frac{\Sigma_{t,k,g}(\boldsymbol{r}_0 + s'\boldsymbol{\Omega}_m^r)}{\sqrt{1-\mu_m^2}}ds'\right).$$

This results in the following expression for ψ as the solution of the axially integrated, multigroup, discrete ordinates transport equation required by the 2D/1D method outlined in Chapter 4.

$$\psi_{k,m,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r) = \psi_{k,m,g}(\boldsymbol{r}_0)\exp\left(-\int_0^s \frac{\Sigma_{t,k,g}(\boldsymbol{r}_0 + s'\boldsymbol{\Omega}_m^r)}{\sqrt{1-\mu_m^2}}ds'\right) + \int_0^s q_{k,m,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)\exp\left(-\int_{s'}^s \frac{\Sigma_{t,k,g}(\boldsymbol{r}_0 + s'\boldsymbol{\Omega}_m^r)}{\sqrt{1-\mu_m^2}}ds''\right)ds', \quad (5.6)$$

where

$$q_{m,g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r) = \sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\Sigma_{s,g'\rightarrow g}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r, \boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{m',g'}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)w_{m'} + \frac{\chi_g(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\nu\Sigma_{f,g'}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)\psi_{m',g'}(\boldsymbol{r}_0 + s\boldsymbol{\Omega}_m^r)w_{m'}. \quad (5.7)$$

In section 5.4, the discretization of Eq. (5.6) is described, and in Section 5.5 the iterative evaluation of the discretized MOC equations to obtain its solution is explained. The following section first presents an analogous derivation for the MOC solution in 3D.

5.2 Method of Characteristics Solution of the Boltzmann Transport Equation in 3D

The derivation of the MOC solution to the Boltzmann neutron transport equation in 3D starts with the steady-state, multigroup, discrete ordinates equation, Eq. (2.23), which is reproduced below for convenience.

$$\begin{aligned}\boldsymbol{\Omega}_m \cdot \boldsymbol{\nabla}\psi_{m,g}(\boldsymbol{x}) + \Sigma_{t,g}(\boldsymbol{x})\psi_{m,g}(\boldsymbol{x}) = &\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{m',g'}(\boldsymbol{x})w_{m'} \\ &+\frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\nu\Sigma_{f,g'}(\boldsymbol{x})\psi_{m',g'}(\boldsymbol{x})w_{m'},\end{aligned}$$

$$\boldsymbol{x}\in V, 1\leq m\leq M_N, 1\leq g\leq G, \tag{5.8a}$$

$$\psi_{m,g}(\boldsymbol{x}) = 0, \quad \boldsymbol{x}\in\partial V, \boldsymbol{\Omega}_m\cdot\boldsymbol{n}<0. \tag{5.8b}$$

The variable q is introduced to simplify the right-hand side:

$$\begin{aligned}q_{m,g}(\boldsymbol{x}) = &\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{m',g'}(\boldsymbol{x})w_{m'} \\ &+\frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\nu\Sigma_{f,g'}(\boldsymbol{x})\psi_{m',g'}(\boldsymbol{x})w_{m'},\end{aligned} \tag{5.9}$$

yielding

$$\boldsymbol{\Omega}_m \cdot \boldsymbol{\nabla}\psi_{m,g}(\boldsymbol{x}) + \Sigma_{t,g}(\boldsymbol{x})\psi_{m,g}(\boldsymbol{x}) = q_{m,g}(\boldsymbol{x}). \tag{5.10}$$

The MOC is then applied. The spatial and angular variables of Eq. (5.10), are transformed into the *characteristic direction* using the following identities:

$$\boldsymbol{x} = x\hat{\boldsymbol{i}} + y\hat{\boldsymbol{j}} + z\hat{\boldsymbol{k}}, \tag{5.11a}$$

$$\boldsymbol{\Omega}_m = \Omega_{x,m}\boldsymbol{i} + \Omega_{y,m}\boldsymbol{j} + \Omega_{z,m}\boldsymbol{k}, \tag{5.11b}$$

$$\boldsymbol{x} = \boldsymbol{x}_0 + s\boldsymbol{\Omega}_m \Rightarrow \begin{cases} x(s) = x_0 + s\Omega_{x,m}, \\ y(s) = y_0 + s\Omega_{y,m}, \\ z(s) = z_0 + s\Omega_{z,m}. \end{cases} \tag{5.11c}$$

This leads to the characteristic form of Eq. (5.10), where the partial derivative can be replaced by the total derivative because of the transformation to the characteristic direction:

$$\frac{d\psi_{m,g}}{ds}(\boldsymbol{x}_0 + s\boldsymbol{\Omega}_m) + \Sigma_{t,g}(\boldsymbol{x}_0 + s\boldsymbol{\Omega}_m)\,\psi_{m,g}(\boldsymbol{x}_0 + s\boldsymbol{\Omega}_m) = q_{m,g}(\boldsymbol{x}_0 + s\boldsymbol{\Omega}_m). \tag{5.12}$$

Equation (5.11) can be solved analytically using the integrating factor:

$$\exp\left(-\int_0^s \Sigma_{t,g}(\boldsymbol{x}_0 + s'\boldsymbol{\Omega}_m)ds'\right),$$

resulting in the following expression for ψ:

$$\psi_{m,g}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_m)=\psi_{m,g}(\boldsymbol{x}_0)\exp\left(-\int_0^s\Sigma_{t,g}(\boldsymbol{x}_0+s'\boldsymbol{\Omega}_m)ds'\right)+\int_0^s q_{m,g}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_m)\exp\left(-\int_{s'}^s\Sigma_{t,g}(\boldsymbol{x}_0+s''\boldsymbol{\Omega}_m)ds''\right)ds', \quad (5.13)$$

and q:

$$q_{m,g}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_m)=\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\Sigma_{s,g'\to g}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_0,\boldsymbol{\Omega}_{m'}\cdot\boldsymbol{\Omega}_m)\psi_{m',g'}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_0)w_{m'}+\frac{\chi_g(\boldsymbol{x}_0+s\boldsymbol{\Omega}_0)}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\sum_{m'=1}^{M_N}\nu\Sigma_{f,g'}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_0)\psi_{m',g'}(\boldsymbol{x}_0+s\boldsymbol{\Omega}_0)w_{m'}. \quad (5.14)$$

Equation (5.13) is the solution of the characteristic form of the multigroup discrete ordinates transport equation in 3D. Note that Eq. (5.13) differs from Eq. (5.6) only in the argument of the exponential, and that Eq. (5.7) and (5.14) are identical for their respective spatial dimensionality. In the following, Section 5.3, approximations to allow for the evaulation of the integrals in Eq. (5.13) are given. Section 5.4 describes the approaches to discretizing this equation for its numerical solution in detail. The approximation and discretization of Eq. (5.13) is essentially the same for Eq. (5.6).

5.3 Approximations of the Characteristics Transport Equation

5.3.1 Constant Material Properties in a Discrete Region

To discretize the spatial domain, the problem is divided into discrete spatial regions, and within each region, it is assumed that the material properties are spatially constant. This spatial discretization, as illustrated in Figure 5.1, leads to a spatial discretization scheme that is first-order accurate.

With these assumptions, Eq. (5.6) and (5.7), or similarly , Eq. (5.6) and (5.7), reduce to the following equations for each characteristic ray k passing through each discrete region i:

$$\psi^{out}_{k,i,g,m,n}=\psi^{in}_{k,i,g,m,n}\exp\left(\frac{-\Sigma_{t,k,i,g}s_{k,i,m,n}}{\sqrt{1-\mu_m^2}}\right)+\int_0^{s_{k,i,m,n}}q_{k,i,g,m}(s')\exp\left(-\frac{\Sigma_{t,k,i,g}(s_{k,i,m,n}-s')}{\sqrt{1-\mu_m^2}}\right)ds', \quad (5.15)$$

$$q_{k,i,g,m}(s)=\sum_{g'=1}^{G}\sum_{m'=1}^{M}\Sigma_{s,k,i,g'\to g}(\boldsymbol{\Omega}^r_{m'}\cdot\boldsymbol{\Omega}^r_m)\psi_{k,i,g',m'}(s)w_{m'}+\frac{\chi_{k,i,g}}{4\pi}\sum_{g'=1}^{G}\nu\Sigma_{f,k,i,g'}\sum_{m'=1}^{M}\psi_{k,i,g',m'}(s)w_{m'}, 0\le s\le s_{k,i,m,n}. \quad (5.16)$$

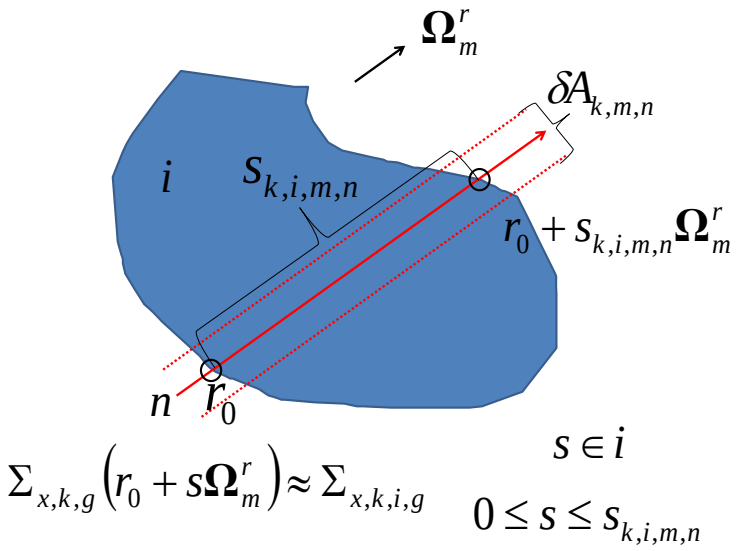


Figure 5.1. Spatial discretization with constant properties

In Eq. (5.15), the shorthand notation,

$$\psi^{in}_{k,i,g,m,n} = \psi_{k,i,g,m,n}(\boldsymbol{r}_0) = \psi_{k,i,g,m,n}(s=0),$$

and

$$\psi^{out}_{k,i,g,m,n} = \psi_{k,i,g,m,n}(\boldsymbol{r}_0 + s_{k,i,m,n}\boldsymbol{\Omega}_m) = \psi_{k,i,g,m,n}(s=s_{k,i,m,n}),$$

is used. For adjacent regions i and $i+1$, the identity

$$\psi^{out}_{k,i,g,m,n} = \psi^{in}_{k,i+1,g,m,n}, \tag{5.17}$$

is also used.

5.3.2 Flat Source Approximation

Next, the source, $q_{k,i,g,m}(s)$, is assumed to be constant within each discrete spatial region. This is commonly referred to as the *flat source approximation*. It is the simplest approximation for the spatial dependence of the source, and is accurate in the fine limit of the spatial mesh.

Other approximations (e.g., linear and quadratic approximations) to the space- dependence of the source have been developed, but for this work, only the flat source approximation is considered. With the flat source approximation, the space-dependent source in each region is replaced by a constant cell-average source, and the remaining integral over

s' in Eq. (5.15) can be evaluated analytically. This leads to the following equations:

$$\psi^{out}_{k,i,g,m,n} = \psi^{in}_{k,i,g,m,n} \exp\left(\frac{-\Sigma_{t,k,i,g} s_{k,i,m,n}}{\sqrt{1-\mu_m^2}}\right) + \frac{q_{k,i,g,m}}{\Sigma_{t,k,i,g}}\left[1 - \exp\left(\frac{-\Sigma_{t,k,i,g} s_{k,i,m,n}}{\sqrt{1-\mu_m^2}}\right)\right], \tag{5.18}$$

$$q_{k,i,g,m} = \frac{\chi_{k,i,g}}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \nu\Sigma_{f,k,i,g'} \sum_{m'=1}^{M} w_{m'} \overline{\psi}_{k,i,g',m'} + \sum_{g'=1}^{G}\sum_{m'=1}^{M} w_{m'} \Sigma_{s,k,i,g'\rightarrow g}(\mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m)\overline{\psi}_{k,i,g',m'}. \tag{5.19}$$

Equation (5.19) introduces a new term, $\overline{\psi}_{k,i,g,m}$, the *region-averaged angular flux*. This is computed using the following:

$$\overline{\psi}_{i,g,m} = \frac{\sum_{n\in i} \tilde{\psi}_{k,i,g,m,n} s_{k,i,m,n} \delta A_{m,n}}{\sum_{n\in i} s_{k,i,m,n} \delta A_{m,n}}. \tag{5.20}$$

Here, $\delta A_{m,n}$ is the cross-sectional area of the characteristic ray (illustrated in Figure 5.1), and $\tilde{\psi}_{k,i,g,m,n}$ is the *segment-averaged angular flux*, defined by:

$$\tilde{\psi}_{k,i,g,m,n} = \frac{\int_0^{s_{k,i,m,n}} \psi_{k,i,g,m}(s')ds'}{\int_0^{s_{k,i,m,n}} ds'} \Rightarrow \frac{\psi^{in}_{k,i,g,m,n} - \psi^{out}_{k,i,g,m,n}}{\Sigma_{t,k,i,g} s_{k,i,m,n}} + \frac{q_{k,i,g,m}}{\Sigma_{t,k,i,g}}. \tag{5.21}$$

Eqs. (5.18) and (5.21) are the fundamental discretized MOC equations that must be evaluated to obtain a solution of the angular flux for a given source q.

5.3.3 Isotropic Scattering Source Approximation

The final approximation is the treatment of the scattering source. The usual way to represent this source is to expand the differential scattering cross section as an infinite series of Legendre polynomials of the cosine of the scattering angle to account for anistropic scattering. For LWR analysis, linearly anistropic scattering is often sufficiently accurate, but it can take significantly more computational resources to explicitly treat anisotropic scattering. Therefore, it is quite common to formulate an isotropic source with some approximation. This subsection, discusses this approximation for problems independent of the spatial discretization (the subscripts i and k are ommitted).

In the P_1 scattering approximation, the source term for energy group g is approximated as:

$$q_g = q_{0,g} + q_{1,g}, \tag{5.22a}$$

$$q_{0,g} = \frac{1}{4\pi}\left(\sum_{g'} \Sigma_{s0,g'\rightarrow g}\phi_{g'} + \chi_g \sum_{g'} \nu\Sigma_{f,g'}\phi_{g'}\right), \tag{5.22b}$$

$$q_{1,g}(\mathbf{\Omega}) = \frac{3}{4\pi}\left(\sum_{g'} \Sigma_{s1,g'\rightarrow g} \int_{4\pi} (\mathbf{\Omega}\cdot\mathbf{\Omega}')\psi_{g'}(\mathbf{\Omega}')d\mathbf{\Omega}'\right), \tag{5.22c}$$

where $q_{0,g}$ is the isotropic source that includes the P_0 scattering and fission sources, $q_{1,g}$ is the linearly anisotropic scattering source, and $\psi_g(\mathbf{\Omega})$ is the regional angular flux determined by Eq. (5.20). The anisotropic scattering source

in Eq. (5.22) can be further simplified by inserting the net currents:

$$q_{1,g}(\mathbf{\Omega}) = \frac{3}{4\pi}\left(\sum_{g'}\Sigma_{s1,g'\rightarrow g}(\mathbf{\Omega}\cdot\boldsymbol{J}_g)\right), \tag{5.23}$$

where

$$\boldsymbol{J}_g = (J_{x,g}, J_{y,g}, J_{z,g}),$$

and the elements of $\boldsymbol{J}$ are defined by Eq. (4.6) and Eq. (4.4) The previous equations indicate that for the P_1 scattering source approximation, the net currents and the scalar flux must be stored in order to determine the anisotropic scattering source.

For many applications, a suitable approximation is applied that replaces a linearly anisotropic scattering operator by an isotropic scattering operator with approximate transport-corrected cross sections. One such approximation is the *outscatter method* [94]. In the outscatter method, the quantity $\bar{\mu}_0\Sigma_s$ is subtracted from both the total scattering cross section Σ_s and the diagonal elements of the $G\times G$ scattering matrix. Here, $\bar{\mu}_0$ is the mean cosine of the scattering angles. The resulting transport-corrected cross sections for the outscatter method are

$$\Sigma^{tr}_{s,g} = (1-\bar{\mu}_0)\Sigma_{s,g} = \Sigma_{s0,g} - \Sigma_{s1,g}, \tag{5.24a}$$

$$\Sigma^{tr}_{s,g'\rightarrow g} = \Sigma_{s0,g'\rightarrow g} \quad (\text{for } g \neq g'), \tag{5.24b}$$

$$\Sigma^{tr}_{s,g\rightarrow g} = \Sigma_{s0,g\rightarrow g} - \bar{\mu}_{0,g}\Sigma_{s0,g} = \Sigma_{s0,g\rightarrow g} - \Sigma_{s1,g}. \tag{5.24c}$$

The P_1 anisotropic scattering cross section is the sum of the differential scattering cross sections, namely:

$$\Sigma_{s1,g} = \sum_{g'}\Sigma_{s1,g'\rightarrow g}. \tag{5.25}$$

The *transport cross section* is now defined as:

$$\Sigma_{tr,g} = \Sigma_{t,g} - \Sigma_{s1,g}, \tag{5.26}$$

and it replaces the total cross section in Eq. (5.18). The outscatter method provides a suitable approximation for cross sections of heavier elements. For lighter elements, the *inscatter* or *inflow* correction is used, except for hydrogen, where the *neutron leakage conservation (NLC)* provides better accuracy. Like the outscatter method, these two methods attempt to approximate an anisotropic scattering operator with an isotropic scattering operator. More information regarding these methods can be found in a publication by Yee (2016) [96].

The use of transport-corrected cross sections converts an anisotropically scattering system into an approximate isotropically scattering system. This allows the source term of Eq. (5.19) to be written as follows:

$$q^{tr}_g = \frac{\chi_g}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}\phi_{g'} + \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma^{tr}_{s0,g'\rightarrow g}\phi_{g'}. \tag{5.27}$$

where $\phi_{g'}$ is the scalar flux computed as

$$\phi_g = \sum_{m'=1}^{M_N}\bar{\psi}_{i,g,m}\omega_m. \tag{5.28}$$

Thus, the use of transport-corrected cross sections eliminates the additional computational requirements necessary to treat the anisotropic scattering source explicitly.

5.4 Discretization of the Characteristics Equations

In Chapter 2, several different discretization methods were introduced for the different variables of the phase space. The discretization techniques for energy (the multigroup approximation) and angle will not be discussed in this section since they are not specific to MOC. The spatial discretization is more or less described in subsections 5.3.1 and 5.3.2. The focus of this section is on the specific discretization techniques required by MOC.

The fundamental discretization in the MOC is the representation of the flight paths (characteristic tracks) of the neutrons with a set of rays that traverse the problem domain. This is illustrated in Figure 5.2. The end goal is to determine the segment lengths from each ray that pass through each discrete region, which are then used as the variable $s_{k,i,m,n}$ in the evaluation of Eqs. (5.18), (5.20), and (5.21). The process of $s_{k,i,m,n}$ is commonly referred to as *ray-tracing*.

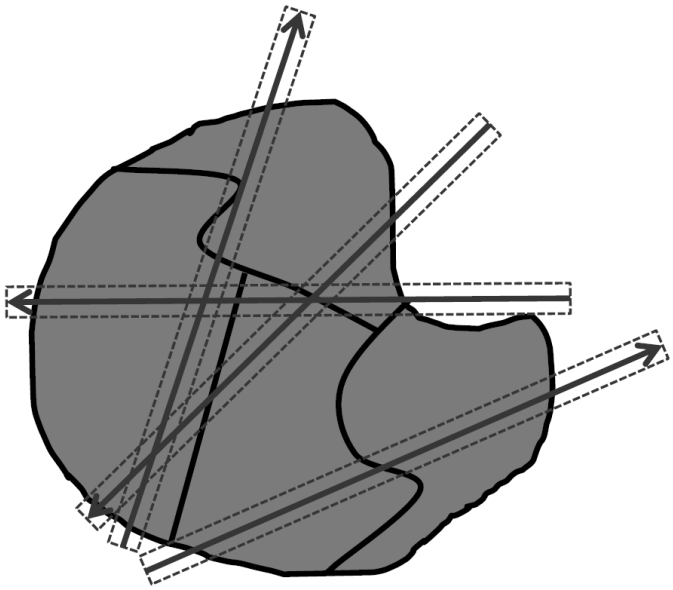


Figure 5.2. Characteristic rays intersecting a set of discrete spatial regions.

In general, for a ray-tracing algorithm, one may choose any set of rays as long as the intersection between a ray and the spatial region boundaries can be determined. Sufficient coverage of the problem domain by the rays is required for good accuracy in the solution. Several possible choices may be made in the design of the ray-tracing algorithm. Those choices specific to the algorithm in MPACT are discussed in the remainder of this section.

For LWR analysis, it is possible to take advantage of the fact that reactors generally have a high degree of regularity in their geometry when developing the ray-tracing algorithm. Considerable computational savings are possible by modeling only a small subdomain of the reactor that exhibits a unique geometry and then constructing ray-tracking information for the entire domain by replicating this information for the whole core. This technique, which has several names (cyclic ray-tracing, direct neutron path linking, modular ray-tracing), is herein referred to as *modular ray-tracing* and is illustrated in Figure 5.3, which has nine *ray-tracing modules* denoted as black squares. The *modular rays*, depicted as blue lines, are defined only within the ray-tracing module and connect at the ray-tracing module boundaries. The *long ray* shown by the red line extends through the entire problem domain and consists of a particular sequence of modular rays.

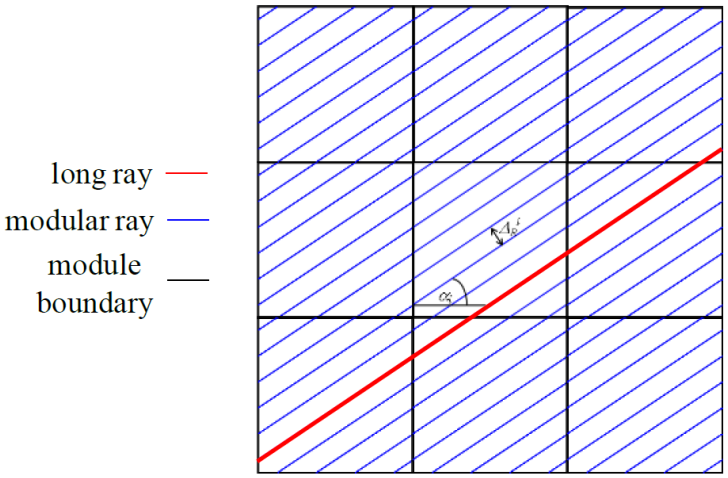


Figure 5.3. Modular ray-tracing concept in 2D.

The use of modular ray-tracing introduces new requirements regarding the choice of the angles, and it also creates other subtle issues. The first requirement for modular ray-tracing is that a structured grid can be overlaid on the problem geometry. For LWR problems, this is a Cartesian grid, and ideally, it will isolate the different, unique geometries of the subdomains. The next requirement is that for rays with a given angle, there must be an equal integer number of ray intersections on opposing surfaces of a ray-tracing module.

This second requirement is most commonly satisfied by using equally spaced rays within a given angle, although this need not be the case. Furthermore, when modular ray-tracing is used an additional step is required in the setup to determine connection information for the modular rays so that a long ray can be tracked across the entire problem domain. The computational advantages of modular ray-tracing can be considerable. If modular ray-tracing is not used, it can increase the storage requirements of the ray-tracing data by a factor of $O(10^7)$ for a full core pressurized water reactor (PWR) problem solved with 3D MOC. If quarter assembly modular ray-tracing is used, then the savings in memory requirements could be as much as $O(10^5)$ for a full core PWR for 3D MOC.

The next detail of the ray-tracing algorithm is the determination of the directions of flight that will be used for the rays. Since the discrete ordinates approximation described in Section 2.3 is used, it is logically consistent to use these as the directions of flight for the rays. To obtain accurate solutions, the discrete directions of flight should be obtained from a quadrature that minimizes the error of the quadrature approximation for the integration of functions of angle. However, quadratures that minimize the integration do not necessarily satisfy the modularity condition described previously. Thus, the approach in MPACT is to start from an existing angular quadrature than introduces small perturbations to the discrete angles so that they satisfy the modularity condition. This is analogous as finding the rational number closest to the quadrature direction that produces a ray spacing smaller than the one requested by the input. The quadrature weights are not updated once the new directions are chosen, so the possibility exists, especially for very coarse ray spacings, to introduce non-trivial errors into the angular quadrature/discretization, so care must be taken.

Once the directions of flight are chosen that satisfy the modularity conditions, it is necessary to set up the rays for a given angle. Each spatial region should be intersected by at least one ray from each angle, and in the ideal case, each region would be intersected by multiple rays from each angle. In MPACT, equally spaced rays are used to minimize the need to store ray-dependent quantities such as the cross sectional area data. Instead of storing the discrete cross sectional area of each characteristic ray in a problem, these data can simply be stored once for all rays or for all rays of a given angle. A potential disadvantage of using uniform rays, however, is that extra rays may be placed in regions that do not necessarily require them. This is illustrated in Figure 5.4 by having many rays near the diameter of the pin cell. One thicker ray would likely be sufficient here.

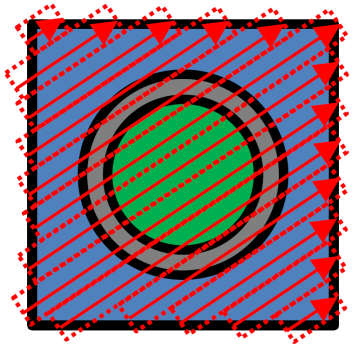


Figure 5.4. Uniformly spaced characteristic rays through a pin cell.

In addition to storing the cross sectional area of the rays, there is a choice of whether or not to store other ray-tracing data, such as the segment lengths and mapping of ray segment indices to region indices). This information can be computed on-the-fly as a ray is swept during the transport sweep, or it can be stored and accessed. In this case, a tradeoff between increased memory storage and repeated computations must be optimized. The choice that is made almost invariably for any 2D MOC implementation is to compute the ray-tracing information once and store it. However, for 3D MOC, some combination may be optimal. When storing the ray-tracing data, this has the benefit of

decoupling the sweep algorithm from the ray-trace algorithm, which allows each to be further optimized and developed independently. Another benefit of storing the ray-tracing data is that during a normal calculation, on the order of 1,000 sweeps can be performed, and each sweep may involve iterating over tens or hundreds of millions of rays, so even the slightest overhead from the repeated computation of the ray-tracing data will substantially increase the total computation time. However, the memory requirements for storage can become prohibitive, particularly for 3D, so other design choices for the algorithm may be made to address this issue in the future. Despite the large memory requirements for storing the ray tracing data, this is the current approach used in MPACT for 2D and 3D.

The final consideration in the MOC discretization involves integrals over volumes. This relates specifically to Eq. (5.20). For non-cartesian flat source regions, the integral (sum) of the segment volumes are an approximation of the region volume. This is shown in Figure 5.5. This is generally a problem of numerical integration stemming from the discrete

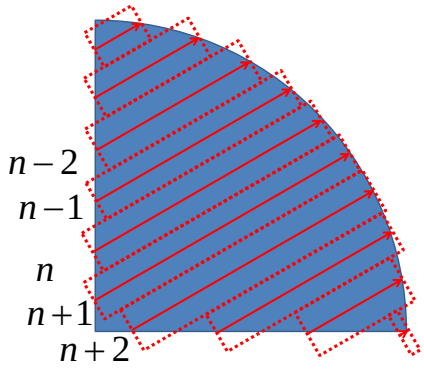


Figure 5.5. Numerical integration of a region volume by ray segments.

widths of the rays. As the limit of the ray spacing approaches zero, the sum of the segment volumes within the region will equal the region volume. However, this usually requires ray spacings that are too fine to be used for practical calculations, so an alternative approach is sought. In MOC codes, segment lengths within a given region are adjusted so that the region volumes are integrated exactly. This also sometimes referred to as *segment renormalization*. The segment renormalization can be done several ways. Two common approaches are to renormalize the segment lengths where (i) the renormalization is performed for each angle separately, (ii) the renormalization is computed for all angles. These renormalizations are given by Eqs. (5.29) and (5.30), respectively:

$$\bar{s}_{k,i,m,n} = s_{k,i,m,n} \left(\frac{V_i}{\sum_{n \in i} s_{k,i,m,n} \delta A_{k,m}} \right), \tag{5.29}$$

$$\bar{s}_{k,i,m,n} = s_{k,i,m,n} \left(\frac{4\pi V_i}{\sum_{m} \sum_{n \in i} s_{k,i,m,n} \omega_m \delta A_{k,m}} \right), \tag{5.30}$$

where V_i is the true volume of region i, $s_{k,i,m,n}$ is the length of the n^{th} ray in direction m and plane k, and $\delta A_{k,m}$ is the distance between rays (ray-spacing) in direction m. The renormalized segments, $\bar{s}_{k,i,m,n}$, are used in Eqs. (5.18), (5.20), and (5.21). Thus, the optical thicknesses of the regions are modified. While this approach is common, it is

certainly questionable in this regard. Note that in addition to renormalizing the segment lengths, no renormalization can be done, so there may be a slightly higher error from an inadequate spatial discretization.

Experience suggests that the normalization of Eq. (5.29) may be more sensitive to the ray spacing, especially if the problem contains thin annular spatial mesh regions. This problem arises because a specific angle might not have many tracks within a single region, resulting in a large volume correction. To address this issue, the renormalization can be performed on the integral (sum) over all angles, Eq. (5.30), rather than once for each angle, as in Eq. (5.29), or it can be neglected entirely. The Eq. (5.30) method helps remedy some of the sensitivities observed using the per-angle method. Furthermore, in problems with low void regions not performing any renormalization or using the renormalization of Eq. (5.29) may lead to calculation stability issues. The renormalization method of Eq. (5.30) appears to be the least sensitive to ray spacing for the approaches mentioned and may generally be the most robust. However, for spatially higher order source formulations, it does not preserve the particle balance. The default method in MPACT performs no volume correction and leaves the segment lengths unadjusted. In examining a subset of problems from the validation suite, not performing a volume correction was shown to provide the best results.

5.5 Iteration Schemes

This section describes the most basic iteration scheme for obtaining the MOC solution of the transport equation. The focus is primarily on how the MOC iteration is performed. The numerous other iteration schemes in MPACT are summarized in Chapter 3 and described in detail in other parts of this manual. However, this section describes the source iteration procedure, and then the implementation of MOC to obtain a solution. Lastly, the choice of convergence criteria to be used to determine when to stop iterating is described.

5.5.1 Source Iteration

The most basic iteration scheme for solving the transport equation is *source iteration*. As a result of the approximations described in Section 5.3, the primary iterative unknown is the scalar flux and not the angular flux. Source iteration for eigenvalue problems is often considered a two-level iteration scheme with an *inner* and *outer* iterations. The inner iteration assumes a fixed source and converges the self-scattering term, while the outer iteration performs an update of the eigenvalue. To facilitate this iteration scheme, the computation of the source is split into two parts:

$$\bar{q}^{tr}_{g,i} = \frac{1}{4\pi}\left(Q_{ext,i,g} + \Sigma^{tr}_{s0,i,g\rightarrow g}\phi_{i,g}\right), \tag{5.31}$$

where

$$Q_{ext,i,g} = \sum_{g'=1,g'\neq g}^{G} \Sigma^{tr}_{s0,i,g'\rightarrow g}\phi_{i,g'} + \frac{\chi_{i,g}}{k}\sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}. \tag{5.32}$$

In the 1-group fixed source problem, the known external source $Q_{ext,i,g}$ is the source from fission and *inscatter*, or scattering from group $g' \neq g$ to group g, and is a function of the scalar flux. Equation (5.31) is updated during the inner iteration, and Eq. (5.32) is updated during the outer iteration. The inner iteration for a 1-group problem is given by Algorithm 5.1.

Algorithm 5.1: Iterative algorithm for the MOC solution of 1-group fixed source problem (*inner iteration*)

1: Guess initial source.
2: **while** not converged **do**
3: Compute outgoing angular fluxes by evaluating Eq. (5.18) for all segments.
4: Compute segment-averaged angular fluxes by evaluating Eq. (5.21) for all segments.
5: Compute region-wise angular fluxes by evaluating Eq. (5.20) for all regions.
6: Compute scalar flux by evaluating Eq. (5.28) for all regions.
7: Update 1-group source by evaluating Eq. (5.30) for all regions.
8: **end while**

This first 4 steps of Algorithm 5.1 are typically functionalized and referred to as an *MOC kernel*. In functional form, this algorithm is presented as:

$$\left(\psi_g^{in,(l+1)}, \phi_g^{(l+1)}\right) = f\left(\psi_g^{in,(l)}, \bar{\mathbf{q}}_g^{(l)}\right), \tag{5.33}$$

In Eq. (5.33), $\psi_g^{in,(l)}$ is a vector of the discrete incoming angular flux boundary conditions in all space and angle for a single group g. $\bar{\mathbf{q}}_g^{(l)}$ is a vector of the group sources computed using Eq. (5.31) for all regions, $\phi_g^{(l)}$ is a vector of the scalar fluxes for all regions, and l is the inner iteration index. While this description assumes a 1-group MOC kernel, a multigroup kernel for solving this fixed source problem is also straightforward and more efficient, as will be discussed later in this chapter.

In Eq. (5.32), k_{eff} is the eigenvalue of the system and must also be determined as a part of the solution. This is traditionally calculated using the iterative *power method*, that is the basis of the *outer iteration*. The general form of the eigenvalue value problem in reactor physics can be written in operator notation as:

$$\mathbf{T}\phi = \frac{1}{k_{\text{eff}}}\mathbf{F}\phi, \tag{5.34}$$

where $\mathbf{F}$ represents the fission operator and $\mathbf{T}$ represents the streaming-plus-collision operator. The power method to solve Eq. (5.34) consists of the following iterative scheme:

$$\phi^{(\ell+1)} = \mathbf{T}^{-1}\frac{1}{k_{\text{eff}}^{(\ell)}}\mathbf{F}\phi^{(\ell)}, \tag{5.35}$$

$$k_{\text{eff}}^{(\ell+1)} = \frac{\|\mathbf{F}\phi^{(\ell+\mathbf{1})}\|_1}{\frac{1}{k_{\text{eff}}^{(\ell)}}\|\mathbf{F}\phi^{(\ell)}\|_1}. \tag{5.36}$$

For the outer iteration, denoted by the index ℓ, the total fission source is computed as shown in Eq. (5.37), and

Eq. (5.32) is rewritten as shown in Eq. (5.38).

$$\Psi_i^{(\ell)} = \frac{1}{k_{\text{eff}}^{(\ell)}} \sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}^{(\ell)}, \tag{5.37}$$

$$Q_{ext,i,g}^{(\ell)} = \chi_{i,g}^{(\ell)}\Psi_i^{(\ell)} + \sum_{g'=1,g'\neq g}^{G} \Sigma_{s0,i,g'\rightarrow g}\phi_{i,g'}^{(\ell)}. \tag{5.38}$$

The equation to update the eigenvalue based on the power method is shown in Eq. (5.39), where V_i is the region volume:

$$k_{\text{eff}}^{(\ell+1)} = \frac{\sum_{i=1}^{I} V_i \sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}^{(\ell+1)}}{\sum_{i=1}^{I} V_i\Psi_i^{(\ell)}}. \tag{5.39}$$

This overall iterative procedure for solving the eigenvalue problem is shown in Algorithm 5.2.

Algorithm 5.2: Iterative algorithm for the MOC solution of steady-state eigenvalue problem (*outer iteration*)

1: Guess initial k_{eff} and scalar flux.
2: **while** not converged **do**
3: Compute total fission source by evaluating Eq. (5.32) for all regions.
4: **for** all groups **do**
5: Compute source for group g by evaluating Eq. (5.31) for all regions.
6: Solve fixed source problem for group g with Algorithm 5.1.
7: **end for**
8: Update k_{eff} by evaluating Eq. (5.39) for all regions.
9: **end while**

The 1-group iteration scheme presented here has the advantage of reducing memory usage by allowing the transport method to only allocate data for a single group, with the exception of the boundary condition. In the loop over groups in Algorithm 5.2, the inscatter source of Eq. (5.32) is also updated in a Gauss-Seidel fashion. This helps to improve convergence for reactor problems because most LWRs are thermal reactors, and the physics of the slowing down source primarily involves the downscatter of neutrons.

However, in a multigroup formulation, where the scattering source is updated by a Jacobi iteration, approximately a 2x speed-up over a 1-group kernel is expected. Additionally, better stability properties for the transport correction used in the isotropic scattering approximation are observable.

5.5.2 Convergence Criterion for Source Iteration

In MPACT, the solution of the MOC equations are obtained iteratively. Therefore, criteria for convergence must be defined. MPACT primarily solves steady-state eigenvalue calculations, and for these types of calculations, two

convergence criteria are defined. First, the eigenvalue k_{eff} must be sufficiently converged. This is determined by comparing the estimates of k_{eff} from two consecutive iterations.

$$\left| k_{\text{eff}}^{(\ell)} - k_{\text{eff}}^{(\ell-1)} \right| < \varepsilon_{k_{\text{eff}}}. \tag{5.40}$$

Second, the shape of the flux must be sufficiently converged. This requirement is considered to be satisfied when the following inequality holds:

$$\sqrt{\frac{1}{N_{fi}} \sum_i \left[\sum_g \left(\frac{F_{i,g}^{(l)}}{\bar{F}^{(l)}} - \frac{F_{i,g}^{(l-1)}}{\bar{F}^{(l-1)}} \right) \right]^2} < \varepsilon_\phi. \tag{5.41}$$

Here, $\varepsilon_{k_{\text{eff}}}$ and ε_ϕ are user-specified constants (e.g., 10^{-4}), N_{fi} is the number of spatial cells in the fine mesh with fission, and

$$\bar{F}^{(\ell)} \equiv \sum_i \sum_g F_{i,g}^{(\ell)},$$

$$F_{i,g}^{(\ell)} \equiv \nu\Sigma_{f,i,g} \phi_{i,g}^{(\ell)}.$$

When these convergence criteria are satisfied, one can expect the error in k_{eff} to be $O(\varepsilon_{k_{\text{eff}}})$ and the root-mean-square (RMS) error in ϕ can be expected to be $O(\varepsilon_\phi)$.

5.5.3 MOC Sweep Procedure

There are several ways one can choose to loop over the segments in the MOC method. MPACT employs a *bidirectional sweep*, which loops over the angles, and within each angle, it loops over all the long rays. This type of sweep makes it easier to order all the associated data structures to ensure good cache coherency. If modular ray-tracing is performed, then the long ray can be constructed once and swept for both the forward and backward directions (e.g., a bidirectional sweep), further improving performance through better cache coherency. Figure 5.6 illustrates how the rays are be swept sequentially for a bidirectional sweep.

This type of sweep ordering requires increased memory storage when there are reflective or periodic boundary conditions as opposed to cyclic sweeping algorithms, and the convergence of the angular flux boundary condition will be more like Jacobi than Gauss-Seidel, especially for the spatial domain decomposed problem.

5.5.4 Gauss-Seidel and Jacobi Inscatter Sweeping Algorithms

This section presents the theory of the 2D MOC sweeping algorithms in MPACT is presented for two inscatter-source iteration schemes: Gauss-Seidel and Jacobi. In the Gauss-Seidel approach, as the MOC solver loops over groups, it uses the flux solution from the previous group to construct the inscatter source for the next group. Alternatively, the Jacobi approach uses only the fluxes from the previous outer iteration to determine the inscatter source for each group. Consequently, for the Jacobi iteration, the loop over groups can be moved from the outermost loop—as is the case

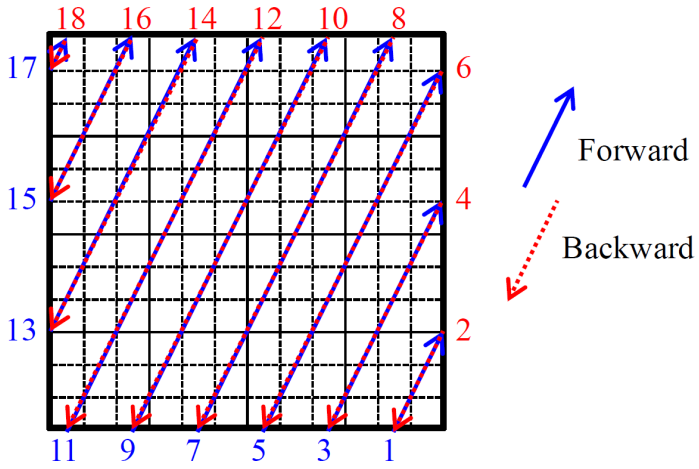


Figure 5.6. Sequential sweep algorithm.

with the Gauss-Seidel sweeper—to the innermost loop. This can substantially improve efficiency by minimizing the overhead of retrieving segment, region, and surface index information from the ray-tracing data.

Algorithm 5.3 outlines a more detailed description of the looping structure of a Gauss-Seidel transport sweep and the tasks performed by each one-group MOC kernel. It can be seen in this figure that the loop over energy groups is the outermost loop, and the source is updated as the loop progresses.

Algorithm 5.3: Pseudocode for Gauss-Seidel sweeping with group on outermost loop

for each group (g from 1 to N_{groups}) **do**
  Setup source for group g (using updated flux solution from previous groups)
  for each inner iteration (i from 1 to N_{inners}) **do**
    for each azimuthal angle (a from 1 to N_{angles}) **do**
      for each longray in angle a (l from 1 to $N_{longrays}(a)$) **do**
        Connect modular rays and determine coarse mesh surfaces
        Calculate exponential values for each segment in longray l and group g
        for each polar angle (p from 1 to N_{polar}) **do**
          Gather incoming angular flux ($\varphi^{in}_{l,p,g}$) at both ends of longray l
          for each segment (s from 1 to $N_{segments}(l)$) **do**
            Evaluate forward direction:
              Calculate outgoing angular flux ($\psi^{out}_{l,p,g}$)
              Tally contribution to fine mesh scalar flux in region $r1$ ($\phi_{g,r1}$)
            Evaluate backward direction:
              Calculate outgoing angular flux ($\psi^{out}_{l,p,g}$)
              Tally contribution to fine mesh scalar flux in region $r2$ ($\phi_{g,r2}$)
          end for
          Store outgoing angular flux ($\psi^{out}_{l,p,g}$)
          for each coarse mesh surface intersection (c from 1 to N_{cm}) **do**
            Tally surface flux and currents for coarse mesh surface c, group g
          end for
        end for
      end for
    end for
  end for
end for

Alternatively, the multigroup flux used to construct the inscatter sources can be lagged, creating a Jacobi-like algorithm in energy. It would be possible to implement this in a manner similar to the scheme outlined in Algorithm 5.3 by moving the source setup outside of the loops entirely. However, some performance can also be gained by moving the iteration over the groups g to the innermost loop (as shown in Algorithm 5.4). With the scheme in Algorithm 5.4, the work performed to connect the modular rays and coarse mesh surface mapping is only completed once for all groups. In contrast, the Gauss-Seidel approach requires that this work be performed once for each group. The potential performance benefits of the Jacobi algorithm yielding a multigroup kernel could be worthwhile as long as the time to set up the modular ray connections is a non-negligible fraction of the total runtime.

Algorithm 5.4: Pseudocode for Jacobi sweeping with group on innermost loop

```
Setup source for all groups
for each inner iteration (i from 1 to N_inners) do
  for each azimuthal angle (a from 1 to N_angles) do
    for each longray in angle a (l from 1 to N_longrays(a)) do
      Connect modular rays and determine coarse mesh surfaces
      Calculate exponential values for each segment in longray l and all groups
      for each polar angle (p from 1 to N_polar) do
        Gather incoming angular flux (φ^in_{l,p,g}) at both ends of longray l
        for each segment (s from 1 to N_segments(l)) do
          for each group (g from 1 to N_groups) do
            Evaluate forward direction:
              Calculate outgoing angular flux (ψ^out_{l,p,g})
              Tally contribution to fine mesh scalar flux in region r1 (φ_{g,r1})
            Evaluate backward direction:
              Calculate outgoing angular flux (ψ^out_{l,p,g})
              Tally contribution to fine mesh scalar flux in region r2 (φ_{g,r2})
          end for
        end for
        Store outgoing angular flux (ψ^out_{l,p,g})
        for each coarse mesh surface intersection (c from 1 to N_cm) do
          Tally surface flux and currents for all groups for coarse mesh surface c
        end for
      end for
    end for
  end for
end for
```

There is also the potential to sweep over the groups multiple times, particularly over the energy groups with upscatter cross section data. An additional loop in both figures would be included to account for this. In MPACT, the Jacobi sweep only uses the fluxes at the beginning of the outer iteration to update the sources, so subsequent "upscatter" iterations effectively perform more inner iterations over the thermal groups.

Though not shown here, the Jacobi approach does provide a substantial performance improvement, reducing the amount of time spent in MOC by roughly a factor of two [80]. Results also indicate that the Jacobi scheme also resolves many convergence issues observed with transport-corrected isotropic scattering (TCP_0), which had been a very

significant issue in MPACT for several years. The current hypothesis is that the Jacobi algorithm effectively under-relaxes the solution, allowing for a more controlled solution path, which is consistent with the relaxation approaches suggested by Tabuchi et al. [85].

6. 1D Axial Solution Methodology

Previous chapters presented the theory behind the 2D/1D method and the MOC used for the radial solvers in 2D/1D. This chapter discusses several methods used in the axial solvers in MPACT, which are used on a pin-wise basis in 2D/1D. Included are the nodal expansion method (NEM), the source expansion nodal method (SENM), the spherical harmonic (P_N) method, and the discrete ordinate (S_N) method.

The first part of the chapter discusses the primary governing equations and a comparison of one-node vs. two-node schemes. In MPACT, the NEM and SENM approaches both use two-node kernels, while the P_N and S_N methods leverage one-node kernels, so understanding both is important. In general, two-node approaches are preferred because of their increased stability [62], but one-node kernels are sometimes necessitated by the method.

The 2D MOC solvers use a flat or linear source/flux approximation because the fine mesh regions used to discretize each rod radially are relatively small, generally yielding ray segment lengths that are fractions of a centimeter. However, the axial discretization is usually in the 2–10 cm range, and a flat or linear approximation would yield significant errors. To mitigate this, nodal methods typically employ 2^{nd}–4^{th} Legendre expansions to accurately represent the intranodal spatial distributions of the source and flux. SENM, for example, goes a step further and incorporates two hyperbolic coefficients into the flux expansion, in addition to a quartic Legendre expansion.

The following sections summarize the primary equations and concepts. More in-depth derivations of each of these is available in the appendices of Stimpson's dissertation [82].

6.1 One-Dimensional Axial Governing Equations

6.1.1 Transport-Based

The axial equations can be derived in by averaging the three-dimensional transport equation, Eq. (2.23), radially over both x and y, as in Eq. (6.1), yielding Eq. (6.2). This leaves only a dependence in z, but with a radial transverse leakage in the source (6.2d):

$$\psi_{g,m}^{XY}(z,\alpha_m,\mu_m) = \frac{1}{A_{xy}}\int_{x_L}^{x_R}\int_{y_L}^{y_R}\psi_{g,m}(x,y,z,\alpha_m,\mu_m)dydx, \tag{6.1a}$$

$$\Sigma_{x,g}^{XY}(z) = \frac{\int_{x_L}^{x_R}\int_{y_L}^{y_R}\Sigma_{x,g}(x,y,z)\phi(x,y,z)dydx}{\int_{x_L}^{x_R}\int_{y_L}^{y_R}\phi(x,y,z)dydx}, \tag{6.1b}$$

$$\mu_m\frac{\partial}{\partial z}\psi_{g,m}^{XY}(z,\alpha_m,\mu_m) + \Sigma_{t,g}^{XY}(z)\psi_{g,m}^{XY}(z,\alpha_m,\mu_m) = \tilde{q}_{g,m}^{XY}(z,\alpha_m,\mu_m), \tag{6.2a}$$

$$\tilde{q}_{g,m}^{XY}(z,\alpha_m,\mu_m) = \overline{q}_{g,m}^{XY}(z,\alpha_m,\mu_m) + TL_{g,m}^{XY}(z,\alpha_m,\mu_m), \tag{6.2b}$$

$$\overline{q}_{g,m}^{XY}(z,\alpha_m,\mu_m) = \frac{\chi_g}{4\pi k_{\text{eff}}}\sum_{g'=1}^{N_g}\nu\Sigma_{f,g'}^{XY}(z)\phi_{g'}^{XY}(z) + \frac{1}{4\pi}\sum_{g'=1}^{N_g}\Sigma_{s,g'\to g}^{XY}(z)\phi_g^{XY}(z), \tag{6.2c}$$

$$\begin{aligned} TL_{g,m}^{XY}(z,\alpha_m,\mu_m) = -\frac{\sqrt{1-\mu_m^2}}{A_{xy}}\Bigg\{ & cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}(x_R,y,z,\alpha_m,\mu_m) - \psi_{g,m}(x_L,y,z,\alpha_m,\mu_m)\Big)dy \\ & + sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}(x,y_R,z,\alpha_m,\mu_m) - \psi_{g,m}(x,y_L,z,\alpha_m,\mu_m)\Big)dx\Bigg\}. \end{aligned} \tag{6.2d}$$

Here, α_m and μ_m are the azimuthal and cosine of the polar angles corresponding to angle m, $\psi_{g,m}^{XY}$ is the radially integrated angular flux for group g and angle m, $TL_{g,m}^{XY}$ is the corresponding radial tranvserse leakage, and A_xy is the radial cross sectional area of each node.

Though the discrete ordinates approximation has already been made, α_m and μ_m are included in the angular dependence to make the angular integrations more clear. In reality, these integrals are replaced with weighted summations over the discrete ordinates. Additionally, because of the separability assumed between the flux and cross sections, the homogenized cross sections are obtained using flux and area weighting, though the fission spectrum is homogenized using the fission source instead of the flux. For completeness, the total cross section should be angle-dependent and homogenized using the angular flux, but a common approximation is to also homogenize it using the scalar flux. This is done to save storage of both the angle-dependent cross sections and angular fluxes.

A variant of this approach is to integrate Eq. (6.2) azimuthally, Eq. (6.3), so that the equations maintain only polar dependence. This formulation was used by Hursin in previous work with DeCART [47, 33]:

$$\mu_m\frac{\partial}{\partial z}\psi_{g,m}^{XY,\alpha}(z,\mu_m) + \Sigma_{t,g}^{XY}(z)\psi_{g,m}^{XY,\alpha}(z,\mu_m) = \tilde{q}_{g,m}^{XY}(z,\mu_m), \tag{6.3a}$$

$$\tilde{q}_{g,m}^{XY,\alpha}(z,\mu_m) = \overline{q}_{g,m}^{XY,\alpha}(z,\mu_m) + TL_{g,m}^{XY,\alpha}(z,\mu_m), \tag{6.3b}$$

$$\overline{q}_{g,m}^{XY}(z,\mu_m) = \frac{\chi_g}{2k_{\text{eff}}}\sum_{g'=1}^{N_g}\nu\Sigma_{f,g'}^{XY}(z)\phi_{g'}^{XY}(z) + \frac{1}{2}\sum_{g'=1}^{N_g}\Sigma_{s,g'\to g}^{XY}(z)\phi_g^{XY}(z), \tag{6.3c}$$

$$TL_{g,m}^{XY,\alpha}(z,\mu_m) = -\frac{\sqrt{1-\mu_m^2}}{A}\int_0^{2\pi}\Bigg\{cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}(x_R,y,z,\alpha_m,\mu_m)-\psi_{g,m}(x_L,y,z,\alpha_m,\mu_m)\Big)dy \\ +sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}(x,y_R,z,\alpha_m,\mu_m)-\psi_{g,m}(x,y_L,z,\alpha_m,\mu_m)\Big)dx\Bigg\}d\alpha, \tag{6.3d}$$

where

$$\psi_{g,m}^{XY,\alpha}(z,\mu_m) = \int_0^{2\pi}\psi_{g,m}^{XY}(z,\alpha_m,\mu_m)d\alpha. \tag{6.4}$$

The leakage term in Eq. (6.3d) can be also be averaged polarly to remove angular dependence for use in the azimuthally integrated equations, Eqs. (6.3). The next subsection, which covers the diffusion-based axial solver, uses isotropic radial transverse leakage, in which case Eq. (6.2d) has no angular dependence, as in Eq. (6.5):

$$TL_{g,m}^{XY}(z) = -\frac{1}{2A}\int_{-1}^{1}\sqrt{1-\mu_m^2}\int_0^{2\pi}\Bigg\{cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}(x_R,y,z)-\psi_{g,m}(x_L,y,z)\Big)dy \\ +sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}(x,y_R,z)-\psi_{g,m}(x,y_L,z)\Big)dx\Bigg\}d\alpha d\mu. \tag{6.5}$$

All three of the transverse leakage equations – Eqs. (6.2d), (6.3d), and (6.5) – can be averaged over z, as in Eq. (6.6), to use the angular fluxes from the radial solvers, Eq. (6.7):

$$\psi_{g,m}^{Z}(z,\alpha_m,\mu_m) = \frac{1}{h_z}\int_{h_B}^{h_T}\psi_{g,m}(x,y,z,\alpha_m,\mu_m)dz, \tag{6.6}$$

where h_z is the axial height of the node at plane z.

$$TL_{g,m}^{XY}(z,\alpha_m,\mu_m) = -\frac{\sqrt{1-\mu_m^2}}{A}\Bigg\{cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}^{Z}(x_R,y,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x_L,y,\alpha_m,\mu_m)\Big)dy \\ +sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}^{Z}(x,y_R,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x,y_L,\alpha_m,\mu_m)\Big)dx\Bigg\}, \tag{6.7a}$$

$$TL_{g,m}^{XY,\alpha}(z,\mu_m) = -\frac{\sqrt{1-\mu_m^2}}{A}\int_0^{2\pi}\Bigg\{cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}^{Z}(x_R,y,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x_L,y,\alpha_m,\mu_m)\Big)dy \\ +sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}^{Z}(x,y_R,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x,y_L,\alpha_m,\mu_m)\Big)dx\Bigg\}d\alpha, \tag{6.7b}$$

$$TL_{g,m}^{XY,\alpha,\mu}(z) = -\frac{1}{2A}\int_{-1}^{1}\sqrt{1-\mu_m^2}\int_0^{2\pi}\Bigg\{cos(\alpha_m)\int_{y_L}^{y_R}\Big(\psi_{g,m}^{Z}(x_R,y,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x_L,y,\alpha_m,\mu_m)\Big)dy + sin(\alpha_m)\int_{x_L}^{x_R}\Big(\psi_{g,m}^{Z}(x,y_R,\alpha_m,\mu_m)-\psi_{g,m}^{Z}(x,y_L,\alpha_m,\mu_m)\Big)dx\Bigg\}d\alpha d\mu. \tag{6.7c}$$

6.1.2 Diffusion-Based Solvers

The diffusion-based axial sweepers (NEM/SENM) are formulated by radially averaging the three-dimensional diffusion equation. The details of obtaining the diffusion equation from the transport equation are deferred to Chapter 7 to avoid duplication, but it is still important to at least present the 1D diffusion equation to better understand the extensions made from that.

$$-D_g\frac{d^2\phi_g(z)}{dz^2}+\Sigma_{r,g}(z)\phi_g(z)=\frac{\chi_g(z)}{k_{\text{eff}}}\sum_{g'=1}^{N_g}\nu\Sigma_{f,g'}(z)\phi_{g'}(z)+\frac{1}{4\pi}\sum_{g'=1,g'\neq g}^{N_g}\Sigma_{s,g'\rightarrow g}(z)\phi_{g'}(z)-TL_g^{XY}(z), \tag{6.8a}$$

$$TL_g^{XY}(z)=\frac{1}{h_x}\Big(J_{L,x,g}(z)-J_{R,x,g}(z)\Big)+\frac{1}{h_y}\Big(J_{L,y,g}(z)-J_{R,y,g}(z)\Big). \tag{6.8b}$$

Here, D_g is the diffusion coefficient for group g, $\Sigma_{r,g}$ is the removal cross section (total minus self-scatter), and J is the current. Since the radial surfaces on which the currents exist span the entire plane, they effectively do not have an axial dependence, so the leakage is more consistent with Eq. (6.9):

$$TL_g^{XY}(z)=\frac{1}{h_x}\Big(J_{L,x,g}^{Z}-J_{R,x,g}^{Z}\Big)+\frac{1}{h_y}\Big(J_{L,y,g}^{Z}-J_{R,y,g}^{Z}\Big). \tag{6.9}$$

This is numerically equivalent to Eq. 6.7c.

6.2 One-Node vs. Two-Node

As previously mentioned, what MPACT defines as a node is simply a single axial mesh region of a single rod (fuel, guide tube, control, or burnable poison rod). Two types of nodal kernels, which are the most basic component of a solver, are primarily used in MPACT: one-node and two-node. In each of these kernels, the underlying physics is the same and is determined by the flux and source expansions being used. However, the boundary conditions and constraints differ in important ways. As a preference, the discussion and illustrations here relate to one- and two-node systems along the x-axis (as opposed to z-axis for axial solvers). Therefore, when one refers to a "left" or "right" node, as will be seen in the figures, it is directly related to a "bottom" or "top" node if considering the same along the z-axis.

In the one-node formulation (Figure 6.1), the incoming partial currents (or angular fluxes for transport-based kernels) are the prescribed boundary conditions, and the outgoing partial currents and flux distribution are output.

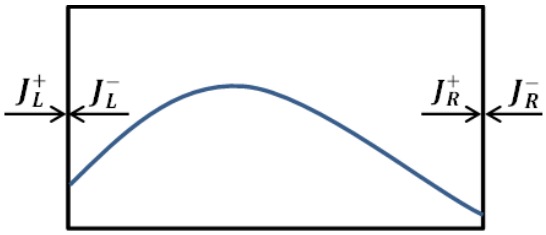


Figure 6.1. One-node kernel.

With the two-node kernels (Figure 6.2), the mesh-averaged scalar fluxes are used as a constraint, and the net current at the interface and the flux distributions in both nodes are output. Additionally, the two-node kernels own two instances of the intranode flux distribution (one for the left interface and one for the right interface). During the iteration process, the two intranodal distributions will not necessarily agree, but they will agree at convergence.

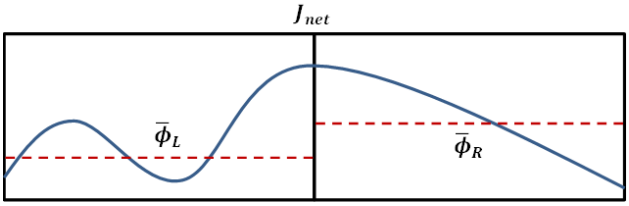


Figure 6.2. Two-node kernel.

The fact that the two-node kernels do not allow the mesh-average scalar flux (zeroth moment) to change is beneficial and considerably more stable. When incorporating the transverse leakage terms, which will be discussed in more detail later, it is possible to encounter negative sources, that can drive the one-node kernels to non-physical negative fluxes, whereas the two-node kernels can handle these more easily.

Unfortunately, two-node formulations are currently only available for the diffusion-based kernels, such as NEM and SENM. If such a formulation were applied to the transport-based kernels, P_N or S_N, mesh-averaged angular fluxes would need to be preserved, which are not available with the existing 2D/1D scheme. However, it is possible that a two-node transport kernel could be used in something, such as the 2D/3D scheme, in which 2D-MOC and 3D-S_N are coupled [98].

Some exploratory work has been performed in MPACT to develop a *hybrid* P_3 kernel that combines characteristics of the one- and two-node solvers using a two-node formulation for the zeroth angular moment and a one-node formulation for the second angular moment [84]. This capability has demonstrated some usefulness, but it has been observed to encounter some convergence issues in certain situations.

There are also other alternatives. The nTRACER code, for example, solves an entire axial rod in a single linear system [49]. This technique provides a tighter coupling between the nodes, and the CMFD solve and can require fewer iterations with the one-node kernels, but it requires more parallel communication.

6.3 Nodal Expansion Method (NEM)

In the nodal expansion method, the source is expanded with quadratic Legendre polynomials and the flux is expanded with quartic polynomials [36], where ξ denotes the normalized spatial variable:

$$Q(\xi) = \sum_{i=0}^{2} q_i P_i(\xi), \tag{6.10a}$$

$$\phi(\xi) = \sum_{i=0}^{4} a_i P_i(\xi). \tag{6.10b}$$

The coefficients (a_i) are determined from the 0^{th} through 2^{nd} moment balance equations:

$$\int_{-1}^{1} P_n(\xi)\left(-\Sigma_D \frac{d^2}{d\xi^2}\phi(\xi) + \Sigma_r \phi(\xi) - Q(\xi)\right) d\xi = 0, \tag{6.11a}$$

$$\Sigma_D = \frac{4D}{h^2}, \tag{6.11b}$$

where D is the diffusion coefficient, h is the size of the mesh, and Σ_r is the removal cross section. This is used in conjunction with the flux and current continuity enforcement:

$$\phi_1(1) = \phi_2(-1), \tag{6.11c}$$

$$J_1(1) = J_2(-1). \tag{6.11d}$$

The node-averaged flux can be preserved by forcing the zeroth moment value in the flux expansion to be the scalar flux, as in Eqs. (6.12):

$$a_{1,0} = \overline{\phi}_1, \tag{6.12a}$$

$$a_{2,0} = \overline{\phi}_2. \tag{6.12b}$$

During an iteration, the flux coefficients are used to construct the source coefficients (q_i) for the next iteration. Each internal two-node problem solves an 8×8 linear system relating the information in the two node being simulated, as

in Eq. (6.13). Here, 1 and 2 indicate the indices of the two nodes:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -\frac{2D_1}{h_1} & -3\frac{2D_1}{h_1} & -6\frac{2D_1}{h_1} & -10\frac{2D_1}{h_1} & \frac{2D_2}{h_2} & -3\frac{2D_2}{h_2} & 6\frac{2D_2}{h_2} & -10\frac{2D_2}{h_2} \\ & -3\Sigma_{D1} & & -10\Sigma_{D1} & & & & \\ \Sigma_{r1} & & -15\Sigma_{D1} & & & & & \\ & \Sigma_{r1} & & -35\Sigma_{D1} & & & & \\ & & & & & -3\Sigma_{D2} & & -10\Sigma_{D2} \\ & & & & \Sigma_{r2} & & -15\Sigma_{D2} & \\ & & & & & \Sigma_{r2} & & -35\Sigma_{D2} \end{bmatrix} \begin{bmatrix} \phi_{1,1} \\ \phi_{1,2} \\ \phi_{1,3} \\ \phi_{1,4} \\ \phi_{2,1} \\ \phi_{2,2} \\ \phi_{2,3} \\ \phi_{2,4} \end{bmatrix} = \begin{bmatrix} -\overline{\phi}_1 + \overline{\phi}_2 \\ 0 \\ q_{1,0} - \Sigma_{r1}\overline{\phi}_1 \\ q_{1,1} \\ q_{1,2} \\ q_{2,0} - \Sigma_{r2}\overline{\phi}_2 \\ q_{2,1} \\ q_{2,2} \end{bmatrix}. \tag{6.13}$$

While more explicit details of how this matrix is formed can be found in Stimpson's dissertation [82], it is useful to provide some understanding of how Eqs. (6.11) are used. For example, the top row corresponds to the flux continuity equation, as shown in Eq. (6.11c), and the second row corresponds to the current continuity using $J = -D\frac{d^2\phi(z)}{dz^2}$. The next three rows correspond to the 0^{th} through 2^{nd} moment balance equations per Eq. (6.11a) for the bottom node and the final three for the top node.

Boundary nodes are solved in a slightly different manner since one of the nodes would technically be outside the system. In these cases, special 4×4 one-node linear systems are formulated, taking into account the boundary condition as appropriate:

$$\begin{bmatrix} (\frac{D}{h}+\alpha) & (3\frac{D}{h}+\alpha) & (6\frac{D}{h}+\alpha) & (10\frac{D}{h}+\alpha) \\ & -3\Sigma_D & & -10\Sigma_D \\ \Sigma_r & & -15\Sigma_D & \\ & \Sigma_r & & -35\Sigma_D \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -\overline{\phi}_1 + \overline{\phi}_2 \\ q_0 - \Sigma_{r1}\overline{\phi}_1 \\ q_1 \\ q_2 \end{bmatrix}, \tag{6.14}$$

where α is the albedo boundary condition at the system boundary.

A single sweep with the NEM solver involves looping over the pin-wise coarse mesh cells, generally solving for two separate two-node problems: one in which the current cell is the lower cell of the two nodes, and one in which it is the upper cell. For boundary cells, one of these two-node solves is replaced with a call to the boundary kernel instead.

With two-node kernels, the cell-averaged scalar flux is a constraint and does not change, so multiple inner iterations do not change any part of the solution; therefore, only one inner iteration is executed.

6.4 Source Expansion Nodal Method (SENM)

With SENM, both the source, Eq. (6.15a), and the flux, Eq. (6.15b), use a quartic Legendre expansion. However, the flux also has two additional hyperbolic terms [97]:

$$Q(\xi) = \sum_{i=0}^{4} q_i P_i(\xi), \tag{6.15a}$$

$$\phi(\xi) = A\sinh(\kappa\xi) + B\cosh(\kappa\xi) + \sum_{i=0}^{4} a_i P_i(\xi), \tag{6.15b}$$

$$\kappa = \frac{h}{2}\sqrt{\frac{\Sigma_r}{D}}. \tag{6.15c}$$

In the previous set of equations, the flux has a homogeneous solution – $A\sinh(\kappa\xi) + B\cosh(\kappa\xi)$ – and a particular solution – $\sum_{i=0}^{4} a_i P_i(\xi)$. The particular solution coefficients can be determined by solving the 0^{th} through 4^{th} order moment balance equations. The homogeneous coefficient B for each node is given from the node-averaged flux and zeroth moment particular coefficient a_0:

$$B = \frac{\kappa}{\sinh(\kappa)}\left(\overline{\phi} - a_0\right). \tag{6.16}$$

Having solved the particular coefficient (a_i) and homogeneous B coefficients, the homogeneous A coefficients for the two nodes, A_1 and A_2, can then be calculated by enforcing flux and current continuity at the interface of the two nodes, as in Eqs. (6.11c) and (6.11d).

Unlike with the NEM formulation, the SENM solver does not set up a linear system to determine the unknown flux coefficients. Instead, the particular coefficients are solved first since they only depend on the source moments (q_i). Then the cosh coefficients (B) are found, and both a_i and B are used to determine the sinh coefficients (A). Also, the flux expansion is projected onto a quartic Legendre expansion (without hyperbolic coefficients) when formulating the source moment values. However, the SENM sweeping strategy is very similar to NEM.

6.5 Spherical Harmonics (P_N)

With this method, the angular flux, which has been azimuthally integrated for a 1D representation, is assumed to have a Legendre expansion with respect to the cosine of the polar angle (μ) [70]:

$$\psi_g(x,\mu) = \sum_{m=0}^{N_{mom}} \frac{2m+1}{2}\psi_{m,g}(x)P_m(\mu). \tag{6.17}$$

By substituting Eq. (6.17) into the transport equation, multiplying by $P_n(\mu)$, and integrating over μ, Eq. (6.18a) can be obtained, which describes the relationship between each of the angular moments:

$$\frac{d}{dx}\left[\frac{n}{2n+1}\psi_{n-1,g}(x) + \frac{n+1}{2n+1}\psi_{n+1,g}(x)\right] + \Sigma_{t,g}(x)\psi_n(x,g) = \Sigma_{sn,g\to g}\psi_{n,g}(x) + Q_g(x)\delta_{n,0}. \tag{6.18a}$$

It also yields the corresponding boundary conditions of (Eqs. (6.18b) and (6.18c)) for the left and right interfaces, respectively:

$$\sum_{m=0}^{N_{mom}} \frac{2m+1}{2}\left(\int_{-1}^{0} \mu P_n(\mu)P_m(\mu)d\mu\right)\psi_{m,g}(0) = \int_0^1 \mu P_n(\mu)\psi_g^b(\mu)d\mu, \tag{6.18b}$$

$$\sum_{m=0}^{N_{mom}} \frac{2m+1}{2}\left(\int_{-1}^{0} \mu P_n(\mu)P_m(\mu)d\mu\right)\psi_{m,g}(X) = \int_{-1}^0 \mu P_n(\mu)\psi_g^b(\mu)d\mu. \tag{6.18c}$$

In MPACT, P_3 and P_5 have been implemented and can be formulated to wrap the one-node NEM kernel by normalizing spatially (ξ) and rearranging the boundary conditions slightly. With this setup, P_N handles the angular distribution, and NEM handles the spatial distribution. Once the equations for the 0^{th} through 3^{rd} moments have been found, the first and third can be solved and substituted into the 0^{th} and 2^{nd} moment equations. These can be further simplified into Eqs. (6.19) and (6.20), which show the final equations for the P_3 kernel:

Zeroth-Moment

$$-\frac{4D_{0,g}}{h^2}\frac{d^2}{d\xi^2}\Phi_{0,g}(\xi) + (\Sigma_{t,g} - \Sigma_{s0,g\to g})\Phi_{0,g}(\xi) = Q_g(\xi) + 2(\Sigma_{t,g} - \Sigma_{s0,g\to g})\Phi_{2,g}(\xi), \tag{6.19a}$$

$$-\frac{D_{0,g}}{h}\frac{d}{d\xi}[\Phi_{0,g}(-1)] + \frac{1}{4}\Phi_{0,g}(-1) = \int_0^1 \mu\psi_{L,g}^b(\mu)d\mu + \frac{3}{16}\Phi_{2,g}(-1), \tag{6.19b}$$

$$\frac{D_{0,g}}{h}\frac{d}{d\xi}[\Phi_{0,g}(1)] + \frac{1}{4}\Phi_{0,g}(1) = \int_{-1}^0 |\mu|\psi_{R,g}^b(\mu)d\mu + \frac{3}{16}\Phi_{2,g}(1). \tag{6.19c}$$

Second-Moment

$$\begin{aligned} -\frac{4D_{2,g}}{h^2}\frac{d^2}{d\xi^2}[\Phi_{2,g}(\xi)] + \left(\frac{9}{5}\Sigma_{t,g} - \frac{4}{5}\Sigma_{s0,g\to g}\right)\Phi_{2,g}(\xi) \\ = -\frac{2}{5}\left(Q_g(\xi) - (\Sigma_{t,g} - \Sigma_{s0,g\to g})\Phi_{0,g}(\xi)\right), \end{aligned} \tag{6.20a}$$

$$-\frac{D_{2,g}}{h}\frac{d}{d\xi}[\Phi_{2,g}(-1)] + \frac{1}{4}\Phi_{2,g}(-1) = \frac{3}{5}\int_0^1 P_3(\mu)\psi_{L,g}^b(\mu)d\mu + \frac{3}{80}\Phi_{0,g}(-1) - \frac{1}{80}\Phi_{2,g}(-1), \tag{6.20b}$$

$$\frac{D_{2,g}}{h}\frac{d}{d\xi}[\Phi_{2,g}(1)] + \frac{1}{4}\Phi_{2,g}(1) = \frac{3}{5}\int_{-1}^0 P_3(\mu)\psi_{R,g}^b(\mu)d\mu + \frac{3}{80}\Phi_{0,g}(1) - \frac{1}{80}\Phi_{2,g}(1), \tag{6.20c}$$

where

$$\Phi_{0,g}(\xi) = \psi_{0,g}(\xi) + 2\psi_{2,g}(\xi), \tag{6.21a}$$

$$\Phi_{2,g}(\xi) = \psi_{2,g}(\xi), \tag{6.21b}$$

$$D_{0,g} = \frac{1}{3\Sigma_{tr,g}}, \tag{6.21c}$$

$$D_{2,g} = \frac{9}{35\Sigma_{t,g}}. \tag{6.21d}$$

The sweeping strategy for this solver is considerably different than NEM/SENM in some respects. First, since a one-node kernel does not use the node-averaged scalar flux as a constraint, this quantity is permitted to use it and does change during the iteration sequence. The incoming boundary conditions based on the partial surface flux moments will also change during iteration. For these reasons, multiple inner iterations are performed to resolve these before updating the sources in the next group, and several loops over groups are also performed.

When possible, the spatial sweeping algorithm is also modified to loop over nodes axially before beginning the solve for the next rod. As a node is solved, outgoing partial surface flux moments are also determined which can then be used to inform the incoming boundary conditions for the nodes around it. Generally, an axial sweep both up and down is used to resolve the boundary conditions in each inner iteration.

Before proceeding to the next section on the S_N axial solver, some clarification on the nomenclature of the P_N solver will be covered. Historically, P_N methods within the context of 2D/1D have been referred to as simplified P_N (SP_N) [48, 49]. To remain consistent with these applications, many MPACT publications before 2017 also refer to these methods as SP_N. However, despite the historical context, referring to these as SP_N has caused confusion at conferences and other meetings. Additionally, it could be misleading since the 2D/1D approximation does not preserve the 3D SP_N approximation when using these axial solvers. To avoid further confusion, these are now being referred to as P_N or NEM-P_N because NEM is used to handle the spatial representation of the flux and source.

6.6 Discrete Ordinates (S_N)

Similar principles to those used to derive MOC can be applied to the characteristics-based discrete ordinates equations to formulate axial kernels. Instead of using a Legendre expansion for the polar angle dependence as with P_N, the polar angles are discretely represented. However, there are several options in how to represent the azimuthal dependence. The most basic option is to assume there is no dependence, in which case the equations are azimuthally integrated, as was the case with P_N. The most detailed option is to explicitly represent every azimuthal angle, but this approach is memory-intensive. An alternative is to approximate the azimuthal dependence using a Fourier series.

6.6.1 Spatial Moments

Instead of using an NEM kernel to account for the spatial representation as was done with P_N, the spatial distributions are handled using a direct Legendre expansion. Currently, options are available to use a 0^{th} through 3^{rd} order expansion, but it is recommended to only use the 3^{rd} order representation. By applying similar moment balance equations to the transport equation, the outgoing and average angular flux for the j^{th} moment is decsribed by Stimpson [82] (Eqs. 4.3 and 4.5). Because these formulas can be fairly complex, Eqs. (6.22a) and (6.22b) are simply written in terms of

condensed functions:

$$\psi_{out} = \psi_{in} e^{-\tau} + \sum_{i=0}^{N_{mom}} q_i f_{out,i}(h,\mu,\Sigma_t), \tag{6.22a}$$

$$\overline{\varphi}_j = \psi_{in} f_{j,in}(h,\mu,\Sigma_t) + \sum_{i=0}^{N_{mom}} q_i f_{j,i}(h,\mu,\Sigma_t), \tag{6.22b}$$

where $f_{out,i}$, $f_{j,in}$, $f_{j,i}$ are the condensed functions, dependent upon the node height (h), cosine of the polar angle (μ), and the total cross section (Σ_t). τ is defined as the optical thickness of the characteristic ray segment ($\tau = l\Sigma_t$), where l is the ray segment length.

To better understand what these condense functions look like, consider a problem with only a linear source. In this case, the outgoing angular flux could be specified as:

$$\psi_{out} = \varphi(1) = \psi_{in} e^{-\tau} + q_0 \frac{1-e^{-\tau}}{\Sigma_t} + q_1 \frac{\left(\tau(1+e^{-\tau}) + 2e^{-\tau} - 2\right)}{\tau\Sigma_t}, \tag{6.23}$$

and the 0^{th} and 1^{st} moment average angular fluxes would be given by:

$$\varphi_0 = \psi_{in} \frac{1-e^{-\tau}}{\tau} + q_0 \frac{(e^{-\tau}-1)+\tau}{\tau\Sigma_t} + q_1 \frac{\left(2-2e^{-\tau}-\tau(1+e^{-\tau})\right)}{\tau^2\Sigma_t}, \tag{6.24a}$$

$$\begin{aligned}\varphi_1 = \psi_{in} \frac{6-6e^{-\tau}-\tau(3e^{-\tau}+3)}{\tau^2} + q_0 \frac{3\left(2e^{-\tau}-2+\tau(1+e^{-\tau})\right)}{\tau^2\Sigma_t} \\ + q_1 \frac{\left(12(1-e^{-\tau}) - 12\tau e^{-\tau} - 3(1+e^{-\tau})\tau^2 + \tau^3\right)}{\tau^3\Sigma_t}.\end{aligned} \tag{6.24b}$$

A full description of the condensed functions up through a cubic source are available in Stimpson's dissertation [82].

6.6.2 Azimuthal Moments

Equation (6.25) is the radially integrated transport equation with explicit azimuthal and polar dependence, which is similar to Eq. (6.2), but considers continuous angular variables:

$$\mu \frac{\partial \psi_g^{XY}(z,\alpha,\mu)}{\partial z} + \Sigma_{t,g} \psi_g^{XY}(z,\alpha,\mu) = \frac{Q_g^{XY}(z)}{4\pi} + TL_g^{XY}(z,\alpha,\mu), \tag{6.25a}$$

$$\begin{aligned}TL_g^{XY}(z,\alpha,\mu) = -\frac{\sqrt{1-\mu^2}}{A_{xy}} \left(\int_{y_l}^{y_r} \cos(\alpha) \Big(\psi_g(x_r,y,z,\alpha,\mu) - \psi_g(x_l,y,z,\alpha,\mu)\Big) dy \right. \\ \left. + \int_{x_l}^{x_r} \sin(\alpha) \Big(\psi_g(x,y_r,z,\alpha,\mu) - \psi_g(x,y_l,z,\alpha,\mu)\Big) dx \right).\end{aligned} \tag{6.25b}$$

From this equation, the azimuthal dependence of the angular fluxes and transverse leakages are assumed to be represented by a Fourier expansion, Eq. (6.26):

$$\psi_g^{XY}(z,\alpha,\mu) = \frac{1}{2\pi}\psi_{g,0}^{XY}(z,\mu) + \frac{1}{\pi}\sum_{m=1}^{N_{mom}}\left(\psi_{g,sm}^{XY}(z,\mu)\sin(m\alpha) + \psi_{g,cm}^{XY}(z,\mu)\cos(m\alpha)\right), \tag{6.26a}$$

$$TL_g^{XY}(z,\alpha,\mu) = \frac{1}{2\pi}TL_{g,0}^{XY}(z,\mu) + \frac{1}{\pi}\sum_{m=1}^{N_{mom}}\left(TL_{g,sm}^{XY}(z,\mu)\sin(m\alpha) + TL_{g,cm}^{XY}(z,\mu)\cos(m\alpha)\right). \tag{6.26b}$$

Substituting these into Eq. (6.25a) and integrating over α, the zeroth moment equation is obtained as Eq. (6.27a):

$$\mu\frac{\partial}{\partial z}\psi_{g,0}^{XY}(z,\mu) + \Sigma_{t,g}\psi_{g,0}^{XY}(z,\mu) = \frac{Q_g^{XY}(z)}{2} + TL_{g,0}^{XY}(z,\mu). \tag{6.27a}$$

Similarly, multiplying by $\sin(n\alpha)$ or $\cos(n\alpha)$ and integrating over α, the equations for the sine and cosine moments are obtained respectively:

$$\mu\frac{\partial}{\partial z}\psi_{g,sn}^{XY}(z,\mu) + \Sigma_{t,g}\psi_{g,sn}^{XY}(z,\mu) = TL_{g,sn}^{XY}(z,\mu), \tag{6.27b}$$

$$\mu\frac{\partial}{\partial z}\psi_{g,cn}^{XY}(z,\mu) + \Sigma_{t,g}\psi_{g,cn}^{XY}(z,\mu) = TL_{g,cn}^{XY}(z,\mu). \tag{6.27c}$$

This derivation yields moment equations that are independent of one another. This differs from the P_N formulation, in which each moment is dependent on the neighboring moments. However, in the P_N formulation, an expansion of μ is assumed, and when substituting into the axial streaming term, which also has a factor of μ, the subsequent integration produces coupled equations. Because the Fourier expansion only depends on the azimuthal angle (α), the additional μ has no impact on the integration.

Additionally, the radial MOC sweeper tallies the transverse leakage coefficients – $TL_{g,0}^{XY}(z,\mu)$, $TL_{g,sn}^{XY}(z,\mu)$, and $TL_{g,cn}^{XY}(z,\mu)$ – using the following equations:

$$TL_{g,0}^{XY}(z,\mu) = \sum_{l=1}^{N_{azi}} w_l TL_{g,m}^{XY}(z,\alpha_m,\mu), \tag{6.28a}$$

$$TL_{g,sn}^{XY}(z,\mu) = \sum_{l=1}^{N_{azi}} \sin(n\alpha_m) w_l TL_{g,m}^{XY}(z,\alpha_m,\mu), \tag{6.28b}$$

$$TL_{g,cn}^{XY}(z,\mu) = \sum_{l=1}^{N_{azi}} \cos(n\alpha_m) w_l TL_{g,m}^{XY}(z,\alpha_m,\mu), \tag{6.28c}$$

where w_l is the angle weight in the quadrature set.

The sweeping algorithm for S_N is very similar to that of P_N/S_N, where wavefront axial sweeping is employed to propogate the boundary conditions of each node effectively, and multiple inner iterations are necessary to converge the boundary conditions.

6.7 Transverse Leakage Interpolation

In the 2D/1D equations, the angular fluxes and currents that come from the radial sweeper have no intranodal axial dependence. However, since all of the axial sweepers use some form of Legendre expansion of the spatial moments, they can incorporate higher order transverse leakage components into the source construction using the transverse leakages from the neighboring planes. To accomplish this, higher order coefficients can be constructed to describe the shape of the transverse leakage in each axial node, as illustrated in Figure 6.3.

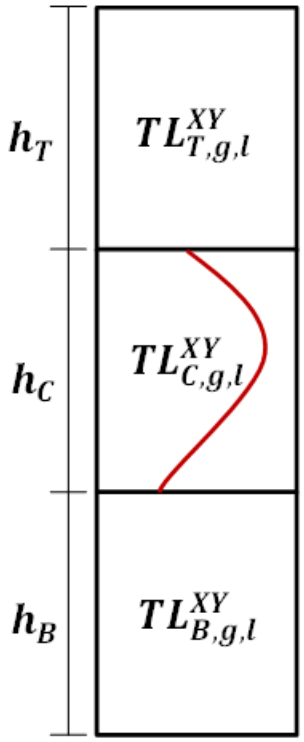


Figure 6.3. Tranverse leakage interpolation.

Eqs. (6.29) show the coefficients for a quadratic interpolation of the transverse leakage using the values from the top and bottom neighboring planes:

$$TL^{XY}_{g,m}(\xi) = \sum_{i=0}^{2} TL^{XY}_{g,l,i} P_i(\xi), \tag{6.29a}$$

$$G = 2(h_C + h_B)(h_C + h_T)(h_B + h_C + h_T), \tag{6.29b}$$

$$TL^{XY}_{g,l,0} = TL^{XY}_{C,g,l}, \tag{6.29c}$$

$$\begin{aligned} TL^{XY}_{g,l,1} = G^{-1} h^c \big[&(TL^{XY}_{T,g,l} - TL^{XY}_{C,g,l})(h_C + 2h_B)(h_C + h_B) \\ &- (TL^{XY}_{B,g,l} - TL^{XY}_{C,g,l})(h_C + 2h_T)(h_C + h_T) \big], \end{aligned} \tag{6.29d}$$

$$TL^{XY}_{g,l,2} = G^{-1}(h_C)^2\left[(TL^{XY}_{T,g,l} - TL^{XY}_{C,g,l})(h_C + h_B) + (TL^{XY}_{B,g,l} - TL^{XY}_{C,g,l})(h_C + h_T)\right]. \tag{6.29e}$$

It is feasible to use cubic or even quartic expansions, but this would require data from farther neighboring planes and potentially more data passing between processors.

It has been shown that the quadratic interpolation can introduce larger errors in cases with more severe axial profiles [10], particularly when using a refined axial mesh. It is likely that higher order expansions would alleviate these errors, but the results from that work suggest that using a flat interpolation is more physical and that a linear fit would also show improvement. However, there are some differences in the quadratic interpolation schemes between that work and what is shown here which can likely avoid these issues.

7. Coarse Mesh Finite Difference Acceleration

An important aspect of efficiently solving the transport equation is the use of an effective technique to accelerate the iterative convergence of a sweep-based method. The goal is to minimize the amount of computational work needed to reach convergence. This is usually achieved by minimizing the number of transport sweeps required to converge. Typically, a good acceleration scheme has the following properties:

- it does not change the solution from that of the un-accelerated (sweep-based) iteration scheme,
- it converges rapidly as compared to the un-accelerated iteration scheme,
- the cost per iteration is not significantly higher than the cost per iteration of the unaccelerated scheme, and
- it is not dependent on a particular discretization or mesh.

One acceleration method that exhibits these qualities is the coarse mesh finite difference (CMFD) method, originally developed to accelerate the convergence of nodal diffusion problems in reactor analysis [78]. The fundamental concept of CMFD applies also to the transport equation and has been shown to be effective at accelerating the convergence of 2D MOC transport problems. In general, CMFD can be understood as a nonlinear coarse mesh diffusion synthetic acceleration (DSA) scheme. [61]

The conventional CMFD method is described in Section 7.1. In Section 7.2, an extension of the CMFD method for spatially decomposed problems is presented. Sections 7.3 and 7.4 address *artificially diffusive* and *optimized* CMFD, which are optimized using a Fourier analysis. Section 7.5 discusses the performance of CMFD for 3D problems using 2D/1D and 3D MOC. In Section 7.8, CMFD is discussed using red-black successive over-relaxation. Section 7.6 describes subplane CMFD, and Section 7.7 presents the convergence criteria used in MPACT to determine when to discontinue performing power iterations.

7.1 Conventional CMFD

The lower order equation used in the CMFD acceleration scheme is based on the following multigroup diffusion equation:

$$-\nabla\cdot(D_g(\mathbf{x})\nabla\phi_g(\mathbf{x}))+\Sigma_{t,g}(\mathbf{x})\phi_g(\mathbf{x})=\left[\sum_{g'=1}^{G}\left(\Sigma_{s0,g'\rightarrow g}(\mathbf{x})+\frac{\chi_g}{k_{\text{eff}}}\nu\Sigma_{f,g'}(\mathbf{x})\right)\phi_{g'}(\mathbf{x})\right]. \tag{7.1}$$

This equation can be obtained by applying Fick's Law to the neutron balance equation that results from integrating the neutron transport equation over angle. In this process, the neutron balance equation is exact with respect to the transport equation; it is the introduction of Fick's Law that makes Eq. (7.1) an approximation.

The discretized neutron balance equation on cell j is given by

$$\sum_s J^{\text{net}}_{j,g,s}A_{j,s}+\Sigma_{t,j,g}\phi_{j,g}V_j=\left[\sum_{g'=1}^{G}\left(\Sigma_{s0,j,g'\rightarrow g}+\frac{\chi_g}{k_{\text{eff}}}\nu\Sigma_{f,j,g'}\right)\phi_{j,g'}\right]V_j, \tag{7.2}$$

where the subscript j is a spatial cell index, and s denotes a surface of spatial cell j. $A_{j,s}$ is the area of surface s of the cell j, and V_j is the cell volume. All of the above quantities with subscript j are averaged over the volume of cell j except for $J^{\text{net}}_{j,g,s}$, which is an area-averaged quantity over surface s of cell j.

CMFD also introduces the coarse mesh concept on which the diffusion equation is solved. This requires development of restriction and prolongation transfer operators for the solution between the fine mesh defined by the MOC flat source regions, and the CMFD coarse mesh. The restriction (homogenization) operator collapses the fine mesh solution onto the coarse mesh and is shown below:

$$\Sigma_{x,j,g}=\frac{\sum_{i\in j}\Sigma_{x,i,g}\phi_{i,g}V_i}{\sum_{i\in j}\phi_{i,g}V_i}, \tag{7.3a}$$

$$\phi_{j,g}=\frac{\sum_{i\in j}\phi_{i,g}V_i}{\sum_{i\in j}V_i}, \tag{7.3b}$$

$$\chi_{j,g}=\frac{\sum_{i\in j}\chi_{i,g}[\sum_{g'=1}^{G}\nu\Sigma_{f,i,g'}\phi_{i,g'}]V_i}{\sum_{i\in j}[\sum_{g'=1}^{G}\nu\Sigma_{f,i,g'}\phi_{i,g'}]V_i}. \tag{7.3c}$$

In classic diffusion theory, the net current is approximated by Fick's Law with a finite-difference spatial discretization. This is shown in Eqs. (7.4), where $h_{j,s}$ is the distance between cell j and its neighboring cell on surface s:

$$J^{\text{net,diff}}_{j,g,s}=-\tilde{D}_{j,g,s}\left(\phi_{j,g}-\phi_{j,g,s}\right), \tag{7.4a}$$

$$\tilde{D}_{j,g,s}=\frac{2D_{j,g}D_{j_s,g}}{h_{j,s}\left(D_{j,g}+D_{j_s,g}\right)}. \tag{7.4b}$$

Here the subscript j_s is the index of the cell bordering cell j via surface s. The diffusion coefficient of cell j is defined via the transport-corrected cross section as follows:

$$D_{j,g}=\frac{1}{3\Sigma_{tr,j,g}}. \tag{7.5}$$

Σ_{tr} is defined in Section 5.3.3.

CMFD introduces a correction coefficient, $\hat{D}_{j,g,s}$, so the expression for the net current is given by:

$$J^{\text{net}}_{j,g,s} = -\tilde{D}_{j,g,s}\left(\phi_{j,g} - \phi_{j,g,s}\right) + \hat{D}_{j,g,s}\left(\phi_{j,g} + \phi_{j,g,s}\right). \tag{7.6}$$

The correction factor, $\hat{D}_{j,g,s}$, introduced in Eq. (7.6), is defined by Eq. (7.7), where the fine mesh transport method determines the neutron net current on the surfaces of the coarse mesh:

$$\hat{D}_{j,g,s} = \frac{J^{\text{net}}_{j,g,s} + \tilde{D}_{j,g,s}\left(\phi_{j,g} - \phi_{j,g,s}\right)}{\left(\phi_{j,g} + \phi_{j,g,s}\right)}. \tag{7.7}$$

This correction factor, along with cross section homogenization, creates equivalence between the solution of the fine mesh MOC equations and the coarse mesh diffusion equations at convergence. The homogenization process preserves all the cell-volume integrated reaction rates based on the fine mesh solution. The correction factor of Eqs. (7.6) and (7.7) allows the low order system to also preserve the cell-surface integrated quantities of the fine mesh solution, and specifically the average leakage. Because of this equivalence, the multiplication factor k_{eff} of the CMFD linear system is the same as that of the fine mesh transport method computed from source iteration upon convergence.

The iterative solution algorithm with CMFD is described in Algorithm 7.1, in which the superscript l denotes the iteration index. Because the CMFD problem is an eigenvalue problem, the prolongation equation for the scalar flux

Algorithm 7.1: Iterative algorithm for the MOC solution of steady-state eigenvalue problem

1: **while** not converged **do**

2: Compute cell-averaged values for CMFD coefficients from Eqs. (7.3) and (7.6):
$$\left\{\phi_{j,g}^{(l+\frac{1}{2})}, \Sigma_{x,j,g}, \chi_{j,g}, \tilde{D}_{j,g,s}, \hat{D}_{j,g,s}\right\} \leftarrow f_{hom}\left(\phi_{i,g}^{(l+\frac{1}{2})}, \Sigma_{x,i,g}, \chi_{i,g}\right)$$

3: Solve CMFD eigenvalue problem given by Eqs. (7.2) for cell-averaged scalar flux and k_{eff}:
$$\phi_{j,g}^{(l+1)} \leftarrow f_{\text{CMFD}}\left(\phi_{j,g}^{(l+\frac{1}{2})}, \Sigma_{x,j,g}, \chi_{j,g}, \tilde{D}_{j,g,s}, \hat{D}_{j,g,s}\right)$$

4: Update fine mesh solution given by Eq. (7.8) and (7.9):
$$\phi_{g}^{(l+1)} \leftarrow f_{\text{pro}}\left(\phi_{g}^{(l+1)}, \phi_{g}^{(l+\frac{1}{2})}\right)$$

5: Perform transport sweep:
$$\left\{\phi_{g}^{(l+\frac{1}{2})}, \varphi_{g}^{in,(l+\frac{1}{2})}\right\} \leftarrow f_{\text{transport}}\left(\varphi_{g}^{in,(l)}, \phi_{g}^{(l)}\right)$$

6: Update fine-mesh fission source

7: Check if solution is converged

8: **end while**

takes a nonlinear form given by:

$$\phi_{i,g}^{(l+1)} = \phi_{i,g}^{(l+\frac{1}{2})} \frac{\phi_{j,g}^{(l+1)}}{\phi_{j,g}^{(l+\frac{1}{2})}},\ i \in j. \tag{7.8}$$

7.2 Spatial Domain Decomposed CMFD

When considering numerical methods for solving the spatially decomposed transport equation, algorithms that preserve the iteration of a serial sweep such as the popular KBA algorithm [58, 4, 3, 5] have limited scalability. However, another way to perform the parallel sweep with better scalability is to take a Jacobi-type approach and allow each subdomain to sweep independently. The downside of this is that the angular flux boundary condition for interior sub-domains becomes lagged, thus reducing the rate of convergence for an un-accelerated source iteration scheme. If a 1-group problem is considered in a semi-infinite 1D purely absorbing medium with a fixed boundary source, then the problem will converge in a single transport sweep in serial. However, if this problem is divided spatially into n_{space} domains, it would take $n-1$ sweeps for the n^{th} sub-domain to receive the correct boundary condition from its neighbor.

In conventional CMFD, the prolongation operator only applies to the cell-averaged scalar flux. SDD-CMFD extends the prolongation operator to also provide an update to the angular flux boundary condition on the decomposed spatial subdomains. This update equation can take a number of forms, so it is written as:

$$\varphi_{i,g,m,k}^{in,(l+1)} = \varphi_{i,g,m,k}^{in,(l)} f_{j,g,s}^{(l+1)},\ k \in s,\ i \in j. \tag{7.9}$$

In MPACT, the form of f is taken to be:

$$f_{j,g,s}^{(l+1)} = \frac{\phi_{j,g,s}^{S,(l+1)}}{\phi_{j,g,s}^{S,\left(l+\frac{1}{2}\right)}}. \tag{7.10}$$

Here, $\phi_{j,g,s}^{S}$ is the surface-averaged scalar flux for surface s. Previous work [83] compared several definitions of f and showed that Eq. (7.10) provides reasonable additional speed-up and guarantees positivity of the update factor. To obtain the coarse mesh surface fluxes required by Eq. (7.10), additional radial coupling coefficients must be introduced that reconstruct the transport surface flux from the coarse mesh cell-averaged flux. These are given below and can be derived in a similar way as the current coupling coefficients:

$$\phi_{j,g,s}^{S} = \tilde{s}_{j,g,s}\phi_{j,g} - \left(1-\tilde{s}_{j,g,s}\right)\phi_{j,g,s} + \hat{s}_{j,g,s}\left(\phi_{j,g}+\phi_{j,g,s}\right), \tag{7.11a}$$

$$\tilde{s}_{j,g,s} = \frac{1}{1+h_{j,s}\frac{\Sigma_{tr,j,g}}{\Sigma_{tr,j,g,s}}}, \tag{7.11b}$$

$$\hat{s}_{j,g,s} = \frac{\phi_{j,g,s}^{S,MOC} - \tilde{s}_{j,g,s}\phi_{j,g} + \left(1-\tilde{s}_{j,g,s}\right)\phi_{j,g,s}}{\left(\phi_{j,g}+\phi_{j,g,s}\right)}. \tag{7.11c}$$

Here, Σ_{tr} is the transport-corrected total cross section defined in Section 5.3.3. It is expected that SDD-CMFD will have similar performance as conventional CMFD, although the importance of the boundary angular flux update component of the prolongation operator increases as the optical thickness of the spatial subdomain decreases. In practice, the convergence penalty for lagging the sub-domain interface angular fluxes for a Jacobi-type parallel sweep is largely eliminated by SDD-CMFD acceleration.

7.3 Artificially Diffusive CMFD

The CMFD acceleration method is known to become unstable when the maximum coarse-cell optical thickness becomes larger than a few mean-free-paths (mfp) thick. The spectral radius predicted by Fourier analysis of a CMFD-accelerated k-eigenvalue 1D S_N sweep using a step characteristic spatial discretization is depicted in Fig. 7.1. (The spectral radius ρ is the asymptotic error-reduction per iteration. If $\rho \ll 1$, then the method is rapidly convergent, if $\rho < 1$, then the method is convergent, and if $\rho \geq 1$, then the method is divergent. Also, 1D S_N is a good proxy for 2D and 3D MOC; the Fourier analysis of the 2D step characteristic method yields results similar to those for 1D.)

In Fig. 7.1, the convergence rate of CMFD (spectral radius) rapidly degrades as the optical thickness of a spatial cell increases from 0.5 to 2.0 mfp. Above 2.0 mfp, standard CMFD is unstable. The spectral radius for the CMFD variant equivalent to partial-current CMFD (pCMFD) is shown in red. The pCMFD-like variant is unconditionally stable. This method consists of adding a constant factor of $\frac{1}{4}$ to the standard CMFD diffusion coefficient $\tilde{D}$:

$$\tilde{D}_{g,s} = \frac{2}{3\left(\Sigma_{tr,g,s^-} h_{s^-} + \Sigma_{tr,g,s^+} h_{s^+}\right)} + \frac{1}{4}, \tag{7.12}$$

where s is the surface index, and the subscripts s^- and s^+ denote the cells on the negative and positive side of the surface, respectively. When the optical thickness is very small, $\tilde{D}_s >> 1$, and the factor of $\frac{1}{4}$ has little effect on the overall iteration. When the optical thickness is moderate, between 0.3–1.0 mfp, this factor degrades the convergence rate of the iteration because the lagged $\hat{D}$ terms are relatively large, due to a poor diffusion coefficient, thus requiring more iterations to converge. However, when the optical thickness is greater than 1 mfp, this scheme will not become

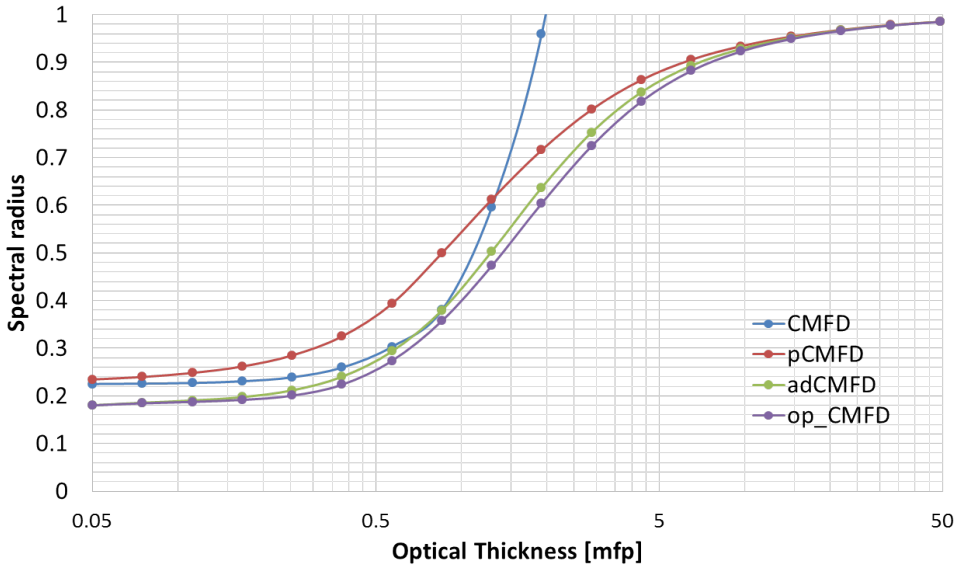


Figure 7.1. Fourier analysis convergence behavior of CMFD-accelerated transport method for CMFD variants (1D IHM S_N, 1 transport sweep per CMFD update, 3 fine cells per coarse cell).

unstable like standard CMFD. The scheme has the effect of damping the flux updates, which prevents oscillatory growth in the error modes that causes divergence. As the optical thickness becomes very large, the diffusion coefficients become a uniform $\frac{1}{4}$ across the problem, and the CMFD-accelerated scheme converges at a rate similar to a source iteration scheme. Overall, the pCMFD-like scheme is favorable because of its guaranteed stability. However, CMFD can converge slightly faster than pCMFD when the spatial cells are moderately thick and stability is not an issue.

The artificially diffusive CMFD (adCMFD) method, shown in green in Fig. 7.1, effectively combines the best of CMFD and pCMFD convergence properties to produce a stable iteration scheme that converges as fast or faster than standard CMFD [99]. This is done by adding a variable amount to the diffusion coefficient on a surface, based on the optical thickness of its neighboring cells:

$$\tilde{D}_{g,s} = \frac{2}{3\left(\Sigma_{tr,g,s-}h_{s-} + \Sigma_{tr,g,s+}h_{s+}\right)} + \theta_{g,s}. \tag{7.13}$$

The variable θ in Eq. (7.13) will be no greater than $\frac{1}{4}$, and in practice it will always be positive. In Fig. 7.1, it is allowed to be negative, which is why the green line is lower than the blue line for small optical thicknesses. Equation (7.13) demonstrates the relationship between adCMFD and pCMFD. The actual implementation of adCMFD is slightly different:

$$\tilde{D}_{g,s} = \frac{2}{3\left(\frac{\Sigma_{tr,g,s-}h_{s-}}{1+3\theta_{g,s-}} + \frac{\Sigma_{tr,g,s+}h_{s+}}{1+3\theta_{g,s+}}\right)}, \tag{7.14}$$

where the optimum θ is a piecewise function of the optical thickness, with upper and lower bounds and a 6th order polynomial between the limits:

$$\theta_g = \begin{cases} 0, & \Sigma_{tr,g}h < 1, \\ \sum_{k=0}^{6} a_k \left(\Sigma_{tr,g}h\right)^{k+1}, & 1 \leq \Sigma_{tr,g}h \leq 14, \\ 0.254\Sigma_{tr,g}h, & \Sigma_{tr,g}h > 14. \end{cases} \tag{7.15}$$

The coefficients a_k are defined by a fit to the optimal θ calculated experimentally by Fourier analysis. Negative values of θ are not allowed because they could give a negative diffusion coefficient $\tilde{D}_{g,s}$, which would likely cause a stability issue.

7.4 Optimal CMFD

In Fig. 7.1, the final curve (purple) shows the spectral radius of *optimal CMFD*. All of the CMFD variants that modify the diffusion coefficient can be expressed in terms of a shaped or prolonged CMFD flux update. For example, a flux update factor Λ for a given coarse cell can be defined as:

$$\Lambda_{j,g} = \frac{\phi_{j,g}^{l+1}}{\phi_{j,g}^{l+\frac{1}{2}}}. \tag{7.16}$$

This can be applied to each fine cell i in coarse cell j, or the update can be given a fine mesh shape f, so that

$$\Lambda_{i,g} = f_i \Lambda_{j,g}, i \in j, \tag{7.17a}$$

$$\frac{1}{V_j} \sum_{i \in j} f_{i,g} V_i = 1. \tag{7.17b}$$

The curve in Fig. 7.1 was generated by running many different cases while varying f for a 1D S_N problem, with 3 uniform fine cells per coarse cell, to find the optimal f. This gives a good indication of the theoretical limit of the convergence rate of the CMFD method. Without fundamentally restructuring the method, a faster convergence rate likely cannot be achieved. Fortunately, the adCMFD method is very close to the apparent limit, and the adCMFD θ is inexpensive to calculate.

The shape of the optimal curve in Fig. 7.1 shows that it is not possible to optimize CMFD to greatly improve the convergence rate over adCMFD for optically thick cells. The only way to improve the convergence rate is to reduce the optical thickness of the coarse cells, which moves the thicknesses toward the left on the plot, thus down the slope of the curve to a lower spectral radius.

7.5 3D MOC CMFD Acceleration

CMFD becomes unstable when accelerating transport methods, but not diffusion-type methods (the 1D P_N kernels). Thus, only the optical thickness of the coarse cells in the x and y dimensions will cause CMFD to be unstable or adCMFD to converge slowly, if using a 1D P_N kernel. When using 2D/1D, the axial height of the MOC planes can become arbitrarily large without affecting CMFD convergence properties.

3D MOC introduces an extra complication to the CMFD convergence properties. Because 3D MOC is a transport method in all three spatial dimensions, the optical thickness in the axial direction also matters. The convergence rate of adCMFD will become very slow as the optical thicknesses of the coarse cells become large (several mfp). When using 2D/1D, the axial heights are often larger than the pin pitch. Thus, if the coarse mesh for 3D MOC is defined with the same axial heights typically used for 2D/1D, the iterations may converge very slowly due to large axial optical thickness. While the size of the coarse mesh in the radial dimension is usually fixed to conform to the periodic Cartesian geometry of an LWR, the axial coarse mesh can be varied more easily. The coarse mesh used for 3D MOC should have similar sizes in each dimension (roughly cubic) to avoid excessive degradation of the convergence rate.

2D/1D with a 1D S_N axial solver can also have CMFD-related stability problems. However, by default the 1D S_N axial solver has five inner sweeps, as well as five up-scatter iterations. Using this many transport sweeps will probably mitigate any CMFD instability unless the optical thickness is extremely large. There are still stability issues related to transverse leakage source positivity, but these are not directly related to classic CMFD instability.

7.6 Subplane CMFD

Using conventional CMFD, each pin cell is homogenized using the quantities defined in Eq. (7.3) in every slice in the model. The radial coupling coefficients defined in Eq. (7.7) are obtained by calculating the current at the interface between each pair of pin cells using the transport sweeper, while the axial coupling coefficients are obtained from the axial currents calculated by the axial solve during the previous iteration.

To ensure stability and minimize runtime of the 2D/1D method, it is desirable to use a small number of thick MOC planes. However, to achieve sufficient accuracy, this axial mesh must sometimes be refined. To do this while avoiding stability issues or significantly increasing the computational burden of the calculations, researchers at the Korea Atomic Energy Research Institute developed the subplane scheme [14, 74]. While preparing the CMFD and axial calculations, this scheme divides the MOC planes into multiple subplanes. This axially refines the CMFD and 1D transport meshes to allow them to capture intra-plane detail without increasing the number MOC planes required. This efficiently produces a more accurate 2D/1D solution without the expense or stability concerns normally associated with 2D/1D mesh refinement. This section explains the modifications to the 2D/1D calculations required for the subplane scheme.

7.6.1 Homogenization

For the traditional CMFD calculations, each pin cell is homogenized into a single CMFD cell. When using the subplane scheme, the homogenized pin cell is divided axially into a stack of cells. This causes the CMFD system to have more cells in the axial direction than the transport mesh, allowing CMFD to capture subplane axial flux shapes that would otherwise be ignored. To do this, a subplane scaling factor is introduced that provides an axial shape within a 2D plane:

$$c_{g,i}^{(l)} = \frac{\phi_{g,i}^{(l-1)}}{\phi_{g,i}^{(l-1)}} = \frac{\phi_{g,i}^{(l-1)} \sum_{i'=1}^{N_{sp}} V_{i'}}{\sum_{i'=1}^{N_{sp}} \phi_{g,i'}^{(l-1)} V_{i'}}, \tag{7.18}$$

where superscripts indicate from which iteration the values are taken, and N_{sp} is the number of subplanes for the pin cell of interest. When the homogenized values are calculated from the 2D transport solution using Eq. (7.3), the fine mesh flux is multiplied by this subplane scaling factor everywhere it appears. It is assumed that materials are constant axially within each pin cell, so this subplane factor has no impact on the homogenized cross sections. However, the homogenized flux $\phi_{g,i}$ and fission source distribution $\chi_{g,i}$ are changed, providing an axial shape for the source term in the eigenvalue calculation.

7.6.2 Coupling Coefficients

In addition to the homogenized cell terms, the coupling coefficients described by Eqs. (7.4) and (7.7) must be calculated for each subplane. To maintain consistency, the area-averaged current calculated by the transport sweep must be preserved across the sub-surfaces used by the subplane scheme. Thus, the current calculated by the transport sweep at an interface is used at the corresponding interfaces for all subplanes in that plane. To maintain consistency, the cell-homogenized flux used in the calculation of the diffusion coefficients must be defined for the entire MOC plane as in Eq. (7.3) rather than using the subplane scaling factor for each subplane.

The axial coupling coefficient can be treated in a more straightforward manner. Because the 1D axial solvers use the same pin-homogenized mesh as the CMFD solver, axial currents are naturally calculated at the top and bottom of each subplane. Thus, these currents can be used together with the subplane fluxes to calculate subplane-dependent axial coupling coefficients.

7.6.3 Projection

The projection of the CMFD flux back to the 2D planes must also account for the presence of the subplanes. To do this, the solution is volume-averaged over all subplanes for each pin cell, resulting in an equation similar to (7.8):

$$\phi^{(k)}_{\text{trans},g,j} = \left(\frac{\sum_{i'=1}^{N_{sp}} \phi^{(k)}_{\text{CMFD},g,i'} V'_i}{\sum_{i'=1}^{N_{sp}} \phi^{(k-1)}_{\text{CMFD},g,i'} V'_i} \right) \phi^{(k-1)}_{\text{trans},g,j}. \tag{7.19}$$

The surface fluxes can also be homogenized axially in the same way to scale the angular flux boundary conditions using Eq. (7.9).

7.7 Solving the CMFD Eigenvalue Problem

7.7.1 Power Iteration

Algorithm 7.1 describes the overall iteration scheme in MPACT, in which CMFD is used to accelerate the transport sweeper. However, the details of solving the CMFD eigenvalue problem were omitted in this algorithm. This section describes the power iteration procedure (with Wielandt shift) for solving the CMFD eigenvalue problem.

Without a Wielandt shift, a single power iteration on Eq. (7.1) can be described as follows:

$$[-\nabla \cdot D_g(\mathbf{x})\nabla + \Sigma_{t,g}(\mathbf{x})]\,\phi_g^{(l+1)}(\mathbf{x}) - \sum_{g'=1}^{G} \Sigma_{s0,g'\to g}(\mathbf{x})\phi_{g'}^{(l+1)}(\mathbf{x}) = \frac{\chi_g}{k_{\text{eff}}^{(l)}} F^{(l)}(\mathbf{x}), \tag{7.20a}$$

$$F^{(l+1)}(\mathbf{x}) = \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\mathbf{x})\phi_{g'}^{(l+1)}(\mathbf{x}), \tag{7.20b}$$

$$k_{\text{eff}}^{(l+1)} = k_{\text{eff}}^{(l)} \frac{\int F^{(l+1)}(\mathbf{x})F^{(l+1)}(\mathbf{x})d^3\mathbf{x}}{\int F^{(l+1)}(\mathbf{x})F^{(l)}(\mathbf{x})d^3\mathbf{x}}. \tag{7.20c}$$

The domain of integration for the integrals in Eq. (7.20c) is the entire spatial domain of the problem. The details of CMFD spatial discretization and the correction factors ($\hat{D}$) have been omitted for simplicity. These details do not affect the content in this subsection and can be found in the previous sections of this chapter.

In matrix notation, Eqs. (7.20) can be rewritten as follows:

$$M\underline{\phi}^{(l+1)} = \frac{1}{k_{\text{eff}}^{(l)}} F\underline{\phi}^{(l)}, \tag{7.21a}$$

$$k_{\text{eff}}^{(l+1)} = k_{\text{eff}}^{(l)} \frac{\left\langle F\underline{\phi}^{(l+1)}, F\underline{\phi}^{(l+1)} \right\rangle}{\left\langle F\underline{\phi}^{(l+1)}, F\underline{\phi}^{(l)} \right\rangle}. \tag{7.21b}$$

In the formulation above, the power iteration scheme always converges to the largest eigenvalue k_{eff}. Fortunately, this corresponds to the only physical mode of the system (i.e., the one with a nonnegative scalar flux).

7.7.2 Wielandt Shift

7.7.2.1 Traditional Wielandt Shifts

The spectral radius of power iteration is given by the dominance ratio of the system, which is the ratio of the second largest (in magnitude) eigenvalue to the largest eigenvalue. For many problems of interest, power iteration alone can be prohibitively slow, requiring $O(100)$ or more iterations.

Thus, the power iteration scheme should be accelerated, and one common technique for doing so is the Wielandt shift [90]. The goal of the Wielandt shift is to "shift" the eigenvalue spectrum by k_s, an estimate of the true k_{eff}, so that the dominance ratio is reduced. With a Wielandt shift applied, the power iteration scheme in Eqs. (7.20) is modified

as follows:

$$\left[-\nabla\cdot D_g(\mathbf{x})\nabla+\Sigma_{t,g}(\mathbf{x})\right]\phi_g^{(l+1)}(\mathbf{x})-\sum_{g'=1}^{G}\left[\Sigma_{s0,g'\rightarrow g}(\mathbf{x})+\frac{\chi_g(\mathbf{x})}{k_s^{(l)}}\nu\Sigma_{f,g'}(\mathbf{x})\right]\phi_{g'}^{(l+1)}(\mathbf{x})$$
$$=\left[\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}}\right]\chi_g F^{(l)}(\mathbf{x}), \tag{7.22a}$$

$$F^{(l+1)}(\mathbf{x})=\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\mathbf{x})\phi_{g'}^{(l+1)}(\mathbf{x}), \tag{7.22b}$$

$$k_{\text{eff}}^{(l+1)}=\left[\left(\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}}\right)\frac{\int F^{(l+1)}(\mathbf{x})F^{(l)}(\mathbf{x})d^3\mathbf{x}}{\int F^{(l+1)}(\mathbf{x})F^{(l+1)}(\mathbf{x})d^3\mathbf{x}}+\frac{1}{k_s^{(l)}}\right]^{-1}. \tag{7.22c}$$

When using a Wielandt shift, care must be taken to avoid over-shifting. In particular, k_s must be chosen to be strictly larger than the true k_{eff}. If $k_s \leq k_{\text{eff}}$, two problems are encountered: the fission source becomes negative (or identically zero), and it is possible for the method to converge to the incorrect eigenmode (one with negative scalar flux components).

This leads to an important question: how should k_s be chosen? Ideally, k_s should be slightly larger than k_{eff}, but k_{eff} is generally not known until the problem has already been solved. In many diffusion codes such as the Purdue Advanced Reactor Core Simulator (PARCS) code [22], an iteration-dependent or adaptive Wielandt shift is employed, defined as:

$$\frac{1}{k_{s,\text{PARCS}}^{(l)}}\equiv\max\left\{\frac{1}{k_{\text{eff}}^{(l)}}-c_1\left|\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_{\text{eff}}^{(l-1)}}\right|-c_0,\frac{1}{k_{\max}}\right\}. \tag{7.23}$$

Here c_1 and c_0 are user-specified constants, and $k_{\max}$ is a user-specified upper bound for $k_{\text{eff}}^{(l)}$ that is usually determined by the physics of the problem. The term with c_1 is a measure of how converged $k_{\text{eff}}^{(l)}$ is, while $c_0>0$ ensures that a shift of exactly k_{eff} is never made, since doing so would yield a singular system in Eq. (7.20a). Some typical values for c_1, c_0, and $k_{\max}$ are 10, 0.01 or 0.02, and 3. The notation simplifies if $\lambda=\frac{1}{k_{\text{eff}}}$ is used instead of k_{eff}, but k_{eff} is chosen to maintain consistency with the rest of this theory manual. The PARCS Wielandt shift is based on the idea that $k_{\text{eff}}^{(l)}$ is a reasonable estimate of k_{eff}. It can provide an order of magnitude reduction in the number of power iterations compared to power iteration with a constant Wielandt shift.

7.7.2.2 Space-Dependent Wielandt Shift

The PARCS Wielandt shift is effective compared to constant Wielandt shifts, but there are situations (i.e., at the beginning of the power iteration scheme) in which $k_{\text{eff}}^{(l)}$ is not well converged and/or deviates significantly from k_{eff}. In these situations, an improved Wielandt shift may be desired. This motivates the development of a space-dependent Wielandt shift (SDWS). As suggested by its name, SDWS can be formulated by adding space-dependence to the shift

k_s. Eqs. (7.22) are modified as follows:

$$\left[-\nabla\cdot D_g(\mathbf{x})\nabla+\Sigma_{t,g}(\mathbf{x})\right]\phi_g^{(l+1)}(\mathbf{x})-\sum_{g'=1}^{G}\left[\Sigma_{s0,g'\rightarrow g}(\mathbf{x})+\frac{1}{k_s^{(l)}(\mathbf{x})}\nu\Sigma_{f,g'}(\mathbf{x})\right]\phi_{g'}^{(l+1)}(\mathbf{x})$$
$$=\left[\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}(\mathbf{x})}\right]\chi_g F^{(l)}(\mathbf{x}), \tag{7.24a}$$

$$F^{(l+1)}(\mathbf{x})=\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\mathbf{x})\phi_{g'}^{(l+1)}(\mathbf{x}), \tag{7.24b}$$

$$k_{\text{eff}}^{(l+1)}=\frac{\int F^{(l+1)}(\mathbf{x})F^{(l+1)}(\mathbf{x})d^3\mathbf{x}}{\int F^{(l+1)}(\mathbf{x})\left[\left(\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}(\mathbf{x})}\right)F^{(l)}(\mathbf{x})+\frac{1}{k_s^{(l)}(\mathbf{x})}F^{(l+1)}(\mathbf{x})\right]d^3\mathbf{x}}. \tag{7.24c}$$

This reduces to Eqs. (7.22) when $k_s^{(l)}(\mathbf{x})$ is constant.

Three SDWS variants are described: SDWS-LE (Local Eigenvalue), SDWS-LEPS (LE Positive Source), and SDWS-ILEPS (Improved LEPS). Each of these variants is an improvement over the previous variant, and SDWS-ILEPS is the option currently available in MPACT. The three shifts are defined as follows:

$$k_{s,\text{LE}}^{(l)}(\mathbf{x})\equiv k_\infty(\mathbf{x}), \tag{7.25}$$

$$k_{s,\text{LEPS}}^{(l)}(\mathbf{x})\equiv\max\{k_{s,\text{LE}}^{(l)}(\mathbf{x}),k_{\text{eff}}^{(l)}\}, \tag{7.26}$$

$$k_{s,\text{ILEPS}}^{(l)}(\mathbf{x})\equiv\min\{k_{s,\text{LEPS}}^{(l)}(\mathbf{x}),k_{s,\text{PARCS}}^{(l)}\}. \tag{7.27}$$

At each point $\mathbf{x}$ (or in each spatial cell), $k_\infty(\mathbf{x})$ is the local infinite-medium eigenvalue, defined as the solution of the following local $G\times G$ problem:

$$\Sigma_{t,g}(\mathbf{x})\phi_{0,g}(\mathbf{x})-\sum_{g'=1}^{G}\Sigma_{s0,g'\rightarrow g}(\mathbf{x})\phi_{0,g'}(\mathbf{x})=\frac{\chi_g}{k_\infty(\mathbf{x})}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\mathbf{x})\phi_{0,g'}(\mathbf{x}). \tag{7.28}$$

In matrix notation, this can be expressed as

$$\left[\Sigma_t(\mathbf{x})-\Sigma_{s0}(\mathbf{x})\right]\underline{\phi_0}(\mathbf{x})=\frac{1}{k_\infty(\mathbf{x})}\underline{\chi}\left[\underline{\nu\Sigma_f}(\mathbf{x})\right]^T\underline{\phi}_0(\mathbf{x}). \tag{7.29}$$

Because this system only has one nonzero eigenvalue (the matrix $\chi\left[\nu\Sigma_f\right]^T$ has a rank of 1), $k_\infty(\mathbf{x})$ can be computed without iteration as follows:

$$k_\infty(\mathbf{x})=\left[\underline{\nu\Sigma_f}(\mathbf{x})\right]^T\left[\Sigma_t(\mathbf{x})-\Sigma_{s0}(\mathbf{x})\right]^{-1}\underline{\chi}. \tag{7.30}$$

The LE shift provides a physically motivated shift whose performance may be better than that of traditional Wielandt shifts such as the PARCS shift. For a homogeneous system with reflective or periodic boundary conditions, the LE shift yields the exact eigenvalue, and the method converges in 1 iteration (if a singular system is solved). For general heterogeneous problems, numerical experiments have shown convergence in $O(10)$ iterations.

However, the LE shift can over-shift the CMFD system, producing a negative fission source in Eq. (7.24a) and causing the method to either diverge or converge to an unphysical solution. The LEPS method was developed to remedy this

issue. The LEPS method, presented in Eq. (7.26), bounds the value of $k_s(\mathbf{x})$ so that the fission source can never have negative components.

The ILEPS shift is a simple improvement of the LEPS shift that leverages the benefits of the PARCS shift. While the LEPS shift generally outperforms the PARCS shift at the beginning of the iteration scheme (when the eigenvalue is not well converged), the PARCS shift generally outperforms SDWS-LEPS at the end of the iteration scheme (when the eigenvalue is relatively converged). SDWS-ILEPS, presented in Eq. (7.27), combines the two shifts, yielding a shift that always uses the more aggressive of the two shifts.

As a result, the number of power iterations required with SDWS-ILEPS is always less than or equal to the number of power iterations required for SDWS-LEPS or the PARCS shifts alone. However, the improvement provided by SDWS-ILEPS over SDWS-LEPS or the PARCS shift is problem-dependent.

More information on the shifts can be found in an article by Yee et al. [95].

7.7.2.3 Impact on Linear Solvers

The use of a Wielandt shift (space-dependent or not) can significantly reduce the number of power iterations required, but the left side of the shifted CMFD system can be significantly more ill-conditioned (closer to singular) than the unshifted CMFD system. Therefore, an iterative linear method may require many more iterations to solve Eq. (7.24a). This presents a trade-off that must be carefully considered. On the one hand, an effective Wielandt shift reduces the number of power iterations required. On the other hand, if the linear solver struggles, then each power iteration can require significantly more time, offsetting the benefit of reducing the number of power iterations. When choosing a linear solver for CMFD power iterations, it is important that the linear solver (and/or preconditioner) is relatively insensitive to the ill-conditioning caused by the Wielandt shift. Often, power iteration does not converge when a Wielandt shift is used unless the maximum number of permitted linear solver iterations is increased.

7.7.3 Convergence Criterion for CMFD Power Iterations

This section describes the logic for determining when to stop performing power iterations on the CMFD system and return to the transport sweeper. MPACT exits the CMFD solver if one of two criteria are met: (1) the number of power iterations reaches the maximum allowed (currently, this maximum is set to 20), or (2) the normalized residual has been reduced by a specified factor (currently, this factor is 100) and k_{eff} is not changing by more than some specified tolerance (currently, this tolerance is 10^{-6}).

If the CMFD system is expressed in matrix-vector notation as

$$M\underline{\phi} = \frac{1}{k_{\text{eff}}} F\underline{\phi}, \tag{7.31}$$

then the second condition can be expressed as follows:

$$\frac{||r^{(l)}||}{||r^{(0)}||} \equiv \frac{\left|\left|M\underline{\phi}^{(l)} - \frac{1}{k_{\text{eff}}^{(l)}} F\underline{\phi}^{(l)}\right|\right| / \left|\left|\underline{\phi}^{(l)}\right|\right|}{\left|\left|M\underline{\phi}^{(0)} - \frac{1}{k_{\text{eff}}^{(0)}} F\underline{\phi}^{(0)}\right|\right| / \left|\left|\underline{\phi}^{(0)}\right|\right|} < 100, \tag{7.32}$$

$$\left|k_{\text{eff}}^{(l)} - k_{\text{eff}}^{(l-1)}\right| < 10^{-6}. \tag{7.33}$$

7.8 Red-Black Successive Over-Relaxation CMFD

7.8.1 Red-Black Gauss-Seidel

The linear systems in CMFD can be solved by a variety of techniques. A PETSc-based [6] GMRES solver has been used as the default in MPACT for several years. One recent suggestion was to pursue a more basic iterative solver which may be more efficient. One of the most basic of such solvers is the Gauss-Seidel scheme, which loops over the coarse mesh cells, solving for the flux in each cell using the most up-to-date neighboring cell data available.

Several different techniques exist for determining the order in which the solution is obtained; one popular approach is the red-black scheme [20], which tags each coarse cell as red or black to produce a checkerboard pattern, as in Fig. 7.2. This shows the layout for a small 2 $\times$ 2 core of assemblies with 3 $\times$ 3 pins each. Gray boundaries show the assembly edges.

The indexing of the spatial cells within each assembly is natural or lexicographic, but it should be noted that globally, the indexing is not natural. This is because the assembly indexing is not typically lexicographic and is usually based on a z-tree partitioning scheme. Figure 7.3 is a flow chart of the solver's iteration strategy, which involves looping over and solving the fluxes for the red indices, passing data as necessary, looping over the black indices, and again passing data as necessary. Because the inner iteration convergence for Gauss-Seidel solvers tends to be slower than that for many other iterative solvers, each CMFD eigenvalue update iteration is restricted to performing a user-specified maximum number of iterations. A maximum of 50–100 inner iterations seems sufficient for 2D cases, but 3D cases with feedback perform best with 100–150 iterations. Otherwise, the differences in the total number of outer iterations are noticeable. To increase the applicability of the solver, both message passing interface (MPI) and open multi-processing (OpenMP) are currently available to provide spatial parallelization. One advantage of this is that PETSc is restricted to using MPI [6], although a hybrid MPI/OpenMP approach is being considered by the PETSc development team.

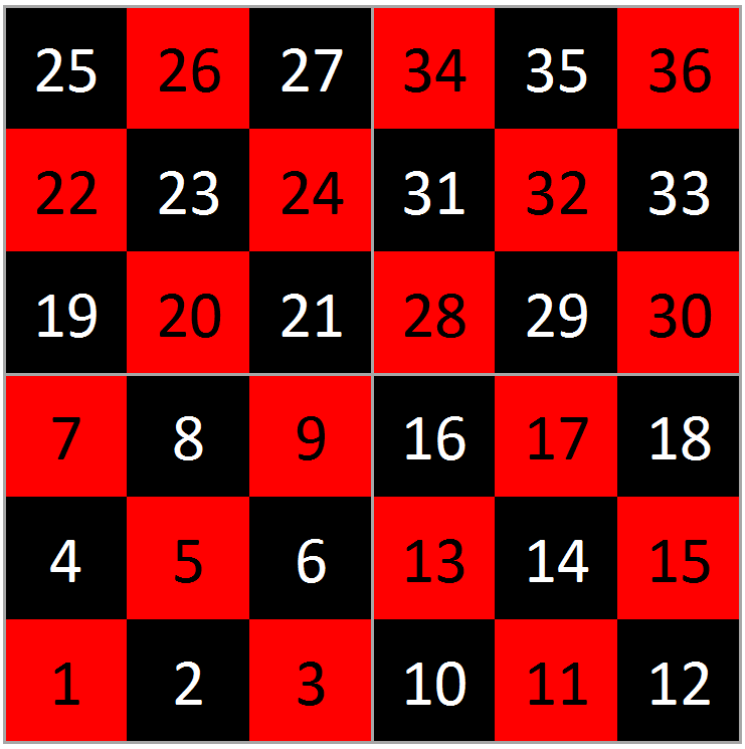


Figure 7.2. Red-black indexing.

7.8.2 Successive Over-Relaxation (SOR)

Gauss-Seidel is a special case of the successive over-relaxation (SOR) solver, in which the relaxation factor equals unity. SOR typically applies a relaxation factor larger than unity to accelerate the convergence of the system [20, 89]:

$$\phi_{n,g}^{(l)} = \omega_{R/B}^{(l-1)} \phi_{n,g}^{(l-\frac{1}{2})} + (1 - \omega_{R/B}^{(l-1)}) \phi_{n,g}^{(l-1)}, \tag{7.34}$$

where l denotes the iteration index, n denotes the cell index, and g denotes the group index. ω is the relaxation factor, which changes each iteration, and is different for red (R) and black (B) cells. The determination of optimal relaxation factors has been studied extensively. The following implementation uses adaptive relaxation factors based on the cyclic Chebyshev semi-iterative (CCSI) method [89, 39], in which the red and black fluxes use different relaxation factors, Eqs. (7.35) and (7.36), that eventually converge to the same value. Equation (7.35) shows the initial relaxation factors, and Eq. (7.36) shows the relaxation factor for subsequent iterations:

$$\omega_R^{(0)} = 1, \tag{7.35a}$$

$$\omega_B^{(0)} = \frac{1}{1 - \frac{1}{2}\rho_J^2}, \tag{7.35b}$$

$$\omega_R^{(l+1)} = \frac{1}{1 - \frac{1}{4}\rho_J^2 \omega_B^{(l)}}, \; l \geq 1, \tag{7.36a}$$

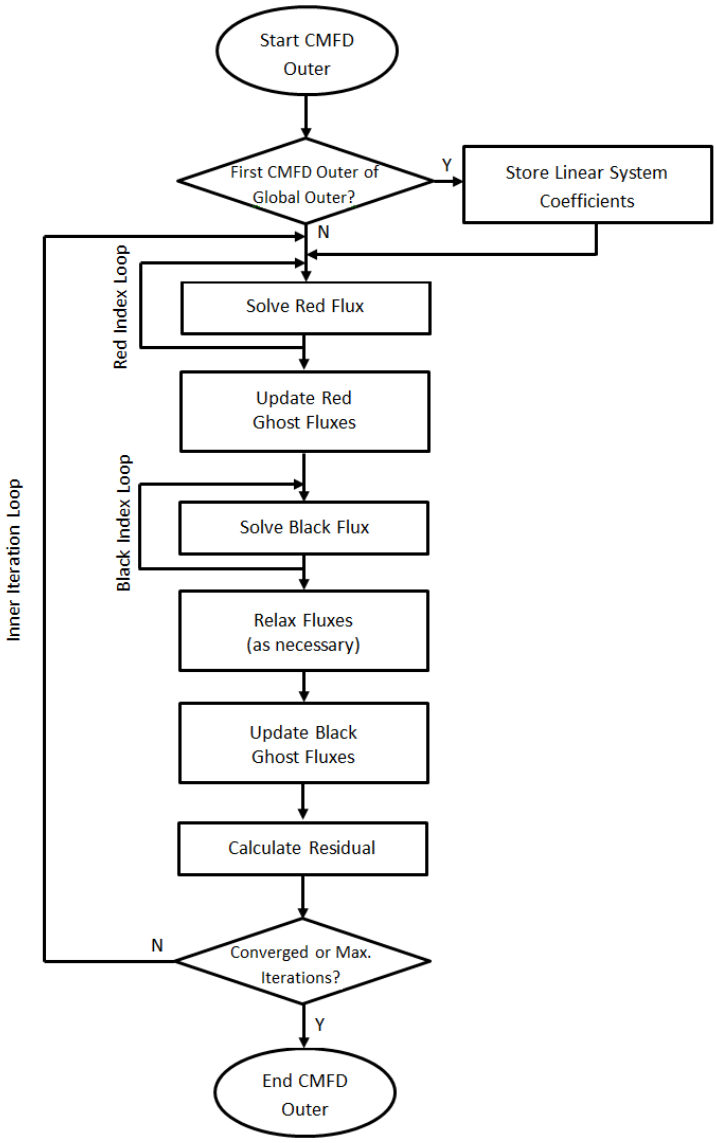


Figure 7.3. Red-black SOR CMFD iteration strategy.

$$\omega_B^{(l+1)} = \frac{1}{1 - \frac{1}{4}\rho_J^2 \omega_R^{(l+1)}}, \; l \geq 1. \tag{7.36b}$$

These equations make use of the Jacobi spectral radius, which in this work is estimated during the first 10 inner iterations without relaxation. The pseudoerror (two-norm of the multigroup flux difference) is used to approximate the Gauss-Seidel spectral radius and is then converted to the Jacobi spectral radius ($\rho_{GS} = \rho_J^2$). It is worth noting that these relaxation factors only seem to be valid when rotational symmetry is not applied to any boundaries. With rotational symmetry, some red-red and black-black neighbors will occur along those boundaries. These will likely invalidate the underlying theory of the derivation. The results presented in this work exclusively consider mirror symmetry, although roughly similar time reductions are observed with rotational symmetry where the relaxation is disabled.

With the RBSOR solver and appropriate limitations on the number of inner iterations performed each outer iteration, a roughly 2× reduction in CMFD solve time is observed, with roughly 40% reduction overall.

7.9 Summary

In this chapter, the CMFD acceleration technique for accelerating transport sweeps is described. CMFD is a technique in which transport sweeps are accelerated by the solution of a lower-order diffusion eigenvalue problem. This acceleration is required to converge realistic reactor physics problems in a reasonable number of transport sweeps. A recently developed adCMFD variant is also described. This variant provides improved convergence and stability. Moreover, the subplane technique for improving the accuracy and stability of the CMFD-accelerated 2D/1D transport problem is described in Section 7.6. Finally, the PI scheme for solving the CMFD eigenvalue problem is described in Section 7.7. Each step in the PI scheme requires the solution of a linear system, and the RBSOR technique for solving this linear system is described in Section 7.8.

8. General Cross Section Data Calculation

To solve the neutron transport equation for a reactor core, the properties that define its interactions with neutrons must be known. These properties determine the coefficients of the transport equation, i.e., the macroscopic cross sections. In Section 2.2, the steady-state Multigroup (MG) transport equation has been derived,

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi_g(\boldsymbol{x},\boldsymbol{\Omega})+\Sigma_{t,g}(\boldsymbol{x})\psi_g(\boldsymbol{x},\boldsymbol{\Omega}) = &\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &+\frac{\chi_g(\boldsymbol{x})}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\int_{4\pi}\nu\Sigma_{f,g'}(\boldsymbol{x})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega', \\ &\boldsymbol{x}\in V,\boldsymbol{\Omega}\in 4\pi, 1\leq g\leq G. \end{aligned} \tag{8.1}$$

Unlike nodal diffusion codes where the problem-dependent MG cross sections are obtained from the upstream lattice codes, MPACT uses a direct transport method without spatial homogenization, so the calculation of MG cross sections should be based on a generic cross section library including isotopic cross sections and resonance parameters. This chapter outlines generation of the MG library and calculation of the cross sections in the MG neutron transport equation. In Section 8.1, the philosophy of the MPACT MG library is introduced. The calculation of the macroscopic cross sections and their correlations to the library data are discussed in Section 8.2. In Section 8.3, the additional data used for transient calculation are discussed. Section 8.4 addresses the data used for estimating the reactor thermal power. For some of the isotopes and the certain energy groups, sophisticated treatments are needed to account for the resonance self-shielding effects on MG cross sections. These methods will be discussed in Chapter 9.

8.1 The Multigroup Cross Section Library

The goal of the MPACT library development is to generate an accurate MG library for general LWR applications with an acceptable computational efficiency in memory and computing time. The efficiency requirement suggests a relatively coarse energy group structure with approximately 50 groups. Various cross section processing programs and methods have been developed to enhance the accuracy of the neutronics simulation [57]. The recently developed ENDF/B-VII.1 MPACT 51-group library has been verified through a code-to-code comparison for various benchmark calculations between MPACT and the continuous-energy (CE) Monte Carlo codes [56]. Validations for the MPACT cross section library and transport solvers are presented in Ref. [21].

The native MPACT cross section library format, which is based on the HELIOS [34] and DeCART library [54] formats, is the primary structure available for MPACT. The AMPX/SCALE and CASL XSTools code packages have been improved and developed to generate the MG cross section library for the CASL neutronics simulator MPACT, which is based on the subgroup method for resonance self-shielding calculation. The AMPX/SCALE code package has been improved to include IR parameters and to enhance accuracy of resonance data and scattering matrices for the Bondarenko approach and transport cross sections of H-1 by developing new modules. In addition, the CASL XSTools have been developed to generate the subgroup data and the MPACT MG cross section library. A new simple superhomogenization method for U-238 has been developed to resolve reaction rate discrepancy issues due to angle-dependent total cross sections, poor scattering matrices, and a poor resonance interference model at coarse group structure. The details of these methods are presented in Ref. [57].

The required data for the steady-state transport calculation are the transport cross section, the fission cross section, the average number of neutrons released from a fission reaction, the scattering matrices, and the fission spectrum for each isotope. Since the absorption (and fission) cross sections are modified through the resonance treatment and are needed for the depletion calculation, these cross sections should be included as well. The (n,2n) and (n,3n) cross sections are also included for depletion calculations. High order (P1-P3) scattering matrices are also included. Therefore, the multigroup data required in the library for steady-state calculation are as follows:

- total cross section $\sigma_{t,g}$
- transport cross section $\sigma_{tr,g}$
- absorption cross section $\sigma_{a,g}$
- fission cross section $\sigma_{f,g}$
- neutrons released from a fission ν
- scattering cross sections $\sigma_{s,l,g}$
- (n,2n) cross section $\sigma_{n\rightarrow 2n,g}$
- (n,3n) cross section $\sigma_{n\rightarrow 3n,g}$
- P_{0-3} scattering matrices $\sigma_{s,l,g'\rightarrow g}$
- fission spectrum χ_g

8.2 Macroscopic Cross Section

The macroscopic cross sections, e.g., $\Sigma_{t,g}$, are calculated by summing up the total cross sections of all isotopes in a material region,

$$\Sigma_{t,g} = \sum_{iso} N_{iso} \sigma_{t,g,iso}. \tag{8.2}$$

In Eq. (8.2), N_{iso} is the number density of an isotope in the unit of $atoms/(barn \cdot cm)$, where $barn$ is a cross section unit of $10^{-24} cm^2$. In practice, the material density ρ (g/cm^3) and the wt% component w_{iso} are often specified, so the number density of an isotope is calculated as,

$$N_{iso} = \rho w_{iso} N_{AVO} 10^{-24} / A_{iso}. \tag{8.3}$$

where $N_{AVO} = 6.02214076 \times 10^{23} mol^{-1}$ is the Avogadro's number, and A_{iso} is the atomic weight of iso in atomic mass units.

The group-wise microscopic cross sections, such as $\sigma_{t,g,iso}$ and $\sigma_{f,g,iso}$ are obtained from the cross section library. For the isotopes and energy groups without resonance behavior, these microscopic cross sections are pre-calculated by typical PWR spectra [57] and are tabulated as a function of temperature. Linear interpolation is used when the cross section data of a temperature are not available in the library. For the resonance isotopes in the resonance energy groups, the self-shielding calculation should be performed to calculate the problem-dependent effective cross sections. The details of self-shielding methods are documented in Chapter 9.

The fission spectrum $\chi_g(\boldsymbol{x})$ in Eq. (8.1) is written to be only dependent on space. In fact, different fissionable isotopes have their own fission spectra provided in the cross section library. To preserve the neutron production for each energy group, the averaged fission spectrum of a material region should be weighted by the isotopic fission sources,

$$\chi_g(\boldsymbol{x}) = \frac{\sum_{iso} \chi_{g,iso}(\boldsymbol{x}) \sum_{g'=1}^{G} N_{iso}(\boldsymbol{x}) \int_{4\pi} \nu\sigma_{f,g',iso}(\boldsymbol{x}) \psi_{g'}(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega'}{\sum_{iso} \sum_{g'=1}^{G} N_{iso}(\boldsymbol{x}) \int_{4\pi} \nu\sigma_{f,g',iso}(\boldsymbol{x}) \psi_{g'}(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega'}. \tag{8.4}$$

Typical MG cross section libraries include coefficients (or moments) of the expansion for scattering matrices, instead of directly storing the data with angular dependence as in Eq. (8.1). To achieve the consistent expansion form, we expand the scattering matrices by Legendre polynomials, and the angular fluxes in the scattering source term by spherical harmonics [44],

$$\Sigma_{s,g'\to g}(\boldsymbol{x}, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}) = \sum_{l=0}^{\infty} \frac{2l+1}{2} \Sigma_{s,l,g'\to g}(\boldsymbol{x}) P_l(\mu), \tag{8.5}$$

$$\psi_g(\boldsymbol{x}, \boldsymbol{\Omega}) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \sum_{m=-l}^{l} \psi_{g,l,m}(\boldsymbol{x}) Y_{l,m}(\boldsymbol{\Omega}). \tag{8.6}$$

If Eqs. (8.5) and (8.6) are inserted into the scattering source term, the spherical harmonic addition theorem can then

be applied. When this is performed, the resultant scattering source term is shown in Eq. (8.7).

$$\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' =$$
$$\sum_{g'=1}^{G}\sum_{l=0}^{\infty}\frac{2l+1}{4\pi}\Sigma_{s,l,g'\rightarrow g}(\boldsymbol{x})\sum_{m=-l}^{l}\psi_{g',l,m}(\boldsymbol{x})R_{l,m}(\boldsymbol{\Omega}). \tag{8.7}$$

In this equation, $R_{l,m}$ are the real components of the spherical harmonics. The microscopic cross sections $\sigma_{s,l,g'\rightarrow g,iso}$ of each isotope are tabulated with a finite number of scattering order L ($L=3$ in the MPACT library). A truncation error is expected in the realistic calculation when using low-order scattering.

The transport correction has been discussed in Section 5.3.3 to save computational resources but mitigate the error from the isotropic scattering assumption. With transport correction, Eq. (8.7) can be reduced to,

$$\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \approx \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma^{tr}_{s,g'\rightarrow g}(\boldsymbol{x})\phi_{g'}(\boldsymbol{x}). \tag{8.8}$$

Consistently, the total cross section in Eq. (8.1) should be replaced by the transport cross section $\Sigma_{tr,g}$,

$$\Sigma_{tr,g} = \Sigma_{t,g} - \Sigma_{s0,g} + \Sigma^{tr}_{s,g}. \tag{8.9}$$

In the MPACT library, $\sigma_{tr,g,iso}$ is provided for each isotope, so Eq. (8.9) can be used reversely to compute the transport corrected scattering cross section.

8.3 Transient Data Calculation

The methodology for solving transient problems will be discussed in Chapter 11, but the time-dependent MG neutron transport equation and the precursor equation are given here to discuss the data needs for transient calculation,

$$\frac{1}{v_g}\frac{\partial\psi_g(\boldsymbol{x},\boldsymbol{\Omega},t)}{\partial t} = -\boldsymbol{\Omega}\cdot\nabla\psi_g(\boldsymbol{x},\boldsymbol{\Omega},t) - \Sigma_{t,g}(\boldsymbol{x},t)\psi_g(\boldsymbol{x},\boldsymbol{\Omega},t)$$
$$+\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\boldsymbol{x},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega},t)\psi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}',t)d\Omega'$$
$$+\frac{1}{4\pi}\left(\chi_{p,g}(\boldsymbol{x},t)(1-\beta(\boldsymbol{x},t))S_F(\boldsymbol{x},t)+\chi_{d,g}(\boldsymbol{x},t)S_d(\boldsymbol{x},t)\right), \tag{8.10}$$

$$\frac{\partial C_\tau(\boldsymbol{x},t)}{\partial t} = \beta_\tau(\boldsymbol{x},t)S_F(\boldsymbol{x},t) - \lambda_\tau C_\tau(\boldsymbol{x},t) \quad ,\tau=1,2,...,n(n=6\,or\,8). \tag{8.11}$$

The subscript τ is the delayed group index, C_τ is the delayed neutron precursor concentration, β_τ is the delayed neutron fraction, λ_τ is the delayed group decay constant, and v_g is the group velocity of neutrons. χ_p and χ_d are the prompt and delayed fission spectra, and S_F and S_d are the total fission source and the delayed neutron source. The rigorous way of defining the kinetics data for prompt and delayed neutron sources should include the isotopic dependence of these data, so the total fission source of group g in Eq. (8.10) should be explicitly defined as,

$$S_g = \sum_{iso}\chi_{p,g,iso}(1-\beta_{iso})S_{F,iso} + \sum_{iso}\chi_{d,g,iso}S_{d,iso}. \tag{8.12}$$

In this equation,

$$S_{F,iso} = \sum_{g'} \nu_{g',iso} \Sigma_{f,g',iso} \phi_{g'}, \tag{8.13}$$

$$S_{d,iso} = \sum_{\tau} \lambda_{\tau,iso} C_{\tau,iso}. \tag{8.14}$$

Also, the precursor equation for each fissionable isotope is given as,

$$\frac{dC_{\tau,iso}(\boldsymbol{t})}{dt} = \beta_{\tau,iso} S_{F,iso}(\boldsymbol{t}) - \lambda_{\tau,iso} C_{\tau,iso}(\boldsymbol{t}) \quad , \tau = 1,2,...,n. \tag{8.15}$$

The delayed neutron fractions in Eq. (8.12) and Eq. (8.15) are defined as,

$$\beta_{(\tau),iso} = \frac{\sum_{g'} \nu^{(\tau)}_{d,g',iso} \sigma_{f,g',iso} \phi_{g'}}{\sum_{g'} \nu_{g',iso} \sigma_{f,g',iso} \phi_{g'}}. \tag{8.16}$$

In the MPACT library, $\chi_{d,g,iso}$ and $\chi_{g,iso}$ are included, so $\chi_{p,g,iso}$ can be calculated. $\nu_{g',iso}$ and $\nu^{(\tau)}_{d,g',iso}$ are the average and delayed number of neutrons released per fission. The energy group dependent $\nu^{(\tau)}_{d,g',iso}$ are not included in the MPACT library. Instead, the effective $\beta_{\tau,iso}$ are pre-generated using typical fission rates from a PWR configuration. Recently, an option has been added in MPACT to evaluate $\beta_{\tau,iso}$ on-the-fly using $\nu^{(\tau)}_{d,g',iso}$ from other kinetics data sets [21].

The MPACT library provides the delayed group decay constants $\lambda_{\tau,iso}$ for every fissionable isotope. However, another common approximation to save memory is neglecting the isotope dependence of precursor concentrations $C_{\tau,iso}$ by calculating the effective λ_{τ}. Several options are available in MPACT, from the approximation of fission source weighted λ_{τ}, to the full isotope-dependent calculation. This treatment is no more a concern for the 8-group delayed data [24], since all the isotopes have the same set of decay constants.

8.4 Data for Thermal Power Calculation

To convert fission rates into energy deposition, which is the true heat generated from a reactor core, the energy release per fission should be used. The energy released by a fission event consists of various energy modes. In addition to the fission energy release, the capture of neutrons, e.g., via (n,γ) or (n,α) reactions also produce energy. Most recoverable energy is released instantaneously in the forms of kinetic energy from fission products and fission neutrons, prompt gamma rays, or gamma rays from the capture of neutrons. About 7% of energy is released some time after the fission event, from the radioactive decay of fission products, in the forms of delayed beta and gamma rays. Equilibrium delayed energy release is reached after a period of steady power history, and usually the time dependence of the delayed energy is neglected in most reactor core simulations, except for estimating the decay heat during reactor shutdown. Figure 8.1 provides the various energy release modes from neutron fission and capture, and the features of these energy modes.

	Energy modes		Fraction	U-235(MeV)	U-238(MeV)	Range	Time
Direct fission energy	Fission fragments		80%	169.13	169.80	Local	Instant
	Beta decay of FPs		3-4%	6.50	8.48	Local	Delayed
	Gammas	Prompt	3-4%	6.60	6.68	Global	Instant
		Delayed	3-4%	6.33	8.25		Delayed
	Neutrinos		4-5%	8.75	11.39	Unrecoverable	Delayed
Indirect energy	Neutron capture		3-5%	6.0-11.0		Global/local	Instant/delayed
	Neutron scattering (slowing-down)		2-3%	4.0-5.0		Local	Instant

Direct fission energies are from ENDF/B-VII.1 (uncertainties below 0.5MeV)

Figure 8.1. Fission Energy Release

By default, MPACT uses the effective fission energy releases [53] for fissionable isotopes tabulated in the MPACT library. This effective energy release includes an equilibrium delayed beta and gamma energy, and an averaged energy release contribution from neutron captures. The incident neutron energy is also considered through a typical PWR spectrum. Given a reactor thermal power, the neutron flux should be normalized by the normalization factor f determined in the following equation,

$$P(t) = f \sum_j V_j \sum_{iso} \widetilde{\kappa}_{f,iso} N_{iso,j} \sum_g \sigma_{f,g,iso,j} \phi_{g,j}. \tag{8.17}$$

where, $\widetilde{\kappa}_{f,iso}$ is effective fission energy release for isotope *iso* (unit: J), and j is the region index. Spatially, the default option in MPACT assumes that all the energy is deposited locally in the fission sites. This means only fuel rods can produce heat. To consider the heat directly generated in moderator via neutron slowing-down and gamma ray reactions, a direct moderator heating fraction is used when coupled with TH calculation.

To improve the energy deposition calculation, explicit energy deposition methods were developed: 1) The time dependence of the delayed energy modes is modeled in depletion and transient problems; 2) The neutron capture and slowing-down heat are calculated explicitly at the place where the reactions occur, so that all the material regions produce energy; 3) Gamma heating effect is approximately considered by developing a gamma smearing model. The details of these methods are presented in Refs. [66] [67].

9. Cross Section Resonance Self-Shielding

To obtain the problem-dependent multigroup cross sections for MPACT, resonance self-shielding calculations must be performed before the whole-core MOC calculations. In general, the goal of resonance self-shielding calculations is to obtain the effective cross sections of an isotope at reaction channel x for group g,

$$\sigma_{x,g}(\mathbf{r}) = \frac{\int_{\Delta u_g} \sigma_x(\mathbf{r},u)\,\phi(\mathbf{r},u)\,du}{\int_{\Delta u_g} \phi(\mathbf{r},u)\,du}\,. \tag{9.1}$$

It is not feasible to accurately determine the neutron spectra of a specific problem before one performs the 3D whole-core transport calculations, so the spectra used in the calculation of effective cross sections are always approximated through the energy and/or spatial domains.

In general, there are two approaches to perform the resonance self-shielding calculation. The approach that guarantees accuracy in the energy domain is to solve the exact slowing-down equations for the problem of interest. Almost no approximations are made on the energy-dependent cross sections except for neglecting the direct fission and up-scattering contribution in the resolved resonance range. Continuous-energy (CE) cross sections (point-wise or ultra-fine group) are needed to resolve the resonance behavior of neutron interactions with isotopes. Because of limited computational resources, CE slowing-down codes such as CENTRM [91] and RMET21 [64] are usually designed for pin cell calculations.

Another approach uses precomputed resonance integral (RI) tables which are established by the CE slowing-down solution of a range of background cross sections. Based on the equivalence theory [79], different methods can be derived to determine the equivalent cross sections for consideration of spatial self-shielding effects. The Bondarenko background cross section method [8] is the conventional method, incorporating Dancoff factors to account for the spatial self-shielding. The subgroup method [19] is another RI table-based method in which the RI tables are converted to a set of subgroup levels and weights so that the equivalence cross sections are subgroup-level dependent. A few years ago, another promising RI table based method, the embedded self-shielding method (ESSM), was proposed [46, 92]. Compared to the conventional Bondarenko method that evaluates Dancoff factors outside the transport calculation, ESSM provides tighter coupling between the neutron transport and self-shielding calculations, so that the heterogeneous self-shielding effects are consistent with multigroup transport calculations of the whole system. Recently, a more advanced method, ESSM-X [68], was developed to account for within-pin physics, such as pin-resolved reaction rates and resonance interference.

The MPACT code can perform the subgroup method, ESSM, and ESSM-X for the resonance self-shielding calcu-

lations based on multigroup libraries with subgroup parameters, such as the HELIOS library [34]. The MPACT production library is processed with ENDF-BVII.0 and ENDF-BVII.1 cross sections, using SCALE-6.2 [9], as well as a few auxiliary codes for generating subgroup parameters, resonance upscattering data, and transport cross section data for hydrogen.

The presentation of the resonance self-shielding treatment begins with the derivation of the neutron spectra to be used in the RI calculations in Section 9.1. The subgroup methods, ESSM, and ESSM-X are discussed in Section 9.2, followed by the treatment of resonance interference and resonance scattering cross sections in Section 9.3. Two methods to improve the performance of subgroup calculations are introduced in Sections 9.4 and 9.5.

9.1 The Resonance Self-Shielding Treatment

The rigorous approach to obtain the neutron spectrum in Eq. (9.1) is to solve the CE slowing down equation,

$$\begin{aligned}&\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\Sigma_{t,i}\left(\mathbf{r},u\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)\\&\qquad=\frac{1}{4\pi}\sum_i\int_{u-\varepsilon_i}^{u}\Sigma_{s,i}\left(\mathbf{r},u'\right)\phi\left(\mathbf{r},u'\right)\frac{\exp^{u'-u}}{1-\alpha_i}du',\end{aligned}\tag{9.2}$$

where α_i is the maximum fraction of energy loss per neutron scattering off isotope i, as defined by its atomic mass A relative to a neutron mass,

$$\alpha_i=(\frac{A_i-1}{A_i+1})^2,\tag{9.3}$$

and ε_i is the maximum lethargy gain per neutron scattering,

$$\varepsilon_i=\ln\frac{1}{\alpha_i}.\tag{9.4}$$

In Eq. (9.2), the neutron energy E has been transformed to lethargy u, as is conventionally done for slowing-down equation. Three assumptions have been made in this equation for the resolved resonance energy range: (1) the scattering source is treated by only considering s-wave elastic reactions, (2) up-scattering is neglected, and (3) direct fission source is neglected. To decouple the lethargy dependence in the scattering source from lethargy $u-\varepsilon_i$ to u, the intermediate resonance (IR) approximation [41] is employed to achieve

$$\begin{aligned}&\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\Sigma_{t,i}\left(\mathbf{r},u\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)\\&\qquad=\frac{1}{4\pi}\left(\sum_i\lambda_i\Sigma_{p,i}\left(\mathbf{r}\right)+\sum_i\left(1-\lambda_i\right)\Sigma_{s,i}\left(\mathbf{r},u\right)\phi\left(\mathbf{r},u\right)\right),\end{aligned}\tag{9.5}$$

where λ_i is the IR factor and $\Sigma_{p,i}$ is the macroscopic potential scattering cross section. If Σ_s is written as $\Sigma_s=\Sigma_p+\Sigma_{RS}$, by neglecting $\lambda_i\Sigma_{RS,i}$ and assuming isotropic fluxes in the second term of the right-hand side, a much simpler equation can be formed without flux dependence in the source term:

$$\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\left(\Sigma_{a,i}\left(\mathbf{r},u\right)+\lambda_i\Sigma_{p,i}\left(\mathbf{r}\right)\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)=\frac{1}{4\pi}\sum_i\lambda_i\Sigma_{p,i}\left(\mathbf{r}\right).\tag{9.6}$$

For a homogenous material, the solution of Eq. (9.6) can be written as

$$\phi_{\text{hom}}(u) = \frac{\sum_i \lambda_i \Sigma_{p,i}}{\sum_i \Sigma_{a,i}(u) + \sum_i \lambda_i \Sigma_{p,i}} . \tag{9.7}$$

The equivalence theory [34] correlates the flux of a homogeneous material with a heterogeneous material by introducing the equivalence cross section Σ_{eq},

$$\phi_{\text{het}}(u) = \frac{\sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq}}{\sum_i \Sigma_{a,i}(u) + \sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq}} = \frac{\Sigma_b}{\sum_i \Sigma_{a,i}(u) + \Sigma_b} , \tag{9.8}$$

where $\Sigma_b = \sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq}$ is the background cross section. By introducing Eq. (9.8) into Eq. (9.1), the effective cross section is only present through the background cross section, so a table of effective cross sections (or RI) can be obtained by varying the background cross sections. With these pre-calculated RI tables, once the equivalence cross sections of a region are properly determined, the effective cross sections can be directly interpolated rather than integrated with the fluxes from the slowing-down solution.

All the RI table-based methods are aimed at estimating the equivalence cross sections of the system. The Bondarenko background cross section method approximately determines the equivalence cross section. The ESSM iteratively solves a fixed source problem (FSP) to converge the equivalence cross section of a system, while the subgroup method evaluates the RI using a quadrature approximation so that equivalence cross sections are calculated on a set of quadrature points, or subgroup levels. Once the equivalence cross section is obtained, the background cross section can be used to either interpolate the RI for all reactions in ESSM or to complete the quadrature calculation for the subgroup method.

9.2 The Subgroup Method, ESSM, and ESSM-X

The subgroup method transforms the integration variable from energy to absorption cross section. Eq. (9.8) indicates that the flux depression is mainly due to the absorption cross sections. Although the absorption cross sections are a strong function of energy (or lethargy), it is more efficient to perform the integration of Eq. (9.1) through absorption cross section rather than neutron energy [34]:

$$\sigma_{x,g} = \frac{\int_{\Delta u_g} \sigma_x(u)\phi(u)\,du}{\int_{\Delta u_g} \phi(u)\,du} = \frac{\int_{\Delta u_g} f(u)\,du}{\int_{\Delta u_g} \phi(u)\,du} = \frac{\int_{\Delta u_g} f(\sigma)\frac{du}{d\sigma}d\sigma}{\int_{\Delta u_g} \phi(\sigma)\frac{du}{d\sigma}d\sigma} , \tag{9.9}$$

where $f(u)$=$\sigma_x(u)\phi(u)$. The integrals of Eq. (9.9) can be cast into a quadrature form represented by the subgroup cross section levels (quadrature points) and weights,

$$\sigma_{x,g} \approxeq \frac{\sum_n \sigma_{x,g,n}\phi_{g,n}w_{x,g,n}}{\sum_n \phi_{g,n}w_{x,g,n}} . \tag{9.10}$$

The subgroup levels and weights are determined by solving a least squares problem for a set of pre-computed RI tables parameterized by background cross sections. To obtain the same set of subgroup levels and weights for the numerator and denominator of Eq. (9.10), the summation of the weights is forced to unity by including zero-level parameters $w_{x,g,0}$ and $\sigma_{x,g,0}$ [34].

The subgroup-level dependent flux $\phi_{g,n}$ in Eq. (9.10) is determined by solving a fixed source problem: the multigroup form of Eq. (9.6),

$$\mathbf{\Omega}\cdot\nabla\varphi_{g,n}(r,\mathbf{\Omega})+\left[\Sigma_{a,g,n}(r)+\lambda_g\Sigma_p(r)\right]\varphi_{g,n}(r,\mathbf{\Omega})=\frac{1}{4\pi}\lambda_g\Sigma_p(r)\Delta u_g\ . \tag{9.11}$$

Since the subgroup parameters are generated for each resonance isotope independently, it is ideal to perform the subgroup FSP calculations for each isotope separately. In practical calculations, to save computing time, the resonance isotopes are grouped into a few categories. Table 1 shows a typical resonance category set. For each category, the FSPs are performed for each energy group and each subgroup level. The resulting fluxes can be used in Eq. (9.10) to estimate the effective cross sections, but we convert the flux into equivalence cross section first. The explanation of this conversion and the detailed subgroup equations are discussed in Section 9.5.

Category	Isotopes
1	U-238
2	U-235, other actinides, FPs
3	Clad isotopes
4	Absorbers (AgInCd, Gd, Hf, etc.)
5	Tungsten

Table 1: A typical resonance category set in MPACT library

ESSM also solves the FSP to determine the equivalence cross sections, but the quadrature representation is replaced by iterations between the fixed source calculations, such as n=1 in Eq. (9.11), and RI table interpolation. Figure 9.1 depicts the ESSM procedures. The effective absorption is a monotonically increasing function of background cross section, as shown by the solid black line. Considering a problem with a true effective cross section $\sigma_{a,\text{true}}$ for which the ESSM is searching, the method starts with an initial guess of $\sigma_{b,0}$ associated with an effective $\sigma_{a,0}$ interpolated from the RI tables. Relative to $\sigma_{a,\text{true}}$, the smaller $\sigma_{a,0}$ introduced into Eq. (9.11) for solving the FSP should result in a lager background cross section $\sigma_{b,1}$ relative to $\sigma_{b,\text{true}}$. Therefore, the iteration is required between the FSP and the RI interpolation to converge the background cross sections, or specifically, the equivalence cross sections.

Comparisons of the subgroup method and ESSM yield the following important differences: (1) the subgroup method requires a separate optimization code to calculate the subgroup levels and weights, which imposes quadrature errors; ESSM is a thoroughly embedded method incorporating everything within the transport calculations, and (2) the computational time of the subgroup method is predetermined by the number of subgroup levels used in the fixed source

calculations, while the computational time of ESSM is cross section-dependent relative to the average number of iterations.

The ESSM-X method performs an additional quasi-1D slowing-down calculation [68] to improve the accuracy of the within-pin resonance effects such as resonance interference and intra-pin temperature distributions,

$$\left[\Sigma_{t,i}(u)+\Sigma_{eq,i}(u)\right]\phi_i(u)=\bar{Q}_i(u)+\Sigma_{eq,i}(u)\,. \tag{9.12}$$

This equation is actually in a 0-D form, but 1-D information for each fuel region i is embedded in the effective scattering source $\bar{Q}_i(u)$ and the equivalence cross section $\Sigma_{eq,i}(u)$. Detailed description of these terms can be found in Liu et al. [68].

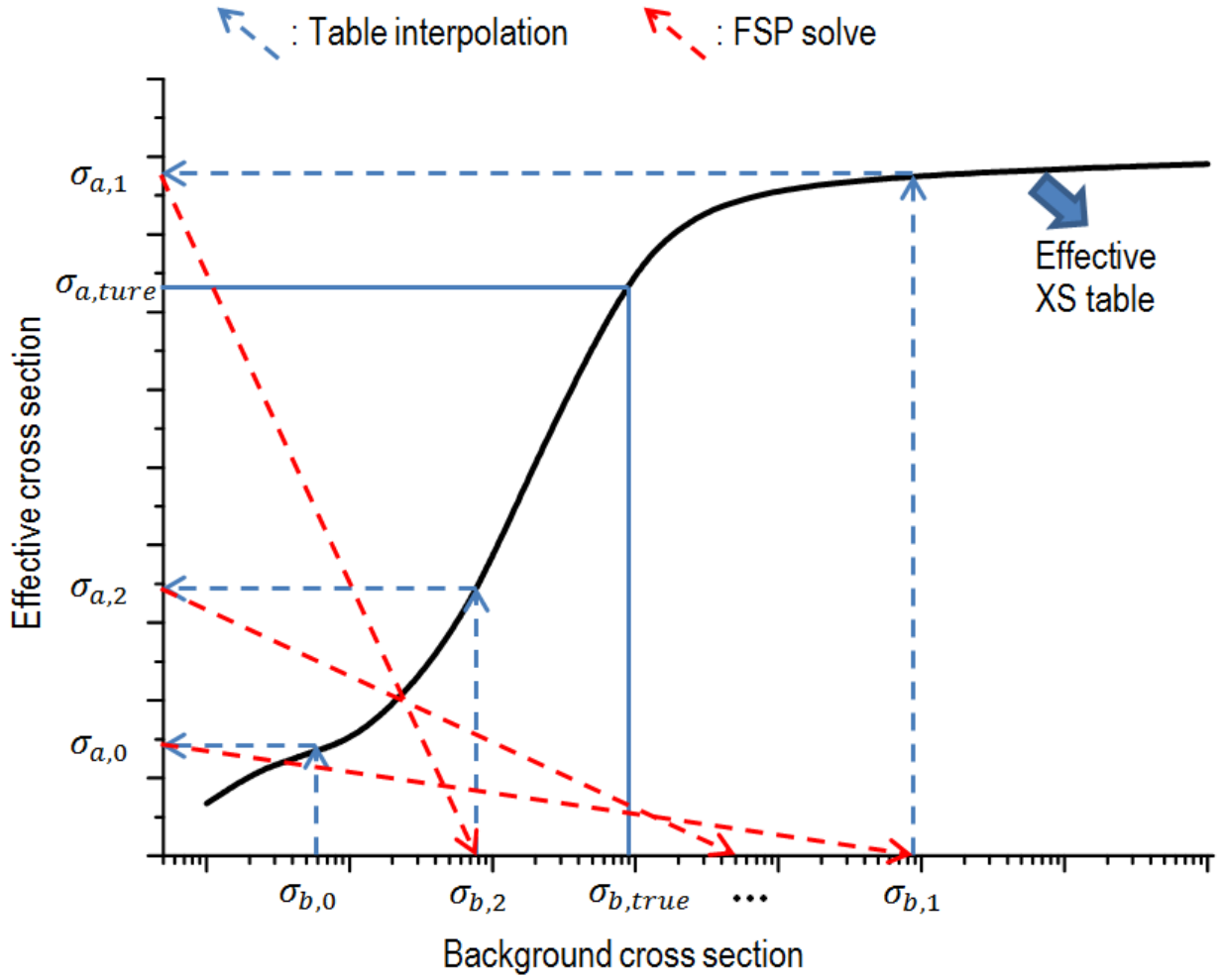


Figure 9.1. Conceptual illustration of ESSM procedures.

9.3 Resonance Interference and Resonance Scattering

All the RI table-based methods such as the subgroup method and ESSM have a common difficulty in treating interference effects among resonant isotopes. Conventionally, the RI tables are generated at different temperatures and dilutions for each single resonant isotope by solving the slowing-down equation with CE cross sections. The interference effect is neglected at this step and is usually being considered afterward on the multi-group level using Bondarenko iteration described in WIMS code [2]. In MPACT, Bondarenko iteration is also used for the subgroup method and ESSM. More accurate models for resonance interference requires on-the-fly slowing-down calculation [65], which is not implemented into the subgroup method and ESSM due to the additional computation burdens. Nevertheless, ESSM-X can model resonance interference by solving the quasi-1D slowing-down equation.

In addition to absorption and fission, the RI tables of resonance scattering are available in the MPACT multigroup library. The resonance scattering effects can be approximately modeled by correcting the 2-D scattering matrices using self-shielded total scattering cross sections. The products of subgroup calculations are the shielded absorption cross sections of resonance nuclides, as well as the tables of the equivalence cross sections against subgroup levels for each resonance category. The shielded absorption cross section of a resonance nuclide is initially converted back to the effective absorption of the representative isotope of the category so that the equivalence cross section can be interpolated using the table of $\Sigma_{eq,n}\left(\sigma_{a,n}\right)$ (see Section 9.5). Then the background cross section associated with shielded absorption can be readily determined. For ESSM, the background cross section associated with the shielded absorption is already determined when the iteration is complete. The background cross section is used to interpolate the integral of resonance scattering, and the shielded P_0 scattering cross section of each resonance group can be calculated. Compared to the unshielded scattering cross section provided by the library, a ratio is defined as

$$f_{\text{iso},g} = \frac{\sigma_{s,\text{iso},g}^{\text{shielded}}}{\sigma_{s,\text{iso},g}^{\text{unshielded}}}. \tag{9.13}$$

The transport corrected scattering matrix can be renormalized by multiplying the ratios:

$$\sigma_{s_0,\text{iso},g\to g'}^{\text{shielded}} = \begin{cases} f_{\text{iso},g}\,\sigma_{s_0,\text{iso},g\to g'}^{\text{unshielded}}, & g' \neq g, \\ f_{\text{iso},g}\left(\sigma_{s_0,\text{iso},g\to g'}^{\text{unshielded}} - \sum\limits_{g''}\sigma_{s_1,\text{iso},g\to g''}^{\text{unshielded}}\right), & g' = g. \end{cases} \tag{9.14}$$

The corrections for high order scattering terms are performed using the same ratios as for the P0 scattering.

9.4 Lumped Parameter MOC for Subgroup

A typical subgroup calculation consists of several loops: (1) over resonant groups, (2) over resonant categories, and (3) over sublevels. These are typically solved one at a time, but the multigroup kernels allow for all or at least some

to be solved concurrently. The calculation scheme for the three-loop approach is shown in Algorithm 9.1. As can be seen, inside these loops, there is an iteration loop in which transport sweeps for each resonant group, category, and level are performed. Here, Σ_p is the potential cross section, Σ_t is the total cross section, and Σ_{eq} is the equivalence cross section. The convergence criteria here are based on the maximum relative difference in the fine mesh scalar flux distribution for any single group, currently checking to a 10^{-6} tolerance.

Algorithm 9.1: Pseudocode for pre-existing subgroup scheme with group on outermost loop.

```
for each resonant group (g from g_res,beg to g_res,end) do
  for each subgroup category (c from 1 to N_cat(g)) do
    for each subgroup level (l from 1 to N_levels) do
      Setup Σ_t,g,c,l based on Σ_a,g,c,l and λ_g Σ_p
      Setup source for this group/category/level based on λ_g Σ_p
      for each iteration (i from 1 to N_iters) do
        Perform transport sweep for this group/category/level
        Compare residual based on scalar flux (terminate if below criteria)
      end for
      Calculate equivalence cross section (Σ_eq,g,c,l)
    end for
  end for
end for
```

To take advantage of the multigroup kernels implemented into MPACT (see Section 5.5.4), the scheme must be restructured slightly. In this document, a single combination of group/category/level is considered to be a *pseudogroup*. The number of pseudogroups for the entire subgroup calculation will be the product of the number of resonant groups, the average number of subgroup categories per group, and the number of subgroup levels. In theory, the number of categories can vary from group to group, but this does not seem to be the case for the current libraries available to MPACT. In the 47-group library [55] used in this work, there are 17 resonant groups, 4 categories, and 4 levels, yielding 272 pseudogroups. Based on this concept, a transport kernel could be constructed to sweep over all pseudogroups concurrently, effectively vectorizing the three loops of the original algorithm. However, the sources, cross sections, scalar fluxes, and angular fluxes must be stored for each pseudogroup up front, whereas in the previous scheme, only one group of storage at a time was necessary. Algorithm 9.2 shows the pseudocode for the refactored scheme, taking advantage of the multigroup kernel concept.

Algorithm 9.2: Pseudocode for subgroup scheme using the multigroup transport kernel.

for each pseudogroup (pg from 1 to $N_{pseudogroups}$) **do**
  Setup and store $\Sigma_{t,pg}$ for this pseudogroup based on $\Sigma_{a,pg}$ and $\lambda_{pg}\Sigma_p$
  Setup and store source for this pseudogroup based on $\lambda_{pg}\Sigma_p$
end for
for each iteration (i from 1 to N_{iters}) **do**
  Perform transport sweep for all pseudogroups
  Compare residual based on scalar flux (terminate if below criteria)
end for
for each pseudogroup (pg from 1 to $N_{pseudogroups}$) **do**
  Calculate equivalence cross section $\Sigma_{eq,pg}$
end for

As might be expected, the memory required for storing the source and flux data for 272 pseudogroups can be a concern. One way to keep the memory low while allowing the scheme to make use of the multigroup kernels is to divide the pseudogroups into batches. Algorithm 9.3 shows the pseudocode for the batched approach, where each batch contains a starting and stopping pseudogroup index:

Algorithm 9.3: Pseudocode for subgroup scheme using the multigroup transport kernel and batching.

for each batch (b from 1 to N_{batch}) **do**
  for each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$) **do**
    Setup and store $\Sigma_{t,pg}$ for this pseudogroup based on $\Sigma_{a,pg}$ and $\lambda_{pg}\Sigma_p$
    Setup and store source for this pseudogroup based on $\lambda_{pg}\Sigma_p$
  end for
  for each iteration (i from 1 to N_{iters}) **do**
    Perform transport sweep for all pseudogroups
    Compare residual based on scalar flux (terminate if below criteria)
  end for
  for each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$) **do**
    Calculate equivalence cross section $\Sigma_{eq,pg}$
  end for
end for

In addition to using multigroup kernels, a lumped parameter MOC approach has been applied to the subgroup self-shielding problem [81]. Because the self-shielding calculation is a purely absorbing fixed source problem, and multiple sweeps are performed only to update the boundary angular fluxes, the sweep procedure can be condensed to allow for the instantaneous propagation of the flux across a spatial domain without the need to sweep along all segments in

a ray as is typically done. This requires an initial sweep to tabulate lumped parameter coefficients for the angular flux propagation. Subsequent sweeps use the lumped parameters to instantly update the angular flux, bypassing all calculations along the ray. Once the boundary angular fluxes are considered to be converged, an additional sweep is completed to tally the scalar flux.

Because the MOC kernels in MPACT sweep over two angles travelling in opposite directions (forward/backward) at the same time, effectively, two equations are needed:

$$\varphi_{pg}^{out,for} = \varphi_{pg}^{in,for} A_{pg} + B_{pg}, \tag{9.15a}$$

$$\varphi_{pg}^{out,back} = \varphi_{pg}^{in,back} A_{pg} + C_{pg}. \tag{9.15b}$$

Here, $\varphi_{pg}^{in,for}$ and $\varphi_{pg}^{in,back}$ are the incoming angular fluxes at each end of an MOC ray (boundary fluxes), one forward along the ray and one backward. Similarly, $\varphi_{pg}^{out,for}$ and $\varphi_{pg}^{out,back}$ are the outgoing angular fluxes. A, B, and C are the lumped parameter coefficients used to condense the sweep.

To visualize this, consider a ray in a simple pin cell problem (Figure 9.2). On the left is the discretization showing 5 segments along the ray (blue) with the incoming and outgoing angular fluxes at the ends of the ray. On the right is the same problem but with all 5 segments condensed into one, as in the lumped parameter approach.

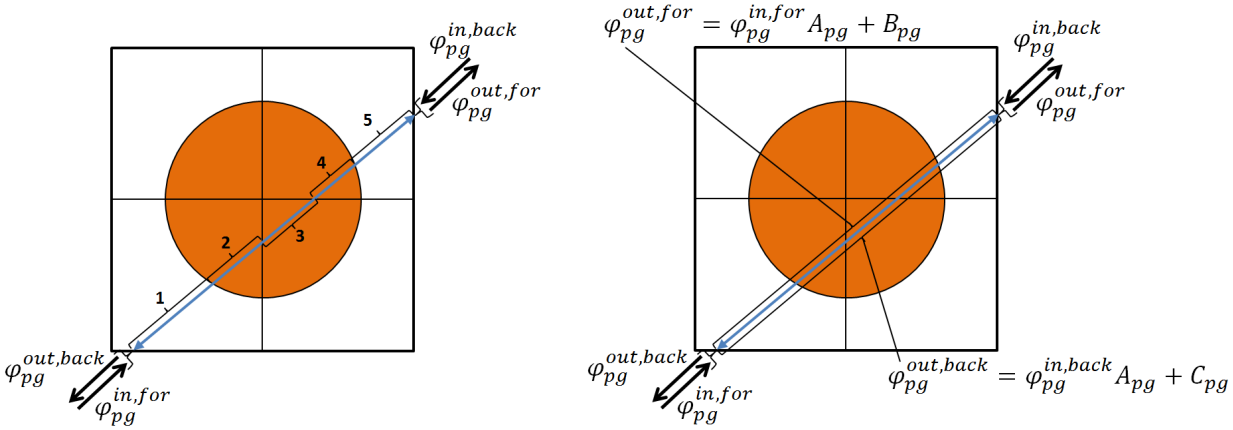


Figure 9.2. Visualization of MOC ray tracing (left) and lumped parameter (right) on a pin cell.

To reiterate, this is only valid and effective because the source is not changing between iterations, as is the case during the eigenvalue calculation sweeps. Thus, the $A/B/C$ lumped parameters can be used in a fast intermediate kernel that only updates the outgoing angular flux.

Two methods are available to derive equations for lumped parameters. It can be concluded that A will be a product

of the exponential terms for each segment, (9.16a). With A established, B and C can be easily calculated using the incoming and outgoing angular flux values, Eqs. (9.16b) and (9.16c), assuming a typical sweep is performed in calculating the factors. Here, N_{seg} is the number of segments along an MOC ray, $\Sigma_{t,i,pg}$ is the total cross section, and l_i is the segment length:

$$A_{pg} = \prod_{i=1}^{N_{seg}} e^{-\Sigma_{t,i,pg} l_i}, \tag{9.16a}$$

$$B_{pg} = \varphi_{pg}^{out,for} - \varphi_{pg}^{in,for} A_{pg}, \tag{9.16b}$$

$$C_{pg} = \varphi_{pg}^{out,back} - \varphi_{pg}^{in,back} A_{pg}. \tag{9.16c}$$

The lumped parameters must be calculated and saved for each angle and ray. Because there will only be three values over O(100) segments, the storage for this is not concerning. Figure 9.4 shows the pseudocode for lumped parameters, which is based on the multigroup kernel with batching. The key changes to note are (1) there is an initial sweep to calculate the lumped parameters (line 6), (2) there are several "fast" sweeps that simply apply the factors to update the angular flux (per Eq. (9.15) and line 8), and (3) a final standard sweep is completed to tally the scalar flux (line 11), which is required for the equivalence cross section calculation (line 13).

Algorithm 9.4: Pseudocode for lumped parameter subgroup scheme using the multigroup transport kernel and batching.

1: **for** each batch (b from 1 to N_{batch}) **do**
2: **for** each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$ **do**
3: Setup and store $\Sigma_{t,pg}$ for this pseudogroup based on $\Sigma_{a,pg}$ and $\lambda_{pg}\Sigma_p$
4: Setup and store source for this pseudogroup based on $\lambda_{pg}\Sigma_p$
5: **end for**
6: Perform and initial sweep accumulating the $A_{pg}/B_{pg}/C_{pg}$ lumped parameters
7: **for** each iteration (i from 2 to N_{iters}) **do**
8: Perform a transport sweep applying $A_{pg}/B_{pg}/C_{pg}$ parameters for this batch
9: Compare residual based on boundary angular flux (terminate if below criteria)
10: **end for**
11: Perform a final, normal sweep accumulating scalar flux (ϕ_{pg})
12: **for** each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$) **do**
13: Calculate equivalence cross section $\Sigma_{eq,pg}$
14: **end for**
15: **end for**

Since only the last iteration yields a scalar flux distribution, the convergence residual for this scheme is based on the angular flux updates instead of the scalar flux, which is used in the current scheme. Choosing the correct convergence criteria is important to ensure consistency between these two schemes. The current scheme imposes a maximum change of 1×10^{-6} for the scalar flux in any region for each pseudogroup. Since the new scheme will perform an additional sweep once the angular flux is considered to be converged, a similar maximum change is imposed on the angular flux, but with a criterion of 1×10^{-5}. In practice, this has been observed to be conservative, in most cases requiring one additional iteration, which is acceptable since it is only one additional fast iteration.

This approach would not be beneficial in problems with fully vacuum radial boundary conditions in serial. In this scenario, only one iteration would be necessary since the boundary conditions do not need to be converged, as a zero incoming angular flux is correct. This is not a likely scenario since most problems are executed with quarter symmetry and in parallel. It is expected that larger spatial domains will reap greater benefits. Pin cell problems would naturally have the least to gain since there are so few segments along a ray. The best case would likely be a full core problem without any radial decomposition, although this is likely impractical because of the substantial memory required. In general, at least 8 radial partitions are used on a quarter core slice to allow for an acceptable amount of memory per core, where a domain consists of a few assemblies of data. In the results section, 73 radial domains are used, which amounts to roughly an assembly per domain, and good performance is observed there.

9.5 Multigroup and 1-Group Subgroup

Standard subgroup calculation requires the FSPs to be solved for every 2D plane in each resonance category, energy group, and subgroup level. To improve the performance of subgroup calculation, besides optimizing the MOC transport sweeper as discussed in Section 9.4, efforts can also be made to reduce the number of FSPs, which leads to the concept of 1-group subgroup. This section compares the standard multigroup and the 1-group subgroup formulations. Also, the detailed equations for the subgroup method are provided as a supplement to Section 9.2. 1-group/multigroup in this section should be interpreted as energy integration of subgroup equation. This is different from the 1-group/multigroup kernels related to the solution scheme of MOC sweep discussed in Section 9.4.

We start with the subgroup FSP of Eq. (9.11),

$$\mathbf{\Omega} \cdot \nabla \varphi_{g,c,n}(r,\mathbf{\Omega}) + \left[\Sigma_{a,g,c,n}(r) + \lambda_g \Sigma_p(r)\right] \varphi_{g,c,n}(r,\mathbf{\Omega}) = \frac{1}{4\pi} \lambda_g \Sigma_p(r) \Delta u_g \, . \tag{9.17}$$

As mentioned in Section 9.2, the multigroup subgroup calculation solves Eq. (9.17) for every energy group g, resonance category c, and subgroup level n, where

$$\Sigma_{a,g,c,n} = \frac{\sum_{i \in c} N^i I^i_{a,g,\infty}(T_{loc})}{I^r_{a,g,\infty}(T_{loc})} \sigma^r_{a,g,n}(T_{loc}),$$

and

$$\lambda_g \Sigma_p = \sum_i \lambda_g^i \Sigma_p^i \ . \tag{9.18}$$

$I^i_{a,g,\infty}(T_{loc})$ is the infinite absorption RI per lethargy for isotope i at local temperature T_{loc}. To adjust the subgroup level for nonuniform temperature effect,

$$\sigma^r_{a,g,n}(T_{loc}) = \frac{\sigma^r_{a,g}(T_{loc})}{\sigma^r_{a,g}(T_{ave})} \bar{\sigma}^r_{a,g,n} = f(T_{loc}) \bar{\sigma}^r_{a,g,n}, \tag{9.19}$$

where $\bar{\sigma}^r_{a,g,n}$ is the subgroup level of the representative isotope in category c in the library. The temperature adjusting ratio can be determined by various methods. In MPACT, base cross sections in the MPACT library are used for $\sigma_{a,g}(T_{loc})$ and $\sigma_{a,g}(T_{ave})$ [69], which are generated using a typical PWR spectrum.

Although the fluxes solved from Eq. (9.17) can be directly used to calculate the effective cross sections, MPACT converts the flux to the equivalence cross section first. By integrating Eq. (9.8) over an energy group, we obtain:

$$\phi_{g,c,n} = \frac{\Sigma_{b,g,c,n} \Delta u_g}{\Sigma_{b,g,c,n} + \Sigma_{a,g,c,n}} = \frac{(\sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq,g,c,n}) \Delta u_g}{(\sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq,g,c,n}) + \Sigma_{a,g,c,n}} \ . \tag{9.20}$$

Equation (9.20) is reversed to determine the equivalence cross section $\Sigma_{eq,g,c,n}$ in terms of $\phi_{g,c,n}$. Instead of directly using $\phi_{g,c,n}$ to calculate effective cross sections, this detour option is chosen because the dependence of $\Sigma_{eq,g,c,n}$ on $\sigma^r_{a,g,n}$ is much weaker than the dependence of $\phi_{g,c,n}$ on $\sigma^r_{a,g,n}$. Therefore, the number of $\sigma^r_{a,g,n}$ capable of describing this dependence can be smaller than the number of subgroup levels used in the quadrature calculation of the effective cross sections, leading to fewer fixed source calculations. Specifically, MPACT uses four subgroup levels for the FSP calculations, and seven levels for the evaluation of effective cross sections. Interpolation is needed to fold the table of $\Sigma_{eq,g,c,n}(\sigma^r_{a,g,n})$ (4 levels) into $\Sigma_{eq,g,c,m}(\sigma^r_{a,g,m})$ (7 levels). This table is not only used for the representative isotope, but it is also used for the other resonance isotopes in the same category c, so a conversion is needed from a resonance isotope to the representative isotope about the subgroup level,

$$\sigma^{i,\arg}_{a,g,m} = \frac{I^r_{a,g,\infty}(T_{loc})}{I^i_{a,g,\infty}(T_{loc})} f(T_{loc}) \bar{\sigma}^i_{a,g,m} \ . \tag{9.21}$$

Here, $f(T_{loc})$ is defined as the same as Eq. (9.19), but is about isotope i. Using $\sigma^{i,\arg}_{a,g,m}$ as the argument to obtain $\Sigma_{eq,g,c,m}(\sigma^{i,\arg}_{a,g,m})$, the background cross section can be computed as

$$\Sigma^i_{b,g,m} = \Sigma_{eq,g,c,m}(\sigma^{i,\arg}_{a,g,m}) + \lambda_g \Sigma_p \ . \tag{9.22}$$

With all the subgroup parameters, the effective cross section is computed as

$$\sigma^i_{a,g} = \frac{\sum_m \bar{\sigma}^i_{a,g,m} \frac{\Sigma^i_{b,g,m}}{\Sigma^i_{a,g,m}(T_{loc}) + \Sigma_{x,g} + \Sigma^i_{b,g,m}} w^i_{a,g,m}(T_{loc})}{\sum_m \frac{\Sigma^i_{b,g,m}}{\Sigma^i_{a,g,m}(T_{loc}) + \Sigma_{x,g} + \Sigma^i_{b,g,m}} w^i_{a,g,m}(T_{loc})} \ , \tag{9.23}$$

where $\Sigma^i_{a,g,m}(T_{loc}) = N^i f(T_{loc}) \bar{\sigma}^i_{a,g,n}$ and $\Sigma_{x,g} = \sum_{j \neq i} \Sigma^j_{a,g}$. This equation means the temperature adjustment is only performed within the flux term by $\Sigma^i_{b,g,m}$ and $\Sigma^i_{a,g,m}(T_{loc})$. In addition, $\Sigma_{x,g}$ term includes the absorption cross section

of all the other resonance isotopes in the same material, so Bondarenko iterations are performed here to determine a converged set of $\sigma^i_{a,g}$ for all resonance isotopes.

In the 1-group subgroup scheme, a similar procedure is performed by integrating over all resonance groups. Two approximations have been made: (1) the IR source is approximated by averaging over the entire resonance energy range, and (2) subgroup levels for FSP are no longer group-dependent. The FSP for 1-group subgroup calculation is given as

$$\mathbf{\Omega}\cdot\nabla\varphi_{c,n}(r,\mathbf{\Omega})+[\Sigma_{a,c,n}(r)+\lambda\Sigma_p(r)]\varphi_{c,n}(r,\mathbf{\Omega})=\frac{1}{4\pi}\lambda\Sigma_p(r)\Delta u. \tag{9.24}$$

This equation should be solved for every resonance category c and subgroup level n (no group dependence), where

$$\Sigma_{a,c,n}=\frac{\sum\limits_{i\in c}\sum\limits_g N^i I^i_{a,g,\infty}(T_{loc})\Delta u_g}{\sum\limits_g I^r_{a,g,\infty}(T_{loc})\Delta u_g}\sigma^r_{a,n}(T_{loc})\ ,$$

and

$$\lambda\Sigma_p=\frac{\sum\limits_g\sum\limits_i \lambda^i_g\Sigma^i_p\Delta u_g}{\sum\limits_g\Delta u_g}\ . \tag{9.25}$$

To adjust the subgroup level for temperature effect,

$$\sigma^r_{a,n}(T_{loc})=\frac{\sigma^r_a(T_{loc})}{\sigma^r_a(T_{ave})}\bar{\sigma}^r_{a,n}=f(T_{loc})\bar{\sigma}^r_{a,n}\ , \tag{9.26}$$

where $\bar{\sigma}^r_{a,n}$ can be the subgroup level for any energy group. MPACT uses the first resonance group defined in the cross section library. $\sigma_a(T_{loc})$ and $\sigma_a(T_{ave})$ are obtained by averaging the base cross section over all resonance groups with flat flux. Once the solution of Eq. (9.24) is available for all subgroup levels, a table $\Sigma_{eq,c,n}(\sigma^r_{a,n})$ can be obtained by the level-dependent flux,

$$\Sigma_{eq,c,n}=\frac{\Sigma_{a,c,n}\phi_{c,n}}{\Delta u-\phi_{c,n}}-\lambda\Sigma_p\ . \tag{9.27}$$

Similarly, the table $\Sigma_{eq,c,n}(\sigma^r_{a,n})$ (n=1-4) can be interpolated into $\Sigma_{eq,c,m}(\sigma^r_{a,m})$ (m=1-7). A conversion is needed from a resonance isotope to the representative isotope about the subgroup level,

$$\sigma^{i,\mathrm{arg}}_{a,g,m}=\frac{I^r_{a,g,\infty}(T_{loc})}{I^i_{a,g,\infty}(T_{loc})}f(T_{loc})\bar{\sigma}^i_{a,g,n}\ . \tag{9.28}$$

Note that Eq. (9.28) retrieves the group index by assuming that the table $\Sigma_{eq,c,m}(\sigma^r_{a,m})$ applies to all resonance groups. In other words, the differences in the source term ($\lambda\Sigma_p$) among groups are neglected when estimating the dependence of $\Sigma_{eq,c,m}$ on $\sigma^r_{a,m}$. The calculations of effective cross sections using subgroup quadratures are identical between multigroup and 1-group formulation.

10. Nuclide Depletion and Decay

In the simulation of depletion, there are two main computational components: (i) the point depletion solver and (ii) time integration algorithm. With regards to the point depletion solver, MPACT can call the ORIGEN code [38], which is included in the SCALE package [76], or it can use an in internal solver. MPACT's internal depletion solver is consistent with the methodology of an earlier version of ORIGEN. The internal solver also uses a separate depletion library. In the remainder of this chapter, the methodology for the internal point depletion solver is described. Then the coupling of the nuclide transmutation equations to the rest of MPACT and the time stepping algorithm are described. The coupling of the transmutation equations and timestepping algorithm apply to both the ORIGEN and internal point depletion solution algorithms.

10.1 Nuclide Transmutation Equation and its Solution

This section first presents the nuclide transmutation equation and common approximations employed for its numerical solutions. The numerical solution of this equation is then described for the internal MPACT solver. A brief discussion of other numerical solution methods is also given.

10.1.1 Nuclide Transmutation Equation

A general expression for the rate of depletion of a nuclide by neutron reactions or radioactive decay can be written as the following coupled system of N first-order differential equations:

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{N} f_{j\rightarrow i}\lambda_j X_j(t) + \bar{\phi}\sum_{k=1}^{N} g_{k\rightarrow i}\sigma_k X_k(t) - (\lambda_i + \sigma_i\bar{\phi})X_i(t),\ 1 \leq i \leq N, \tag{10.1}$$

where

$X_i(t) =$ density of nuclide i at time t,

$\lambda_i =$ radioactive disintegration constant for nuclide i,

$\sigma_i =$ 1-group spectral averaged neutron absorption cross section of nuclide i,

$\bar{\phi} =$ time-integrated 1-group neutron scalar flux ,

$f_{j\rightarrow i} =$ fraction of radioactive disintegrations by nuclide j that lead to the formation of nuclide i,

$g_{k\rightarrow i} =$ the fraction of neutron reactions by nuclide k that lead to the formation of nuclide i.

Equation (10.1) already contains some approximations: specifically, the treatment of the scalar flux. This term is time-dependent. However, to avoid the need to solve a non-linear equation, it is assumed that the scalar flux is invariant in time over the time interval of interest. Generally, this is an adequate assumption, and it only breaks down when the discretized time step becomes too large. Another approximation implicit in Eq. (10.1) is the assumption of no space dependence. There are several different physical phenonma that drive the relocation of certain nuclides. Moreover, the scalar flux is also a spatially dependent quantity. The point approximation for 0D is also very common, and for general applications it is assumed that the spatial discretization needed for the transport equation is sufficient to discretize the scalar flux in space. For LWR analysis, the operating temperatures are typically low enough that there is typically minimal diffusion of nuclides in the UO_2, with the potential exception of fission gas. However, most codes that solve the coupled neutron transport and transmutation equations do not model fission gas release because this physical phenomenon can typically be neglected when solving the transport equation.

Returning to Eq. (10.1), if the nuclide concentrations are expressed as a vector,

$$\mathbf{X} = (X_1, X_2, \ldots, X_i, \ldots, X_N)^T,$$

then Eq. (10.1) can be written in matrix form as:

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{A} \cdot \mathbf{X}(t), \tag{10.2}$$

where $\mathbf{A}$ is an $N \times N$ matrix constructed from characteristic neutron reaction rates and fractions, as well as radioactive decay rates and fractions.

In principle, the matrix exponential method can be used to obtain the solution of Eq. (10.2) as:

$$\mathbf{X}(t) = \exp(\mathbf{A}t)\mathbf{X}(0), \tag{10.3}$$

where the vector $\mathbf{X}(0)$ represents the known particle number densities at the beginning (the initial condition). To facilitate the subsequent derivations, Eq. (10.3) is rewritten assuming a discretization in time.

$$\mathbf{X}(t_n + \Delta t) = \exp(\mathbf{A}\Delta t)\mathbf{X}(t_n). \tag{10.4}$$

Obtaining $\mathbf{X}(t_n + \Delta t)$ then becomes a matter of calculating $\exp(\mathbf{A}\Delta t)$.

10.1.2 Computing the Matrix Exponential

According to the literature, the calculation of the matrix exponential can be performed in one of 19 dubious ways [71]. Several have been used for the solution of the nuclide transmutation equations. The earliest approach taken, used in ORIGEN, is to use a Taylor series expansion, with lots of modifications for accuracy and efficiency. More recent implementations have favored the Chebyshev Rational Approximation method (CRAM), now used in ORIGEN, as well as Krylov methods built on the Arnoldi iteration.

In MPACT, the matrix exponential is computed using a Taylor series expansion:

$$\exp(\mathbf{A}\Delta t) = I + \mathbf{A}\Delta t + \frac{(\mathbf{A}\Delta t)^2}{2!} + \cdots = \sum_{m=0}^{\infty} \frac{(\mathbf{A}\Delta t)^m}{m!}. \tag{10.5}$$

However, if all the nuclides are included in the transition matrix $\mathbf{A}$, then $\mathbf{A}$ becomes very large, sparse, and ill-conditioned; this complicates the calculation of the matrix exponential equation in Eq. (10.3)). Issues also arise in obtaining sufficient accuracy of the solution because of the floating point arithmetic involved in the summation of very large and very small numbers. Consequently, steps are taken to split and precondition the $\mathbf{A}$.

Preconditioning/Splitting of the Nuclide Transmutation Matrix

Because the full problem cannot be solved efficiently by the Taylor series expansion, the matrix is divided into two parts: one for long-lived nuclides, and the other for short-lived nuclides. This has the effect of lowering the condition number of the transmutation matrix by removing the highest magnitude coefficients and leaving only those coefficients with very small magnitudes. With this splitting, the matrix exponential in Eq. (10.3) can be accurately and efficiently computed for the long-lived nuclides only. The solution of the short-lived nuclides is obtained by a different means and is discussed later in the chapter. The criterion for the matrix separation is based on the removal half-life of a nuclide. Since any concentration of a nuclide essentially becomes zero after 10 half-lives, the long-lived nuclides are defined to be the nuclides for which the irradiation time interval is less than 10 times their removal half-life, i.e.,

$$\Delta t \leq 10 t_{r,1/2} \quad \text{where} \quad t_{r,1/2} = \left(\frac{\ln 2}{\lambda_i + \sigma_i \bar{\phi}} \right). \tag{10.6}$$

Given that the transition matrix is only considers long-lived nuclides, the suggestion of Ball and Adams [7] that the transitions involving short-lived nuclides with large removal rates be considered instantaneous is adopted. If we have a decay chain of $A \rightarrow B \rightarrow C$ and the removal half-life for B is very small, then the matrix is reformulated for the decay chain $A \rightarrow C$, where the transition coefficient is formulated considering the impact of the intermediate short-lived nuclide, B, that has been removed. Similarly, if the removal half-life for A is very small, the decay chain is rewritten as $B \rightarrow C$. However, in this case, the amount of isotope B initially present must be adjusted to include the short-lived precursor A so that the initial particle number density of B will be equal to $A + B$ for the matrix exponential calculation. In this process of removing these short-lived nuclides to create a *reduced transition matrix* (or preconditioned transition matrix), special care must be taken to ensure the chains are collapsed correctly.

A generalized treatment of the full transition matrix to produce the *reduced transition matrix* is achieved by condensing sections of the decay chains involving short-lived nuclides and adjusting the coefficients for the formation of the long-lived nuclides at the ends of these intermediate chains. Practically, this done by searching through the individual decay chains and forming a queue of all short-lived precursors for each long-lived nuclide. The queue is terminated when the farthest removed precursor is no longer short-lived. Bateman's equations are then applied to this queue to obtain the rate constants for the reduced transition matrix. For an arbitrary forward branching chain, the general solution for the i^{th} member in a chain at time $t_n + \Delta t$ may be written in the form:

$$X_i(t_n+\Delta t) = X_i(t_n)e^{-d_i\Delta t} + \sum_{k=1}^{i-1} X_k(t_n)\left[\sum_{j=k}^{i-1}\left(\frac{e^{-d_j\Delta t}-e^{-d_i\Delta t}}{d_i-d_j}a_{j+1,j}\prod_{m=k,m\neq j}^{i-1}\frac{a_{m+1,m}}{d_m-d_j}\right)\right]. \tag{10.7}$$

The notation $a_{i,j}$ is used for the first-order rate constant that comes from Eq. (10.1), and $d_i = -a_{i,i}$. In the present application, Eq. (10.7) is recast in the form given below:

$$X_i(t_n+\Delta t) = X_i(t_n)e^{-d_i\Delta t} + \sum_{k=1}^{i-1} X_k(t_n)\prod_{m=k}^{i-1}\frac{a_{m+1,m}}{d_m}\left[\sum_{j=k}^{i-1}\left(d_j\frac{e^{-d_j\Delta t}-e^{-d_i\Delta t}}{d_i-d_j}\prod_{m=k,m\neq j}^{i-1}\frac{d_m}{d_m-d_j}\right)\right], \tag{10.8}$$

by multiplication and division of

$$\prod_{m=k}^{i-1} d_m\,.$$

The first product in Eq. (10.8) is the fraction of atoms of isotope k that follow a particular sequence of decays and captures. If this product becomes less than 10^{-6}, contributions from nuclide k and its precursors to the concentration of nuclide i are neglected. This procedure is unnecessary for evaluating the outer summation because all the terms in this sum are known to be positive. To avoid a division by zero when two removal constants are approximately equal ($d_i \approx d_j$), the bracketed term in Eq. (10.8) is replaced by:

$$\sum_{j=k}^{i-1}\left(d_j\frac{e^{-d_j\Delta t}-e^{-d_i\Delta t}}{d_i-d_j}\prod_{m=k,m\neq j}^{i-1}\frac{d_m}{d_m-d_j}\right) \approx \sum_{j=k}^{i-1}\left(d_j\Delta t e^{-d_j\Delta t}\prod_{m=k,m\neq j}^{i-1}\frac{d_m}{d_m-d_j}\right).$$

An analogous expression is derived for the case when $d_m \approx d_j$. These forms of the Bateman equations are applied when two isotopes in a chain have the same diagonal elements or when a cyclic chain is encountered, in which case a nuclide is considered to be its own precursor. The new rate constant can then be considered as the coefficient of $X_k(t_n)$, where the product over m is over the queue of short-lived nuclides. In this approach, the Bateman solutions complement the exponential matrix method. The exponential matrix method is quite accurate when the transition coefficients are small, but it is less accurate when including large rate constants are included. Conversely, the Bateman solution has numerical difficulties for extremely small rate constants, but it is stable and accurate for large rate constants.

Calculation of the Long-lived Nuclide Number Densities

As a result of the splitting up and preconditioningthe full transition matrix to the reduced transition matrix the Taylor series expansion for the matrix exponential calculation of the long-lived nuclides is written as:

$$\exp(\tilde{\mathbf{A}}\Delta t) = I + \tilde{\mathbf{A}}\Delta t + \frac{(\tilde{\mathbf{A}}\Delta t)^2}{2!} + \cdots = \sum_{m=0}^{\infty}\frac{(\tilde{\mathbf{A}}\Delta t)^m}{m!}. \tag{10.9}$$

However, even with the removal of the short-lived nuclides from the transition matrix $\mathbf{A}$, it is still quite large and sparse, and it is and computationally inefficient to store the reduced transition matrix, $\tilde{\mathbf{A}}$, in its entirety. To avoid having to store the entire matrix and still be able to compute the Taylor series of Eq. (10.9) efficiently, a recursion relation for the expansion is developed. Considering the Taylor expansion for the equation of a single nuclide in $\tilde{\mathbf{A}}$, the terms may be equivalently rewritten as:

$$\begin{aligned}\tilde{X}_i(t_n+\Delta t) =& \tilde{X}_i(t_n)+\Delta t\sum_{j=1}^{N}\tilde{a}_{i,j}\tilde{X}_j+\frac{\Delta t}{2}\sum_{k=1}^{N}\left(\tilde{a}_{i,k}\Delta t\sum_{j=1}^{N}\tilde{a}_{k,j}\tilde{X}_j\right)\\ &+\frac{\Delta t}{3}\sum_{m=1}^{N}\left[\tilde{a}_{i,m}\frac{\Delta t}{2}\sum_{k=1}^{N}\left(\tilde{a}_{m,k}\Delta t\sum_{j=1}^{N}\tilde{a}_{k,j}\tilde{X}_j\right)\right]+\ldots.\end{aligned} \tag{10.10}$$

Here the *tilde* denotes that the equations are operating on just the long-lived nuclides. From Eq. (10.10) the recursion relation becomes more apparent, where each term in the expansion can be written as:

$$C_i^{m+1}=\frac{t}{m+1}\sum_{j=1}^{N}\tilde{a}_{ij}C_j^m, \quad \text{where} \quad C_i^0=\tilde{X}_i(t_n). \tag{10.11}$$

Lastly, the number of terms to include in the Taylor expansion for sufficient accuracy in the computation of the matrix exponential must be determined. For this, the following norm from Lapidus and Luus [60]:

$$||\tilde{\mathbf{A}}||=\min\left[\max_{1\le j\le N}\left(\sum_{i=1}^{N}|\tilde{a}_{i,j}|\right)\cdot\max_{1\le i\le N}\left(\sum_{j=1}^{N}|\tilde{a}_{i,j}|\right)\right]. \tag{10.12}$$

Calculation of the Short-lived Nuclide Number Densities

The nuclide number densities for short-lived nuclide chains beginning with a long-lived precursor are assumed to be in secular equilibrium with their precursors. Therefore, the short-lived nuclide concentrations are computed once the long-lived concentrations are known. To compute the short-lived nuclide number densities, a simple iterative method is used. Since secular equilibrium is assumed, the system of differential equations of the short-lived nuclides can be written as a a system of algebraic equations because the time rate of change is zero:

$$\frac{d\hat{X}_i}{dt}\approx 0=\sum_{j=1}^{N}\hat{a}_{i,j}\hat{X}_j. \tag{10.13}$$

Here the "hat" denotes that we are operating on just the short-lived nuclides. Equation (10.13) can be solved easily using a Gauss-Seidel iteration. The coefficients in Eq. (10.13) have the property that all the diagonal elements of the matrix are negative and all off-diagonal elements are positive. The algorithm involves inverting Eq. (10.13) and using assumed or previously calculated values for the unknown concentrations to estimate the concentration for the next iteration:

$$X_i^{\ell+1}=-\frac{1}{\hat{a}_{i,i}}\sum_{j\neq i}\hat{a}_{i,j}\hat{X}_j^{\ell}. \tag{10.14}$$

This iterative procedure has been found to converge rapidly since cyclic chains are not usually encountered for these short-lived isotopes, and the procedure reduces to a direct solution.

10.1.3 MPACT Point Depletion Solution Algorithm

The previous subsection described how the matrix exponential is formulated and calcuated. In this description, the overall procedure for this was mentioned, but was not explicitly defined. This subsection explicitly defines the algorithm for performing the point depletion calculation in MPACT. Within ORIGEN, a separate algorithm may be used; for a description of the latest algorithms available in ORIGEN, see the most current SCALE manual [76].

The point depletion algorithm in MPACT is illustrated in Sections 10.1 and 10.2. It is noted here that the time discretization scheme that is used is explicit.

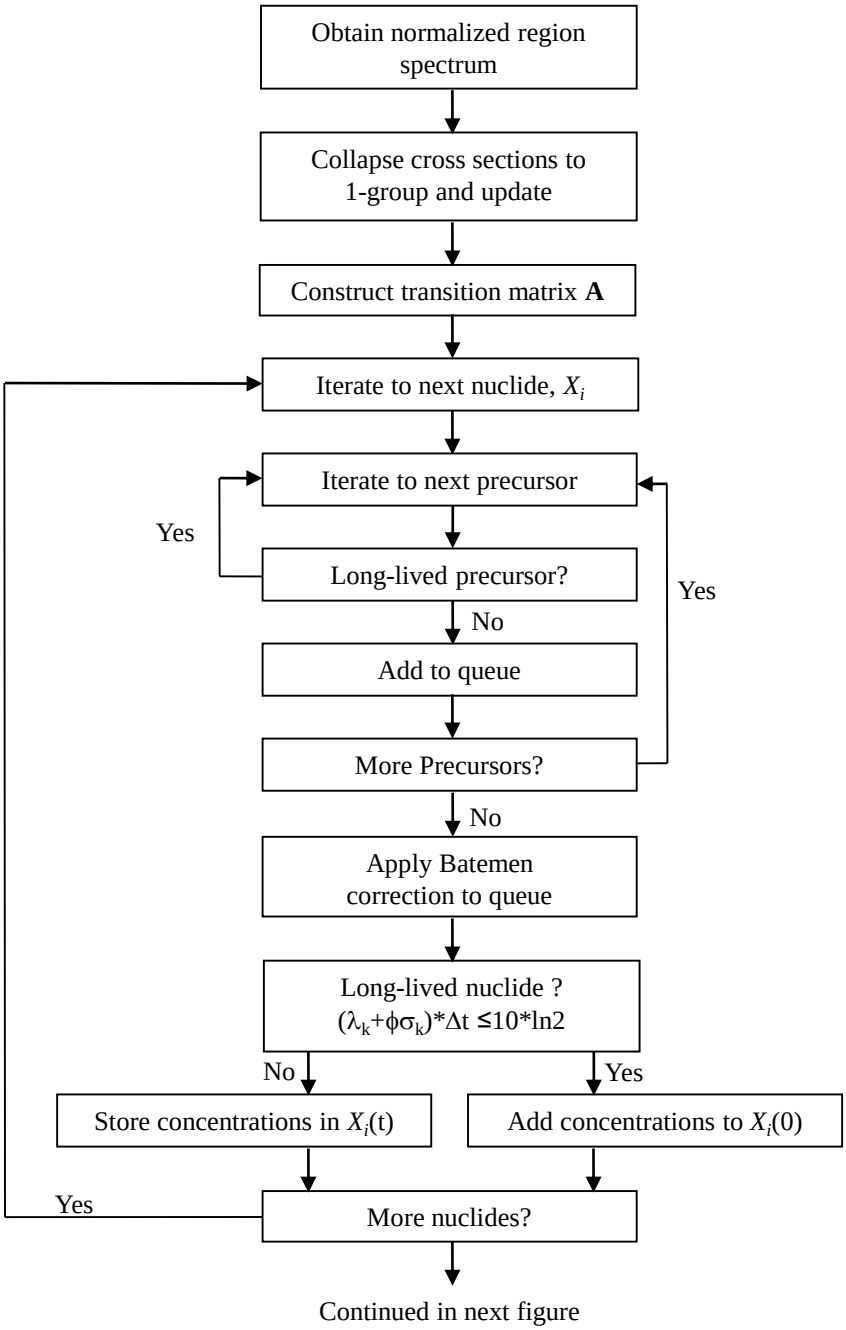


Figure 10.1. Point depletion algorithm (1 of 2).

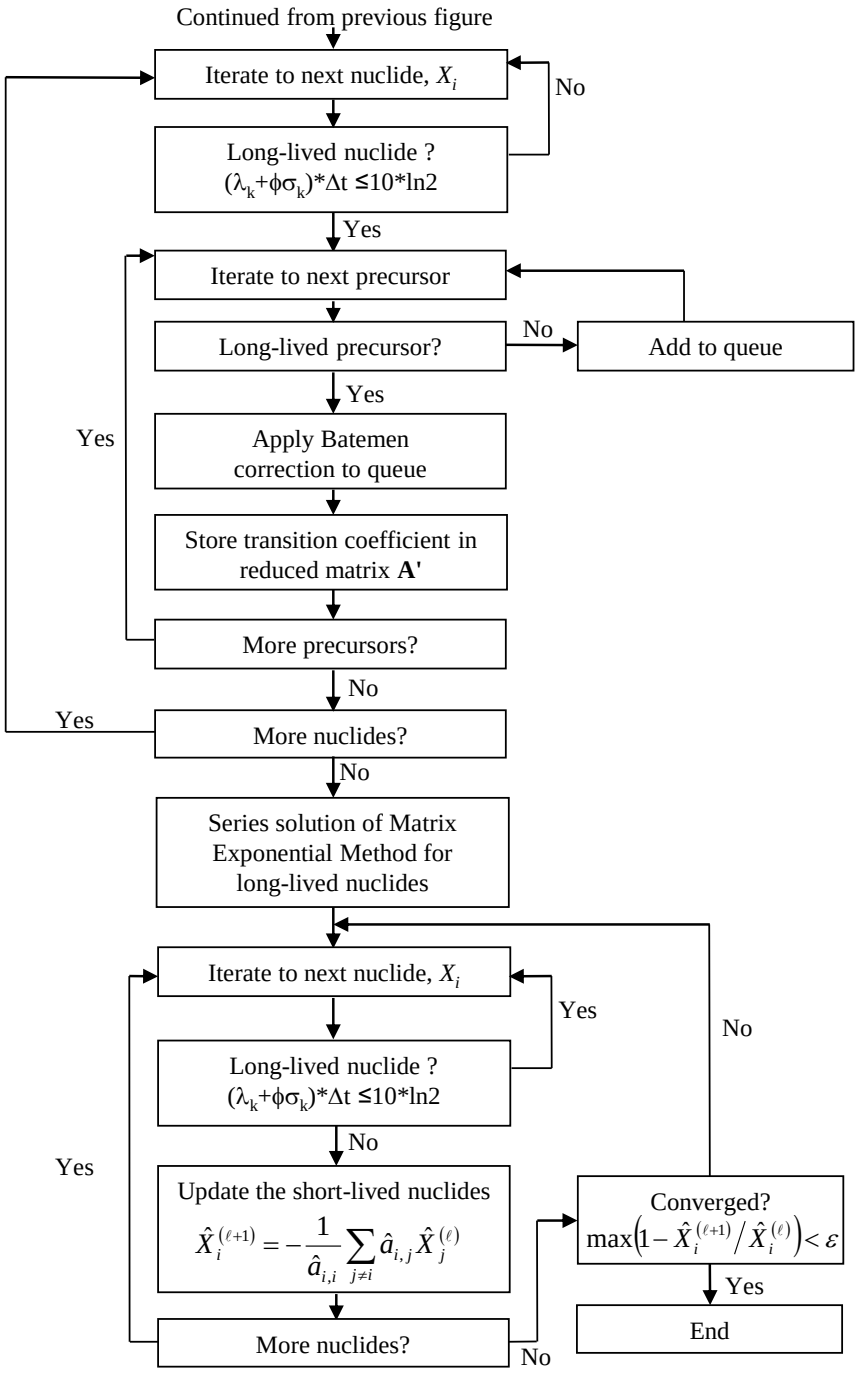


Figure 10.2. Point depletion algorithm (2 of 2).

10.2 Coupling of the Neutron Transport and Nuclide Transmutation Equations

In MPACT, the flat source regions of the fuel pin are azimuthally and radially dependent. These regions have the finest discretization of the scalar flux. Currently, the depletable region of the pin is only radially dependent. The difference in these spatial regions is illustrated in Figure 10.3, where the right-hand side of the figure shows the flat source or flat scalar flux region, and the left-hand side shows a uniform cross section or depletion region. This approach of mapping the flux and cross section regions has been shown to be adequate unless there are very strong local asymmetries. The capability to treat azimuthally varying depletable regions is a simple modification, but it will considerably increase memory requirements.

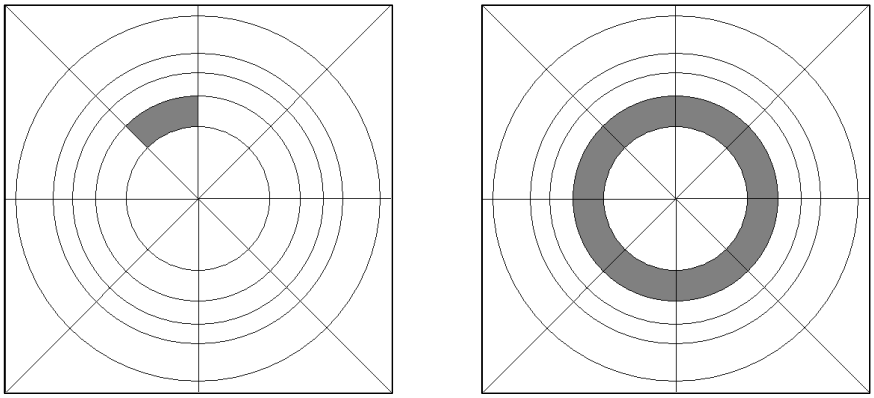


Figure 10.3. Depletion zones in MPACT pin cell.

10.2.1 Predictor-Corrector

In each step of a depletion calculation, the flux is assumed to be constant with time. There are several techniques for incorporating the time dependence of the flux into the depletion calculation. This is typically accomplished by dividing the depletion problem into a series of time steps and periodically performing transport calculations. However, because the time dependence of the flux has nonlinear feedback from the change in the fuel composition, the optimum depletion step size is often not known *a priori*, and to maintain an accurate solution, the time steps are often very small. This leads to longer computation time, which is undesirable. Therefore, to reduce computation time and allow for longer burnup step sizes, MPACT adopts two commonly used techniques: (i) the predictor-corrector and (ii) sub-step methods. The predictor-corrector method works by computing a predicted nuclide concentration for a given time step and then a corrected nuclide concentration. The basic predictor-corrector approach is illustrated by Figure 10.4

and given in Eq. (10.15).

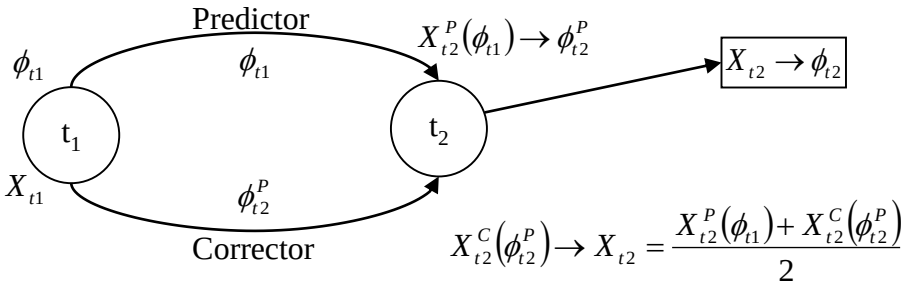


Figure 10.4. Illustration of depletion predictor-corrector.

$$X_{t2} = \frac{X_{t2}^{P}(\phi_{t1}, \sigma_{t1}) + X_{t2}^{C}(\phi_{t2}^{P}, \sigma_{t2}^{P})}{2}. \tag{10.15}$$

The predictor step includes the typical depletion calculation to obtain the particle number densities, $X_{t2}(\phi_{t1}, \sigma_{t1})$, at burnup t_2 by using the 1-group flux and cross sections at the time of burnup t_1. At this point the new predicted 1-group flux (ϕ_{t2}^{p}) and cross sections (σ_{t2}^{p}) are obtained through a transport calculation using the predicted concentration, $X_{t2}^{P}(\phi_{t1}, \sigma_{t1})$. Next, the corrector step performs a depletion calculation using the new 1-group flux (ϕ_{t2}^{p}) and cross section (σ_{t2}^{p}), and the new corrected particle number densities, $X_{t2}^{C}(\phi_{t2}^{P}, \sigma_{t2}^{P})$, are obtained. The final particle number densities for t_2 are then taken to be the arithmetic mean of the predicted and corrected concentrations. Once X_{t2} is obtained, then a transport calculation is performed to obtain the steady-state flux distribution at t_2, (ϕ_{t2}).

10.2.2 Substep Method

The substep method is applied to perform multiple depletion calculations between transport calculations. This has the effect of reducing the numerical error in the evaluation of the matrix exponential. Furthermore, since the depletion calculation typically takes less time than the transport calculation, this will often save computational time. Mathematically, this results in the normalization factor of the flux, f, becoming time-dependent, but the eigenvector, which represents the spatial flux distribution is still assumed to be constant between transport calculations. For M substeps the m^{th} flux, representing the flux at time $t_1 + m\Delta t/M$, used by the depletion calculations, is:

$$\phi_m = \phi_{t1} f_{m-1} \quad \text{where} \quad f_{m-1} = \frac{P_{t1}}{\sum_j \sum_i X_{m-1}^{i,j} \kappa\sigma_{t1}^{f,i,j} \phi_{t1}^{j}}, \tag{10.16}$$

where P_{t1} is the total power at t_1, ϕ_{t1}^{j} and $\kappa\sigma_{t1}^{f,i,j}$ are the eigenvector for region j ,and the energy per fission multiplied by the microscopic fission cross section of region j and nuclide i at t_1, respectively, and $X_{m-1}^{i,j}$ is the nuclide concentration of sub-step $m-1$. The substep method allows for even coarser burnup steps without a loss in accuracy.

10.2.3 Depletion Time-stepping algorithm

An overview of the depletion algorithm in MPACT is depicted in Figure 10.5. This algorithm is the same in MPACT regardless of the point depletion solution methodology being used.

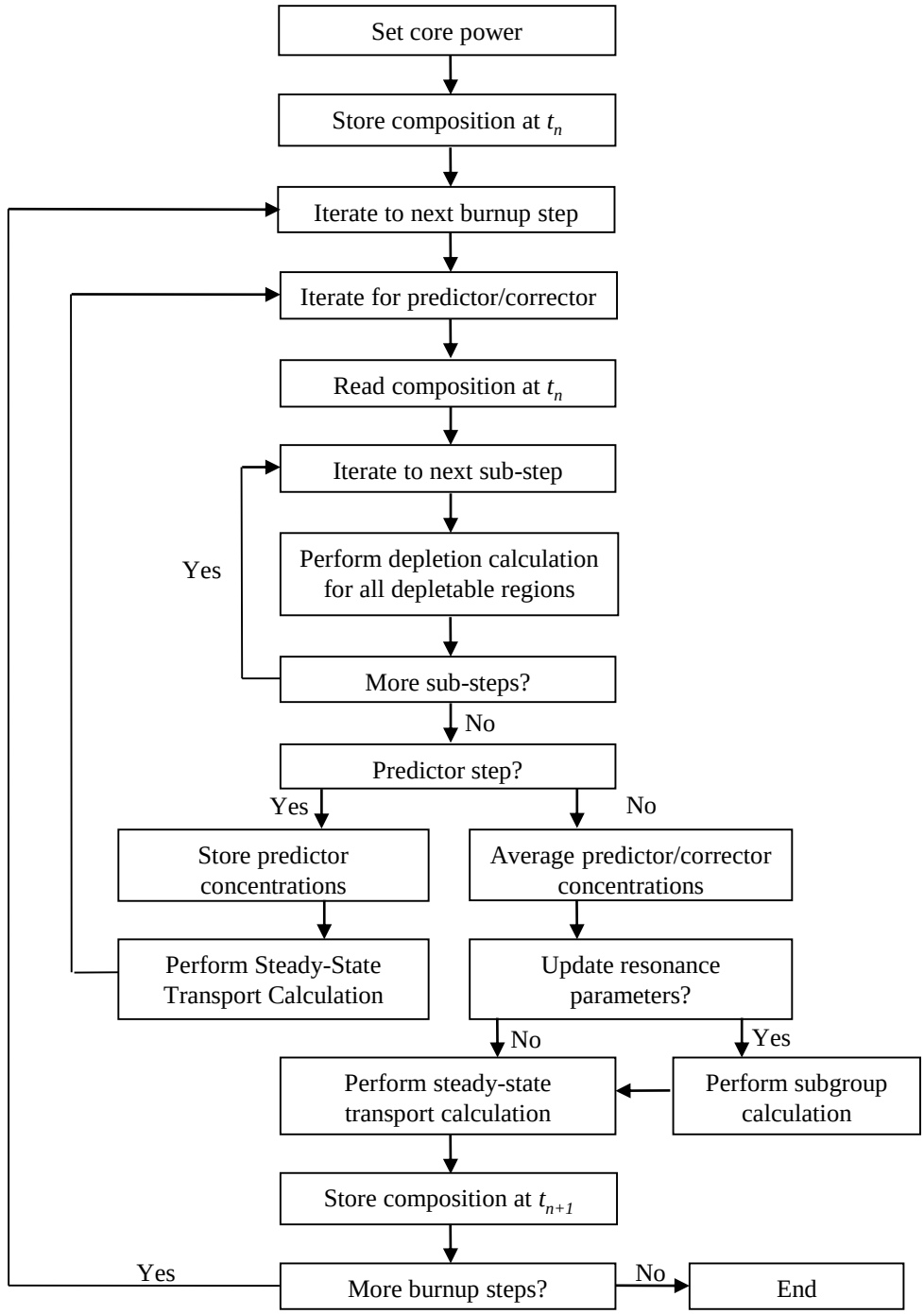


Figure 10.5. MPACT depletion algorithm.

11. Transient Methods

11.1 Transient Methods within the 2D/1D Framework

This chapter describes the methods for extending the 2D-1D steady-state solution to the time-dependent problem. The derivation of the MPACT transient solution method begins with the formulation of the 3D transient fixed source problem (TFSP), that involves the time discretization and the precursor integration technique. The chapter then gives the 2D-1D discretization of the TFSP, first for 2D MOC, then for the 1D axial solver. This follows closely with chapters 2, 4, 5, and 6 where the discretizations for energy, space, and angle are essentially the same. Following this, the Transient Multilevel (TML) method is derived in detail. Most of this description originates from Zhu et al. [27], [28]. Because thermal-hydraulic feedback is an essential part of the transient analysis, a discusssion of the TML method with feedback is also included at the end.

11.1.1 3D Time-dependent Neutron Transport Equations

The time-dependent multigroup neutron transport equation is given in Eq. (11.1), and the precursor equation is given in Eq. (11.2).

$$\begin{aligned}\frac{1}{v_g}\frac{\partial \psi_g(\mathbf{r},\mathbf{\Omega},t)}{\partial t} &= -\mathbf{\Omega}\cdot\nabla\psi_g(\mathbf{r},\mathbf{\Omega},t) - \Sigma_{t,g}(\mathbf{r},t)\psi_g(\mathbf{r},\mathbf{\Omega},t) \\ &\quad + \sum_{g'=1}^{G}\int_0^{4\pi}\Sigma_{s,g'\to g}(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}',t)\psi_{g'}(\mathbf{r},\mathbf{\Omega}',t)d\Omega' \\ &+ \frac{1}{4\pi}\left(\chi_{p,g}(\mathbf{r},t)(1-\beta(\mathbf{r},t))S_F(\mathbf{r},t) + \chi_{d,g}(\mathbf{r},t)S_d(\mathbf{r},t)\right), \end{aligned} \tag{11.1}$$

$$\frac{\partial C_\tau(\mathbf{r},t)}{\partial t} = \beta_\tau(\mathbf{r},t)S_F(\mathbf{r},t) - \lambda_\tau C_\tau(\mathbf{r},t) \quad ,\tau = 1,2,...,6 \tag{11.2}$$

where ψ_g and $\Sigma_{x,g}$ have the usual definitions. The subscript τ is the delayed group index, C_τ is the delayed neutron precursor concentration, β_τ is the delayed neutron fraction, λ_τ is the delayed group decay constant, and v_g is the group velocity defined from its continuous form to the multi-group form by:

$$v_g = \frac{\int_{E_g}^{E_{g-1}} dE}{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)}dE}. \tag{11.3}$$

χ_p and χ_d are the prompt and delayed fission spectrums, respectively S_F and S_d are the total fission source and the delayed neutron source defined as:

$$S_F(\mathbf{r},t) = \frac{1}{k_{\text{eff}}} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\mathbf{r},t)\phi_{g'}(\mathbf{r},t), \tag{11.4}$$

$$S_d(\mathbf{r},t) = \sum_{\tau=1}^{6} \lambda_\tau C_\tau(\mathbf{r},t). \tag{11.5}$$

In the above equations, the S_F value is adjusted by the eigenvalue determined in the steady-state calculation and is used to initialize the transient. This is necessary to ensure a null reactivity from the initial conditions. Various time-discretization methods may be used for solving the time-dependent neutron transport equation, including the θ-method, the backward differentiation formula (BDF) method [30], etc. A detailed discussion of these methods can be found in [45]. The most commonly used second order θ-method requires the explicit calculation and storage of the RHS terms of Eq. (11.1). However, storage of the angularly dependent terms would be a considerable computational expense for practical problems. Therefore, the more computationally efficient and numerically stable backward Euler method is used for solving the time-dependent transport equation.

For a given time step size at time step n, the LHS of Eq. (11.1) can be discretized using the backward Euler method as:

$$\frac{\psi_g^n(\mathbf{r},\mathbf{\Omega}) - \psi_g^{n-1}(\mathbf{r},\mathbf{\Omega})}{v_g \Delta t_n} = R_g^n(\mathbf{r},\mathbf{\Omega}), \tag{11.6}$$

with R_g^n indicating all right-hand side terms of Eq. (11.1) at time step n.

Because of the practical difficulty for explicitly solving the angularly dependent time-derivative term, the isotropic approximation is used as defined in Eq. (11.7). This approximation assumes that the derivative of angular flux with respect to time is isotropic. It has been shown to be sufficiently accurate [86],[45], and is widely used in current state-of-art time-dependent transport solver codes [32],[77], [26].

$$\frac{\psi_g^n(\mathbf{r},\mathbf{\Omega}) - \psi_g^{n-1}(\mathbf{r},\mathbf{\Omega})}{v_g \Delta t_n} \approx \frac{\phi_g^n(\mathbf{r},\mathbf{\Omega}) - \phi_g^{n-1}(\mathbf{r},\mathbf{\Omega})}{4\pi v_g \Delta t_n}. \tag{11.7}$$

11.1.2 Precursor Integration and Formulation of the Transient Fixed Source Problem

The transport equation, Eq. (11.1), is coupled to the precursor equations Eq. (11.2). To avoid the complexity of solving this coupled set of time-dependent partial differential equations, the precursor equations are first integrated analytically and then substituted into Eq. (11.1). The rest of this section describes this procedure.

To allow for the integration of Eq. (11.1) the second-order approximation for the fission source is assumed:

$$\begin{aligned}\beta_\tau(\mathbf{r},t)S_F(\mathbf{r},t) \approx & \beta^n(\mathbf{r})S_F^n(\mathbf{r})\frac{\tilde{t}^2+\tilde{t}\gamma\Delta t_n}{(1+\gamma)(\Delta t_n)^2} \\ & +\beta^{n-1}(\mathbf{r})S_F^{n-1}(\mathbf{r})\left(1-\frac{\tilde{t}^2+\tilde{t}^2(\gamma-1)\Delta t_n}{\gamma(\Delta t_n)^2}\right) \\ & +\beta^{n-2}(\mathbf{r})S_F^{n-2}(\mathbf{r})\frac{\tilde{t}^2+\tilde{t}\Delta t_n}{(1+\gamma)\gamma(\Delta t_n)^2},\end{aligned} \tag{11.8}$$

where

$$\tilde{t} = t - t_{n-1},$$
$$\gamma = \frac{\Delta t_{n-1}}{\Delta t_n}.$$

Effectively this assumption states that the fission source will vary quadratically in a time step. That quadratic shape may be reconstructed using information from previous time steps. For the first two time steps, the initial condition is used as the previous steps fission source.

Applying an integrating factor to Eq. (11.2) gives

$$e^{\lambda_\tau t}\frac{\partial C_\tau(\mathbf{r},t)}{\partial t}+\lambda_\tau e^{\lambda_\tau t}C_\tau(\mathbf{r},t)=\beta_\tau(\mathbf{r},t)S_F(\mathbf{r},t)e^{\lambda_\tau t}, \tag{11.9}$$

and substituting by Eq. (11.8) into Eq. (11.9) one can perform the integration analytically over time step n. After some algebra the solution for the integrated precursor concentrations is obtained as a function of S_F^n.

$$\begin{aligned}C_\tau^n(\mathbf{r}) = & \tilde{\Omega}_\tau^0(\tilde{\lambda}_\tau^n)C_\tau^{n-1}(\mathbf{r})+\frac{1}{\lambda_\tau^n} \\ & \left(\beta^n(\mathbf{r})S_F^n(\mathbf{r})\tilde{\Omega}_\tau^n(\tilde{\lambda}_\tau^n)+\beta^{n-1}(\mathbf{r})S_F^{n-1}(\mathbf{r})\tilde{\Omega}_\tau^{n-1}(\tilde{\lambda}_\tau^n)+\beta^{n-2}(\mathbf{r})S_F^{n-2}(\mathbf{r})\tilde{\Omega}_\tau^{n-2}(\tilde{\lambda}_\tau^n)\right),\end{aligned} \tag{11.10}$$

where the following expressions have been used to simplify Eq. (11.10)

$$\tilde{\lambda}_\tau^n = \lambda_\tau^n \Delta t_n, \tag{11.11a}$$

$$\kappa_0(x) = 1-e^{-x}, \tag{11.11b}$$

$$\kappa_1(x) = 1-\frac{\kappa_0(x)}{x}, \tag{11.11c}$$

$$\kappa_2(x) = 1-\frac{2\kappa_1(x)}{x}, \tag{11.11d}$$

$$\tilde{\Omega}_\tau^0(\tilde{\lambda}_\tau^n) = e^{-\tilde{\lambda}_\tau^n}, \tag{11.11e}$$

$$\tilde{\Omega}_\tau^n(\tilde{\lambda}_\tau^n) = \frac{\kappa_2(\tilde{\lambda}_\tau^n)+\gamma\kappa_1(\tilde{\lambda}_\tau^n)}{1+\gamma}, \tag{11.11f}$$

$$\Omega_\tau^{n-1}(\tilde{\lambda}_\tau^n) = \kappa_0(\tilde{\lambda}_\tau^n)-\frac{\kappa_2(\tilde{\lambda}_\tau^n)+(\gamma-1)\kappa_1(\tilde{\lambda}_\tau^n)}{\gamma}, \tag{11.11g}$$

$$\Omega_\tau^{n-2}(\tilde{\lambda}_\tau^n) = \frac{\kappa_2(\tilde{\lambda}_\tau^n)-\kappa_1(\tilde{\lambda}_\tau^n)}{\gamma(1+\gamma)}. \tag{11.11h}$$

Next the solution of the precursor equations obtained as Eq. (11.10) is substituted into Eq. (11.5) yielding:

$$S_d^n(\mathbf{r}) = \tilde{\Omega}^n(\mathbf{r}) S_F^n(\mathbf{r}) + \tilde{S}_d^{n-1}(\mathbf{r}), \tag{11.12}$$

where

$$\tilde{\Omega}^n(\mathbf{r}) = \sum_{\tau=1}^{6} \beta_\tau(\mathbf{r}) \tilde{\Omega}_\tau^n(\tilde{\lambda}_\tau^n), \tag{11.13}$$

$$\tilde{S}_d^n(\mathbf{r}) = \sum_{\tau=1}^{6} \lambda_\tau \tilde{\Omega}_\tau^0(\tilde{\lambda}_\tau^n) C_\tau^{n-1}(\mathbf{r}) + S_F^{n-1} \sum_{\tau=1}^{6} \beta_\tau^{n-1}(\mathbf{r}) \tilde{\Omega}_\tau^{n-1}(\tilde{\lambda}_\tau^n) + S_F^{n-2}(\mathbf{r}) \sum_{\tau=1}^{6} \beta_\tau^{n-2}(\mathbf{r}) \tilde{\Omega}_\tau^{n-2}(\tilde{\lambda}_\tau^n). \tag{11.14}$$

By inserting the delayed neutron source terms of Eqs. (11.14) into Eq. (11.1), utilzing the time discretization of Eq. (11.7) and explicitly writing out $R_g^n(\mathbf{r},\mathbf{\Omega})$ of Eq. (11.6) the final form of the discretized transport transient fixed source problem may be written as:

$$\begin{aligned}\mathbf{\Omega}\nabla\psi_g^n(\mathbf{r},\mathbf{\Omega}) + \Sigma_t^n \psi_g^n(\mathbf{r},\mathbf{\Omega}) = &\sum_{g'=1}^{G} \int_0^{4\pi} \Sigma_{s,g'\to g}^n(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}') \psi_{g'}^n(\mathbf{r},\mathbf{\Omega}') d\mathbf{\Omega}' \\ &+ \frac{1}{4\pi}\left(\chi_g^n(\mathbf{r}) S_F^n(\mathbf{r}) + S_{tr,g}^n(\mathbf{r})\right),\end{aligned} \tag{11.15}$$

where $\chi_g^n(\mathbf{r})$ is defined as

$$\chi_g^n(\mathbf{r}) = \chi_{p,g}^n(\mathbf{r})(1-\beta(\mathbf{r})) + \chi_{d,g}^n(\mathbf{r})\beta(\mathbf{r}), \tag{11.16}$$

and the transient source given in Eq. (11.15) is defined as:

$$S_{tr,g}^n(\mathbf{r}) = A_g^n(\mathbf{r})\phi_g^n + B_g^n(\mathbf{r}) S_F^n + C_g^n(\mathbf{r}). \tag{11.17}$$

In the previous equation, $A_g^n(\mathbf{r})$ and $B_g^n(\mathbf{r})$ are the flux and fission source dependent coefficients,respectively, and $C_g^n(\mathbf{r})$ is a constant coefficient depending on only previous time-step quantities. These terms are given as:

$$A_g^n(\mathbf{r}) = -\frac{1}{v_g \Delta t_n}, \tag{11.18a}$$

$$B_g^n(\mathbf{r}) = \chi_{d,g}^n(\mathbf{r})(\tilde{\Omega}^n(\mathbf{r}) - \beta(\mathbf{r})), \tag{11.18b}$$

$$C_g^n(\mathbf{r}) = \chi_{d,g}^n(\mathbf{r}) \tilde{S}_d^{n-1}(\mathbf{r}) + \frac{\phi_g^{n-1}(\mathbf{r})}{v_g \Delta t_n}. \tag{11.18c}$$

The form of Eq. (11.15) should be familiar, it is written in such a way that it may readily solved any standard steady-state neutron transport solver after discretizing over space, angle. The 2D MOC solution of Eq. (11.15) is discussed next in Section 11.1.3.

11.1.3 2D MOC Solution of the Transient Fixed Source Problem

The discretized steady-state trasnport equation from Eq (5.3) from Chapter 5 is reproduced below, where the subscript k has been dropped for simplicity, and the index for the time step, n has been added.

$$\Omega_{x,m} \frac{\partial \psi_{m,g}^n(x,y)}{\partial x} + \Omega_{y,m} \frac{\partial \psi_{m,g}^n(x,y)}{\partial y} + \Sigma_{t,g}^n(x,y) \psi_{m,g}^n(x,y) = q_{m,g}^n(x,y). \tag{11.19}$$

Eq. (11.15) can be written in such a form by defining $q^n_{m,g}$ as:

$$q^n_{m,g}(x,y) = \sum_{g'=1}^{G}\sum_{m'=1}^{M}\Sigma^n_{s,g'\to g}(x,y,\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m)\psi^n_{m',g'}(x,y)w_{m'} + \frac{1}{4\pi}\left[\chi^n_g(x,y)S^n_F(x,y)+A^n_g(x,y)\phi^n_g+B^n_g(x,y)S^n_F+C^n_g(x,y)\right] - \frac{1}{4\pi\Delta z}\left[J^n_{z,T,g}(x,y)-J^n_{z,B,g}(x,y)\right]. \quad (11.20)$$

In this formulation of the 2D MOC problem note that all the coefficients in Eq. (11.20) are averaged over the axial direction z.

$$A^n_g(x,y) = \frac{\int_{z_B}^{z_T} A^n_g(x,y,z)\phi^n_{g,m}(x,y,z)dz}{\int_{z_B}^{z_T}\phi^n_{g,m}(x,y,z)dz}, \quad (11.21a)$$

$$B^n_g(x,y) = \frac{\int_{z_B}^{z_T} B^n_g(x,y,z)S^n_F(x,y,z)dz}{\int_{z_B}^{z_T} S^n_F(x,y,z)dz}, \quad (11.21b)$$

$$C^n_g(x,y) = \frac{\int_{z_B}^{z_T} C^n_g(x,y,z)dz}{\int_{z_B}^{z_T} dz}. \quad (11.21c)$$

Since the MOC solution of Eq (5.3) in Chapter 5 is done in terms of the variable $q_{m,g}$ the steps for obtaining the MOC solution of Eq. (11.19) are identical.

11.1.4 Transient 1D Nodal Method

The transient 1D equations of the 2D-1D method are formulated and solved by essentially the same procedure as section 11.1.3. The steady-state 1D equation Eq. (4.9) from Chapter 4 is rewritten below.

$$\mu\frac{\partial\hat{\psi}}{\partial z}(\mathbf{r},\mu)+\Sigma_t\hat{\psi}(\mathbf{r},\mu) = \frac{\Sigma_s}{2}\int_{-1}^{1}\hat{\psi}(\mathbf{r},\mu')d\mu' + \frac{\nu\Sigma_f}{2k_{\text{eff}}}\int_{-1}^{1}\hat{\psi}(\mathbf{r},\mu')d\mu' - \frac{1}{2}\left[\frac{\partial J_x}{\partial x}(\mathbf{r})+\frac{\partial J_y}{\partial y}(\mathbf{r})\right]. \quad (11.22)$$

The discretizations yielding Eq. (11.15) are still amenable to the 2D-1D procedure, and thus the transient form of Eq. (11.22) is simply:

$$\mu\frac{\partial\hat{\psi}^n}{\partial z}(\mathbf{r},\mu)+\Sigma^n_t\hat{\psi}^n(\mathbf{r},\mu) = \frac{\Sigma^n_s}{2}\int_{-1}^{1}\hat{\psi}^n(\mathbf{r},\mu')d\mu' + \frac{\nu\Sigma^n_f}{2k_{\text{eff}}}\int_{-1}^{1}\hat{\psi}^n(\mathbf{r},\mu')d\mu' - \frac{1}{2}\left[\frac{\partial J^n_x}{\partial x}(\mathbf{r})+\frac{\partial J^n_y}{\partial y}(\mathbf{r})\right] + S^n_t r(\mathbf{r}). \quad (11.23)$$

As noted in Chapter 6 the solution of equations of the form of Eq. (11.23) can be performed using either a transport method or a diffusion method. Either the NEM (section 6.3) or NEM-P_3 (section 6.5) kernel can be used for transient calculations in MPACT. These kernels are modified to include the transient source of Eq. (11.17). In these kernels the

same basis polynomials used to expand the steady-state components of the source in the axial direction are used to expand the transient source.

$$S^n_{tr,g,m}(\xi) = \sum_{k=0}^{2} S_{tr,k} P_k(\xi), \tag{11.24}$$

where ξ denotes the normalized spatial variable and $P_k(\xi)$ are basis polynomials.

The coefficients of the transient source term are calculated the same way as the transverse leakage term. This forces the averaged transient source in Eq. (11.24) to be identical to the transient source at three adjacent nodes.

11.1.5 CMFD Transient Fixed Source Problem

Several methods exist for solving the CMFD TFSP. These include a multi-group sweeping (MGS) CMFD, and a multi-group matrix (MGM) CMFD. The MGS CMFD method solves the CMFD equations group by group, and requires multiple iterations over energy to converge the energy-dependent source, especially for transient problems with a large power change. The MGM CMFD formulates one single matrix for the whole multi-group 3D CMFD problem and uses PETSc [6] to solve the linear system. The advantage of the MGM CMFD is that the solution of the TFSP is obtained by solving the MGM linear system, rather than by solving multiple linear systems as is necessary in the MGS formulation. The MGM CMFD is used in this work, since for all problems analyzed thus far, the MGM CMFD performs much more efficiently than MGS CMFD.

The derivation of the CMFD operator is essentially the same procedure for transient as it is for steady-state. The discretized *steady-state* CMFD operator given in Eq. (7.2) is rewritten below:

$$\sum_s -\tilde{D}_{g,j,s}\left(\phi_{g,j} - \phi_{g,j,s}\right) + \hat{D}_{g,j,s}\left(\phi_{g,j} + \phi_{g,j,s}\right)\delta A_{j,s} + \Sigma_{t,g,j}\phi_{g,j}V_j = \left[\sum_{g'=1}^{G}\left(\Sigma_{s0,g'\rightarrow g,j} + \frac{\chi_g}{k_{\text{eff}}}\nu\Sigma_{f,g',j}\right)\phi_{g',j}\right]V_j, \tag{11.25}$$

where $\tilde{D}$ and $\hat{D}$ have the usual definitions from Eq. (7.4) and Eq. (7.7). $\delta A_{j,s}$ in Eq. (11.25) and Eq. (11.26) is the surface area of face s of coarse cell j.

With the addition of the transient source the resulting equation is:

$$\sum_s -\tilde{D}^n_{g,j,s}\left(\phi^n_{g,j} - \phi^n_{g,j,s}\right) + \hat{D}^n_{g,j,s}\left(\phi^n_{g,j} + \phi^n_{g,j,s}\right)\delta A_{j,s} + \Sigma^n_{t,g,j}\phi^n_{g,j}V_j = \left[\sum_{g'=1}^{G}\left(\Sigma^n_{s0,g'\rightarrow g,j} + \frac{\chi^n_g}{k_{\text{eff}}}\nu\Sigma^n_{f,g',j}\right)\phi^n_{g',j}\right]V_j + S^n_{tr,g,j}V_j. \tag{11.26}$$

Instead of homogenizing the fine mesh transient source term ($S^n_{tr,g}$ in Eq. (11.26)), the transient source coefficients A,

B, and C are homogenized using the following equations:

$$A^n_{g,j} = \frac{\sum_{i\in j} A^n_{g,i}\phi^n_{g,i}V_i}{\sum_{i\in j}\phi^n_{g,i}V_i}, \tag{11.27a}$$

$$B^n_{g,j} = \frac{\sum_{i\in j} B^n_{g,i}\chi^n_g S_{F,g,i}V_i}{\sum_{i\in j}\chi^n_g S_{F,g,i}V_i}, \tag{11.27b}$$

$$C^n_{g,j} = \frac{\sum_{i\in j} C^n_{g,i}V_i}{\sum_{i\in j} V_i}. \tag{11.27c}$$

The solution of the aforementioned equations is next described in subsections 11.1.5.1 and 11.1.5.2. To facilicate the notation in these sections we now introduce the following *operator* notation.

$$\text{Removal term:} \quad \mathbf{M} = \mathbf{D} + \boldsymbol{\Sigma}_t \tag{11.28a}$$

$$\text{Scattering:} \quad \mathbf{S} = \boldsymbol{\Sigma}_s \tag{11.28b}$$

$$\text{Fission source:} \quad \mathbf{F} = \frac{\boldsymbol{\chi}\boldsymbol{\nu}\boldsymbol{\Sigma}_f}{k_{\text{eff}}} \tag{11.28c}$$

$$\text{Transient source:} \quad \mathbf{S_{tr}} = \mathbf{A}\phi + \mathbf{BF}\phi + \mathbf{C} \tag{11.28d}$$

Here $\boldsymbol{\Sigma}_t$ is a diagonal matrix, and $\mathbf{D}$ is a sparse matrix containing the $\tilde{D}$ and $\hat{D}$ terms.

11.1.5.1 MGS CMFD Formulation

The iteration scheme for multi-group sweeping formulates the operators of Eq. (11.28) for all coarse mesh cells and a single group g and iterates through these linear systems in a Gauss-Seidel fashion. As a result, the CMFD transient source can be updated by the new CMFD flux and fission source as:

$$(\mathbf{M_g} - \mathbf{S_g})\phi_{\mathbf{g}}^{\ell+\mathbf{1}} = \mathbf{F_g}\phi_{\mathbf{g}}^{\ell} + \mathbf{S}_{\mathbf{tr,g}}^{\ell}, \tag{11.29a}$$

$$\mathbf{S}_{\mathbf{tr,g}}^{\ell+\mathbf{1}} = \mathbf{A_g}\phi_{\mathbf{g}}^{\ell+\mathbf{1}} + \mathbf{B_g F}\phi_{\mathbf{g}}^{\ell+\mathbf{1}} + \mathbf{C_g}. \tag{11.29b}$$

The overall solution algorithm is shown in Figure 11.1. The MGS CMFD iteration technique is similar to the traditional source iteration technique. The motivation for going with this type of iteration is to minimize storage requirements for linear system solved in Eq. (11.29) and also to keep the condition number from getting too large. One disadvantage of the MGS method is that it requires multiple iterations on the CMFD linear systems to converge before each MOC sweep, and the number of iterations can be very large for transient problems with rapid changes in the power since the right-hand side (fission source and transient source) can change considerably during each transient step.

11.1.5.2 MGM CMFD Formulation

As an alternative to the MGS CMFD, the MGM CMFD method was developed. This method formulates the transient fixed source problem for the full space-energy linear system as:

$$(\mathbf{M}-\mathbf{S}-\mathbf{F}-\mathbf{A}-\mathbf{BF})\phi=\mathbf{C}. \tag{11.30}$$

As noted previously, the transient calculation does not require updating the eigenvalue during a transient step. Therefore, the CMFD flux-dependent source (including the transient source and fission source) on the RHS in the transient matrix can be moved to the left-hand side. In this approach no source iterations are required. Once the linear system is solved the flux solution is obtained. Eq. (11.30) is a standard linear system and can be solved using any matrix inversion method. Given the size and sparsity of the matrix, the GMRES solver with a block Jacobi preconditioner from PETSc that is used for the steady solution of the CMFD linear system is also used for the transient fixed source problem.

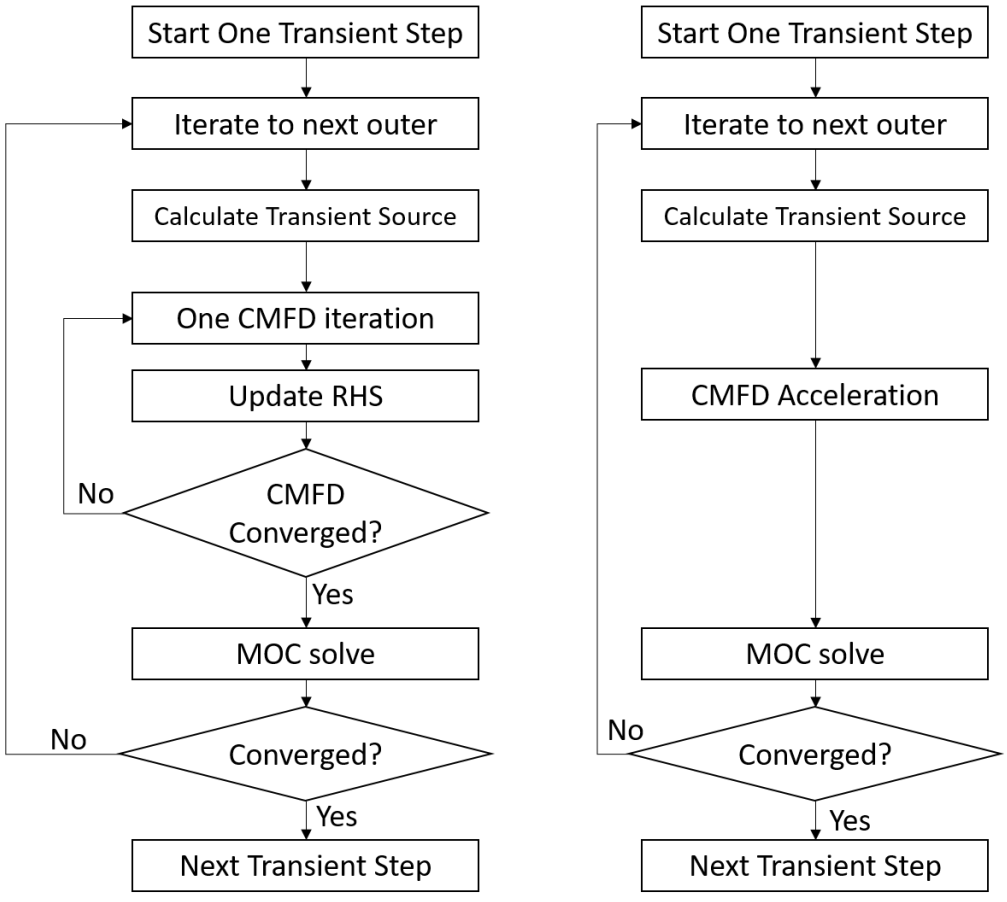


Figure 11.1. Flow charts for MGS (left) and MGM (right) CMFD.

11.1.6 Iteration Strategy

The overall iteration scheme of the MPACT transient algorithm is shown in Figure 11.2. The 2D MOC transport problem or the 2D MOC/1D NEM problem are iteratively solved with 2D or 3D CMFD acceleration until the convergence criteria are satisfied.

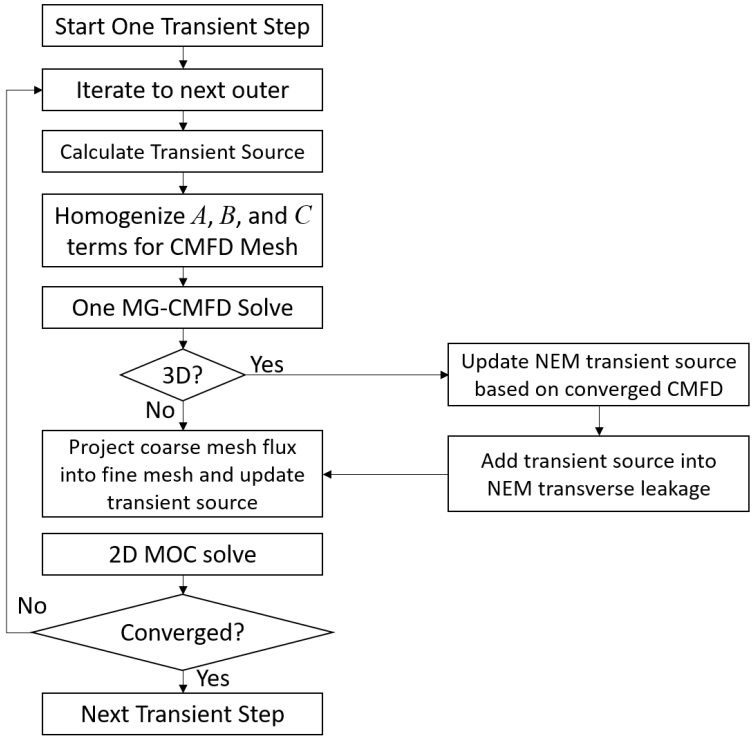


Figure 11.2. 2D-1D transient iteration scheme.

11.2 Transient Multilevel (TML) Method

The design and implementation of the 2D/1D scheme and the MGM CMFD acceleration method considerably reduce the computational burden for full core transient modeling with pin-resolved detail. However, the required CPU effort is still too large for practical applications. This chapter discusses the innovative methods used to further reduce the computational time by increasing the size of the time step required for the time-dependent MOC method. A transient multilevel (TML) method based on the predictor-corrector quasi-static method (PCQM) was developed in which the first level involves the 3D transport with CMFD acceleration. The second level is a pure 3D CMFD TFSP. Finally the coarsest, third level makes use of the exact point kinetic equations (EPKE). The essential idea of the method is to take advantage of the differences in the time variation of the angular, spatial, and magnitude components of the flux in the reactor after a reactivity change.

This chapter introduces the adjoint flux necessary to couple the spatial (CMFD) and amplitude (EPKE) flux levels, and then provides the details of the derivation and implementation of the TML method.

11.2.1 CMFD Adjoint Flux

The fundamental mode adjoint neutron flux has always been useful in reactor physics for the treatment of perturbations of eigenvalue problems. The adjoint flux provides a convenient method to estimate the perturbed eigenvalue without exactly solving the often very complicated perturbed systems [72]. One of the important perturbations for practical neutron transport applications is the insertion of reactivity, where the change of material composition, temperature, etc., will either increase or decrease the eigenvalue of the system.

The MOC-based adjoint flux calculation was previously implemented in MPACT to generate the asymptotic diffusion coefficient [88]. Because of the computational complexity of the MOC-based adjoint flux, the CMFD-based adjoint flux calculation capability was also designed and implemented in MPACT [25] to accurately approximate the MOC-based adjoint flux for practical reactor core applications. The detailed comparison of the MOC and CMFD adjoint flux can be found in [25]. As described in Ott and Neuhold [72], the adjoint operator is defined by the scalar product Eq. (11.31) to hold for all allowed ψ and ϕ of the functional space, where $\mathbf{H}^*$ is the adjoint operator of $\mathbf{H}$ and $<>$ is the operation to integrate over all space:

$$< \mathbf{\Psi}, \mathbf{H\Phi} >=< \mathbf{H}^*\mathbf{\Psi}, \mathbf{\Phi} >=< \mathbf{\Phi}, \mathbf{H}^*\mathbf{\Psi} > . \tag{11.31}$$

The steady-state multigroup CMFD equation expressed in operator notation is:

$$(\mathbf{M}-\mathbf{S})\phi = \frac{1}{k_{\text{eff}}}\mathbf{F}\phi. \tag{11.32}$$

As shown in Ott and Neuhold [72], the adjoint to Eq. (11.32) that satisfies Eq. (11.31) can be obtained by solving the adjoint equation expressed as:

$$(\mathbf{M}-\mathbf{S})^*\phi^* = \frac{1}{k^*_{\text{eff}}}\mathbf{F}^*\phi^*, \tag{11.33}$$

where the fundamental mode forward (k_{eff}) and adjoint (k^*_{eff}) eigenvalues are identical.

The matrix $(\mathbf{M}-\mathbf{S})$ is explicitly constructed and stored during the forward CMFD solve in MPACT, and the transpose operation is simply performed by the intrinsic transpose subroutine provided by PETSc [6]. The fission source operator $\mathbf{F}$ is not constructed explicitly, but rather can be written in matrix form in Eq. (11.34), and the transpose operation can be performed by simply switching the fission spectrum and fission cross section vector as shown in Eq. (11.35):

$$\mathbf{F} = \chi\nu\mathbf{\Sigma_f}, \tag{11.34}$$

$$\mathbf{F}^* = \nu\mathbf{\Sigma}^*_{\mathbf{f}}\chi^*. \tag{11.35}$$

Equation (11.33) can also be solved using the power iteration, which is the same procedure to solve the forward CMFD equation. Since the matrices $(\mathbf{M}-\mathbf{S})$ and $\mathbf{F}$ are fixed for the given problem, there is no need to update them during the power iteration. In addition, the majority of the RHS term can be shifted to the LHS since the eigenvalue is known.

11.2.2 Point Kinetics Equations

The point kinetics equations (PKEs) are derived by integrating the equations defined in Eq. (11.1) and Eq. (11.2). This procedure first integrates over angle, $\mathbf{\Omega}$, then uses the integration defined in Eq. (11.31) to integrate over space and energy. The detailed derivation is provided in Dulla et al. [35, 23]. The result of this procedure yields the EPKE's as:

$$\frac{dp(t)}{dt} = \frac{\rho(t)-\beta^{eff}(t)}{\Lambda(t)}p(t) + \frac{1}{\Lambda(0)}\sum_{\tau}\lambda_{\tau}(t)\zeta_{\tau}(t), \tag{11.36}$$

$$\frac{d\zeta_{\tau}(t)}{dt} = \frac{\Lambda(0)}{\Lambda(t)}\beta_{\tau}^{eff}(t)p(t) - \lambda_{\tau}^{PK}(t)\zeta_{\tau}(t) \quad , \quad \tau = 1,2,...,6, \tag{11.37}$$

where $p(t)$ represents the core wise amplitude function, and $\zeta_{\tau}(t)$ is the adjoint flux weighted precursor number density for delayed group τ.

The reactivity, delayed neutron fractions, neutron generation time, and delayed neutron constants are defined in Eqs. (11.38) through (11.42):

$$\rho(t) = \frac{<\phi^*(\mathbf{r},E)(\mathbf{F}-\mathbf{M})\phi(\mathbf{r},E,t)>}{F(t)} \quad , \tag{11.38}$$

$$\beta_{\tau}^{eff}(t) = \frac{<\phi^*(\mathbf{r},E)\chi_{d,\tau}(\mathbf{r},E)\beta_{\tau}(\mathbf{r})S_F(\mathbf{r},t)>}{F(t)}, \quad \tau = 1,2,...,6, \tag{11.39}$$

$$\beta^{eff}(t) = \sum_{\tau}\beta_{\tau}^{eff}(t), \tag{11.40}$$

$$\Lambda(t) = \frac{<\phi^*(\mathbf{r},E)\frac{1}{v(E)}\phi(\mathbf{r},E,t)>}{F(t)}, \tag{11.41}$$

$$\lambda_{\tau}^{PK}(t) = \frac{<\phi^*(\mathbf{r},E)\lambda_{\tau}(\mathbf{r},t)\chi_{d,\tau}(\mathbf{r},E)C_{\tau}(\mathbf{r},t)>}{<\phi^*(\mathbf{r},E)\chi_{d,\tau}(\mathbf{r},E)C_{\tau}(\mathbf{r},t)>}, \tag{11.42}$$

where $F(t) =< \phi^*(\mathbf{r},E)\chi(\mathbf{r},E)S_F(\mathbf{r},t) >$, and the matrix operators are the same as those used in Eq. (11.28).

Eqs. (11.36) and (11.37) are solved using the same discretization applied to the transport transient equation, including implicit Euler discretization for the $dp(t)/dt$ term and a second order approximation of the amplitude function during a time step n to obtain an integrated form of Eq. (11.37). Therefore the EPKE solution of Eq. (11.37) corresponding to Eq. (11.10) is:

$$\zeta_{\tau}^{n} = \Omega_{\tau}^{PK,0}\zeta_{\tau}^{n-1} + \frac{\Lambda(0)}{\lambda_{\tau}^{PK}}\left[p^n\frac{\beta_{\tau}^{eff,n}}{\Lambda^n}\tilde{\Omega}_{\tau}^{PK,n-1} + p^n - 1\frac{\beta_{\tau}^{eff,n-1}}{\Lambda^{n-1}}\tilde{\Omega}_{\tau}^{PK,n-1} + p^n - 2\frac{\beta_{\tau}^{eff,n-2}}{\Lambda^{n-2}}\tilde{\Omega}_{\tau}^{PK,n-2}\right], \tag{11.43}$$

where $\tilde{\Omega}_{\tau}^{PK,i}$ denotes the corresponding $\tilde{\Omega}_{\tau}^{i}$ from Eq. (11.11), except they are evaluated with the new effective delayed constant defined in Eq. (11.42) rather than region-wise delayed constants.

By applying the implicit Euler discretization to $dp(t)/dt$ and then inserting Eq. (11.43) into Eq. (11.36), the current step power can be obtained, and the precursor equation is then calculated by inserting the power back into Eq. (11.43).

11.2.3 Transient Multilevel Method

The essential idea of the transient multilevel (TML) method is to capture the flux change in space, energy, and angle in a time domain consistent with its physical variation during a transient. As illustrated in Figure 11.3, the time step of the 3D transport transient solver is coarse since the time variation of the angular sub-pin flux distribution is generally slower than the changes in the spatial shape and amplitude of the flux. The 3D CMFD transient solver uses the 3D whole core matrix in the intermediate time step to maintain the accuracy of the pin-wise scalar flux distribution. Finally, the time step of the EPKE is the smallest in order to capture the time variation of the flux magnitude, which is driven by the prompt neutron generation time. In this manner, the three-level transient solver maintains a consistent accuracy, while minimizing the overall computational expense for the transient simulation of a large core problem. In the following sections, the methods developed use the PCQM scheme to couple 3D transport to the 3D CMFD and the 3D CMFD to the EPKE.

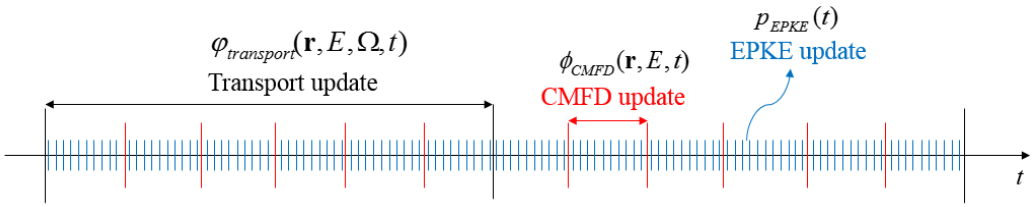


Figure 11.3. Illustration of TML scheme

11.2.3.1 3D Transport and CMFD Coupling

The coupling of the 3D-transport and 3D-CMFD equations begins by factoring the neutron angular flux into an amplitude and shape function:

$$\psi_g(\mathbf{r},\mathbf{\Omega},t) = P_g(\mathbf{r},t)\Phi_g(\mathbf{r},\mathbf{\Omega},t), \tag{11.44}$$

where $P_g(\mathbf{r},t)$ is the amplitude function and $\Phi_g(\mathbf{r},\mathbf{\Omega},t)$ is the shape function, representing the angular and fine mesh flux distribution. If $P_g(\mathbf{r},t)$ is spatially flat for a coarse mesh j, Eq. (11.44) can be expressed as:

$$\psi_g(\mathbf{r},\mathbf{\Omega},t) = P_{g,j}(t)\Phi_g(\mathbf{r},\mathbf{\Omega},t) \quad , \quad \mathbf{r} \in j. \tag{11.45}$$

The amplitude and shape functions defined in Eq. (11.45) are arbitrary, and thus a constraint for the shape function is introduced which requires the integral of the shape function in each CMFD cell ($\mathbf{r} \in j$) domain to be unity:

$$\frac{1}{\int\limits_{\mathbf{r}\in j} dV} \int\limits_{\mathbf{r}\in j} \int\limits_{\mathbf{\Omega}} \Phi_g(\mathbf{r},\mathbf{\Omega},t) d\mathbf{\Omega}, dV = 1. \tag{11.46}$$

If Eq. (11.45) is integrated over space and angle in the CMFD cell domain and the constraint in Eq. (11.46) is applied, the amplitude function in each CMFD cell is identical to the corresponding CMFD scalar flux:

$$\frac{1}{V_j}\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\psi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV=\frac{1}{V_j}P_{j,g}(t)\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\Phi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV=P_{g,j}(t). \tag{11.47}$$

As a result, the shape function in each CMFD cell can be easily obtained by Eq. (11.48).

$$\Phi_g(\mathbf{r},\mathbf{\Omega},t)=\frac{\psi_g(\mathbf{r},\mathbf{\Omega},t)}{\phi_{g,j}(t)}, \tag{11.48}$$

where $\phi_{g,j}(t)$ is the scalar flux in the coarse mesh.

If Eq. (11.1) is integrated over angle and space in each CMFD cell j, and by using the constraint in Eq. (11.46) again, the left hand side reduces to Eq. (11.49).

$$\begin{aligned}
&\frac{1}{V_j}\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\frac{\partial\psi_g(\mathbf{r},\mathbf{\Omega},t)}{v_g\partial t}d\mathbf{\Omega}dV\\
&=\frac{1}{V_jv_g}\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\left(P_{g,j}(t)\frac{\partial\Phi_g(\mathbf{r},\mathbf{\Omega},t)}{\partial t}+\Phi_g(\mathbf{r},\mathbf{\Omega},t)\frac{\partial P_{g,j}(t)}{\partial t}\right)d\mathbf{\Omega}dV\\
&=\frac{1}{V_jv_g}P_{g,j}(t)\frac{\partial}{\partial t}\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\Phi_g(\mathbf{r},\mathbf{\Omega},t)+\frac{1}{V_jv_g)}\frac{\partial P_i(E,t)}{\partial t}\int_{\mathbf{r}\in j}\int_{\mathbf{\Omega}}\Phi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV\\
&=\frac{1}{v_g}\frac{\partial P_{g,j}(t)}{\partial t}.
\end{aligned} \tag{11.49}$$

The right hand side would be the same as Eq. (7.1)

The above derivation indicates that the CMFD amplitude function can be evaluated using exactly the same diffusion equation as in Eq. (11.26), where the coupling coefficients are generated during the transport/CMFD solution step as previously defined. This similar approach for solving the coarse mesh flux amplitude is referred to as the TCMFD method [27].

The PCQM algorithm for transport/CMFD level is thus:

- Step 1: Evaluate the steady state flux distribution based on the 3D transport solver.
- Step 2: Solve the transport TFSP for coarse time step dt_n given by Eq. (11.15) and accelerated by CMFD.
  - The flux distribution $\psi_g^n(\mathbf{r},\mathbf{\Omega})$ in this step is predicted, and it is assumed the shape function is accurate in the coarse time step, but the amplitude function has some inaccuracy and will be corrected using fine time steps.
  - The shape function is generated by using the predicted CMFD scalar flux defined in Eq. (11.48), and the CMFD coefficients at the end of the time step are calculated.
- Step 3: Linearly interpolate the CMFD coefficients between t_{n-1} and t_n, build the CMFD transient source based on Eq. (11.17), and solve the CMFD TFSP using the $dt_{n,CMFD}=dt_n/N$ time step, where N is the number of CMFD steps per transport step.

- Step 4: Multiply the CMFD scalar flux by the shape function to determine the corrected transport angular dependent flux distribution:

$$\psi_g^{C,n}(\mathbf{r},\mathbf{\Omega}) = \psi_g^{P,n}(\mathbf{r},\mathbf{\Omega})\frac{\phi_{g,j}^{C,n}}{\phi_{g,j}^{P,n}} \quad \mathbf{r} \in j. \tag{11.50}$$

- Step 5: Use the corrected transport flux distribution to solve the precursor equations in Eq. (11.10).

The same procedure is used to solve all transport time steps using steps 2–5.

11.2.3.2 3D CMFD and EPKE Coupling

Similar to the coupling of the transport/CMFD level, the CMFD flux is factorized as follows:

$$\phi_{g,j}(t) = p(t)\varphi_{g,j}(t). \tag{11.51}$$

where $\phi_{g,j}$ is the CMFD flux in cell j for group g, and $p(t)$ and $\varphi_{g,j}(t)$ represent the whole core amplitude and coarse mesh cell dependent shape functions, respectively. The constraint for this separation is to maintain the integration of the shape function to be a constant K:

$$K = \sum_j \sum_g \phi_{g,j}^* \frac{1}{v_g}\varphi_{g,j}(0) \quad = \quad \sum_j \sum_g \phi_{g,j}^* \frac{1}{v_g}\varphi_{g,j}(t). \tag{11.52}$$

With the constraint in Eq. (11.52), the amplitude can calculated as in Eq. (11.53), and the corresponding shape function is given by Eq. (11.54).

$$p(t) = \frac{\sum_j \sum_g \phi_{g,j}^* \frac{1}{v_g}\varphi_{g,j}(t)}{K}, \tag{11.53}$$

$$\varphi_{g,j}(t) = \frac{\phi_{g,j}(t)}{\sum_j \sum_g \phi_{g,j}^* \frac{1}{v_g}\varphi_{g,j}(t)}. \tag{11.54}$$

The amplitude function defined in Eq. (11.51) is obtained from the EPKE and is defined in Eq. (11.36) and Eq. (11.37). The detailed derivation can be found in Cho (2005) [32].

The PCQM algorithm for CMFD/EPKE level is then:

- Step 1: Evaluate the steady-state CMFD-based adjoint flux $\phi_{g,j}^*$ [25] in addition to the forward steady-state flux distribution.
- Step 2: Solve the CMFD TFSP for CMFD time step $dt_{n,CMFD}$ using Eq. (11.26).
  - The predicted CMFD scalar flux distribution $\phi_{g,j}^P(t_n)$ is computed, and it is assumed that the shape function is accurate in the coarse time step, but the amplitude function will be corrected using a finer time step.
  - The shape function $\varphi_{g,j}(t_n)$ is generated using Eq. (11.54). The point kinetics parameters used for solving the EPKE are calculated and stored for the end of the time step.

- Step 3: Linearly interpolate the point kinetics parameters between $t_{n-1,CMFD}$ and $t_{n,CMFD}$, and solve the EPKE using $dt_{n,EPKE} = dt_{n,CMFD}/N$ time step, where N is the number of EPKE step per CMFD step.
- Step 4: The final CMFD scalar flux is corrected by the $p(t_n)$:

$$\phi^C_{g,j}(t_n) = \phi^P_{g,j}(t_n)\frac{p(t_n)K}{\sum_j\sum_g \phi^*_{g,j}\frac{1}{v_g}\phi^P_{g,j}(t_n)}. \tag{11.55}$$

- Step 5: The corrected CMFD scalar flux is used to solve the precursor equation with Eq. (11.43).

The same procedure is used to solve all CMFD time steps using steps 2–5.

11.2.4 Iteration Scheme with TH Feedback

The TH feedback was applied to the neutronic solver in MPACT [29] using a simplified internal TH module. This simplified model includes 1D radial heat conduction and 1D axial mass/energy equations of conservation for convection. This model is primarily simplified in the convection aspect. Although several physical phenomena related to the conduction such as creep, swelling, thermal expansion, and the like are not treated explicitly. This model provides the pin-wise radially dependent Doppler feedback for the channel flow with the constant pressure assumption. In the steady-state condition, the TH/neutronics coupling is first fully converged, and then the transient TH/neutronics solvers are used to march through time using an *explicit* coupling. One note is that the due to the explicit TH/neutronics coupling and sub-pin Doppler feedback, special attention was required with the TML couplings. Additionally, no simplified transient "convection" model exists, so the adiabatic boundary condition is assumed at the clad surface.

The TH feedback may also be provided by COBRA-TF [75] which does include more sophisticated models for convection and fuel performance.

11.2.5 First Level TH Coupling for 3D Transport/3D CMFD

In each transport time step, the transport transient equation is solved first, and then the CMFD transient equation is solved to correct the pin-wise scalar flux. Due to the subpin-dependent Doppler temperature feedback, the TH calculation is performed in each transport time step; that means, at present TML is solely a *neutronics* accelerator. As shown in Figure 11.4, steps 0–2 represent the transport time steps, and each transport step is then subdivided into several fine CMFD time steps. Step 2 is used when the material is linearly changed from m_0 to m_5 (time evolves from t_0 to t_5).

To interpolate the CMFD matrix during the CMFD corrector step, two sets of CMFD macroscopic cross sections for Σ^n and Σ^{n+1} are required, one each at the beginning and end of a transport time step.

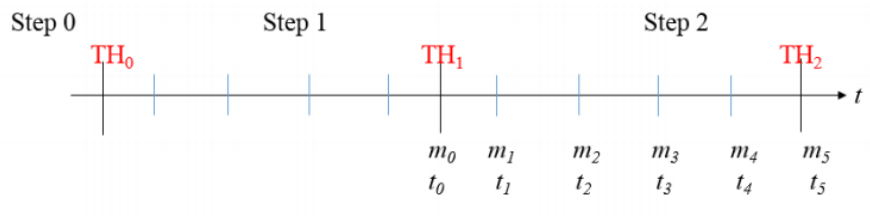


Figure 11.4. Illustration of first level TH coupling for TML.

With explicit coupling, the TH condition is updated at the end of each transport step is used for the next transport time step. For example, TH_1 calculated at the end of step 1 is used as the TH condition for the whole step 2, while TH_2 is used for further time steps. As a result, the two sets of cross sections are evaluated as:

$$\Sigma^n(TH_1, m_0, \Psi(TH_1, m_0, t_0)), \tag{11.56}$$

$$\Sigma^{n+1}(TH_1, m_5, \Psi(TH_1, m_5, t_5)), \tag{11.57}$$

where $\Psi(TH_1, m_0, t_0)$ is the sub-pin flux distribution evaluated at TH_1, m_0, and t_0. Since only one TH and neutronics calculation is performed in each time step, it is not possible to have two sets of flux distributions $\Psi(TH_1, m_0, t_0)$ and $\Psi(TH_1, m_5, t_5)$ simultaneously for each time step. The $\Psi(TH_1, m_5, t_5)$ is explicitly calculated after the transport solve in each transport step, and an approximation for $\Psi(TH_1, m_0, t_0)$ is thus required. The following three approximations are presented:

- Approximation 1 ($\overline{TH}_0 \approx TH_1$):

$$\Sigma^n(TH_1, m_0, \Psi(TH_1, m_0, t_0)) \approx \Sigma^n(\overline{TH}_0, m_0, \Psi(\overline{TH}_0, m_0, t_0)). \tag{11.58}$$

- Approximation 2 ($\Psi(\overline{TH}_0, m_0, t_0) \approx \Psi(TH_1, m_0, t_0))$) :

$$\Sigma^n(TH_1, m_0, \Psi(TH_1, m_0, t_0)) \approx \Sigma^n(TH_1, m_0, \Psi(\overline{TH}_0, m_0, t_0)). \tag{11.59}$$

- Approximation 3 ($\Psi(TH_1, \overline{m_5}, \overline{t_5}) \approx \Psi(TH_1, m_0, t_0)$) :

$$\Sigma^n(TH_1, m_0, \Psi(TH_1, m_0, t_0)) \approx \Sigma^n(TH_1, m_0, \Psi(TH_1, \overline{m_5}, \overline{t_5})). \tag{11.60}$$

The line above the parameters highlights the approximations.

A typical LWR single pin case Figure 11.5 is used to demonstrate the accuracy of these approximations. A super-prompt reactivity insertion with TH feedback pulse history is shown in Figure 11.6.

As shown in Figure 11.6, approximation 3 predicts the solution very well, while approximation 1 and approximation 2 overestimate and underestimate the pulse height, respectively. In approximation 1, the TH condition used is from

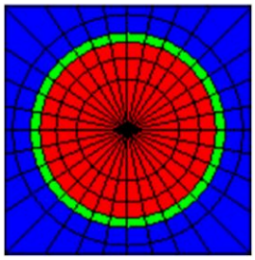


Figure 11.5. Single pin case geometry to demonstrate TH feedback.

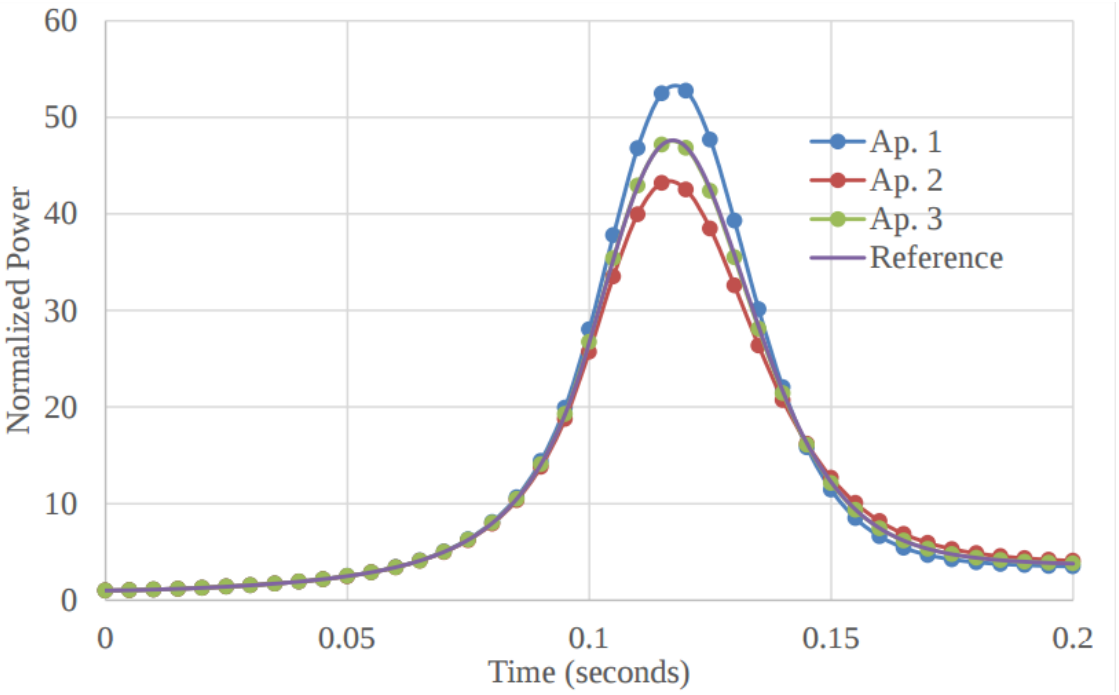


Figure 11.6. Power pulse with three approximations.

the previous step, and the Doppler feedback is delayed, resulting in a higher pulse. In approximation 2, the flux used to weight Σ^n is from the previous time step, and it underestimates the pulse height as well. This is because when the temperature increases, the absorption in the fuel increases due to the Doppler feedback effect, and the flux in the fuel region is reduced. If the weighting flux is from the previous time step, the flux decrement due to the increase of absorption is ignored, resulting in overestimation of the Doppler feedback and underestimating the pulse height.

Approximation 3 is a reasonable assumption since, initially, the flux distribution in Eq. (11.56) is used to homogenize the CMFD-based cross sections rather than to directly calculate the fission source, delayed source, etc. Therefore, a reasonable approximation to the flux is acceptable since it is only a weighting function. Secondly, the TH sub-pin temperature distribution is global, while the material variation from m_0 to m_5 is local. This observation validates the advantages of approximation 3 over approximation 2. Finally, the sub-pin distribution varies slowly with respect to time once the TH conditions and materials are the same, meaning this should introduce only a small discrepancy for

the time advancement from t_0 to t_5.

11.2.5.1 Second Level TH Coupling for 3D CMFD/EPKE

Once the feedback is applied to the model, the CMFD matrix coefficients change simultaneously. As a result, the dynamic coefficients need to be re-evaluated with the new CMFD matrix coefficients.

11.2.5.2 Overall Flow Chart for TML with TH Feedback

The overall flow chart for TH feedback in the TML algorithm is shown in Figure 11.7, where the three vertical blocks present the three levels of TML. The left column of blocks represents the general transport transient iteration scheme with TH feedback, where the angular and subpin flux shapes are assumed to be accurate. The pin-wise amplitude function of the transport solution is corrected using intermediate time steps by performing CMFD steps; this is shown in the middle column of blocks. Similarly, the global shape function predicted by the CMFD steps are assumed to be accurate, and the whole core amplitude is corrected by the fine EPKE steps illustrated in the right column of blocks.

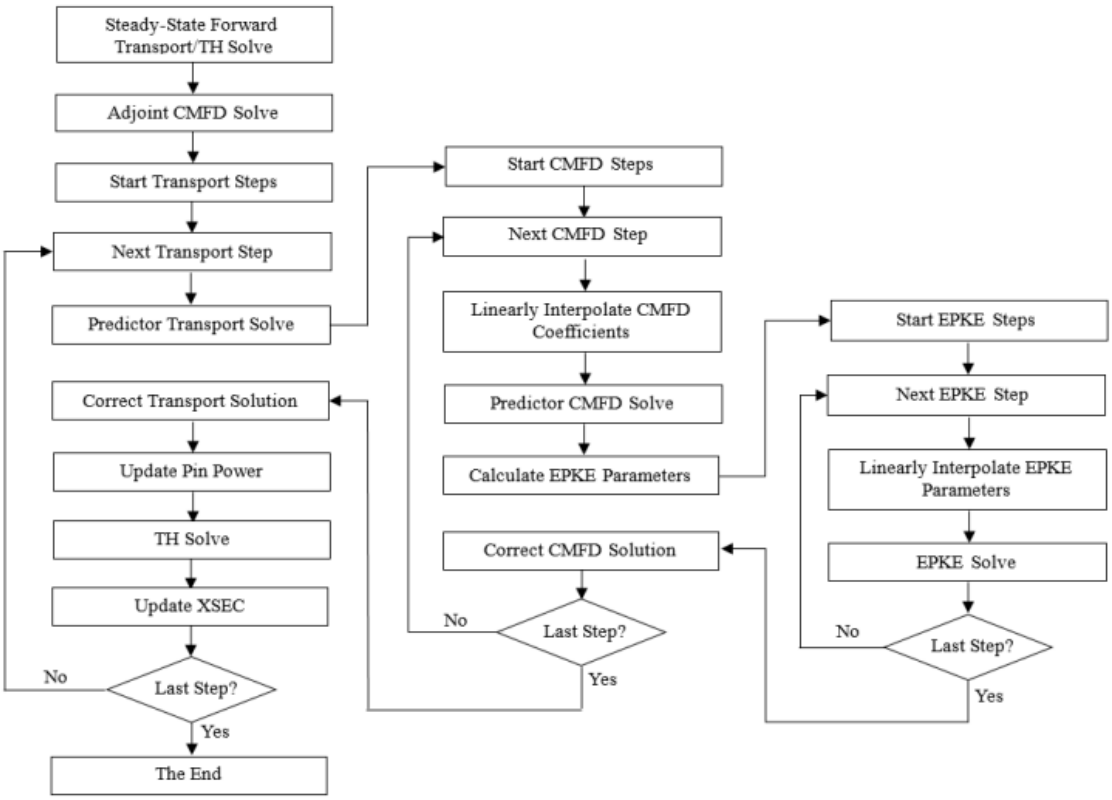


Figure 11.7. Flow chart for TML with TH feedback.

12. Simplified Thermal Hydraulic Model

12.1 Introduction

The simplified TH solver in MPACT provides a basic mechanism to apply TH feedback to the reactor. It applies this feedback in two steps. The first step is to solve for the flow distribution through the reactor to obtain the coolant temperature and density. When the coolant conditions are obtained, the second step is to solve for the fuel temperature for each pin in the model. This process is outlined below.

12.2 Fluid Flow Model

The flow model in MPACT provides the basic mass-energy balance of the fluid flow through the reactor. The first approximation involves the use of closed flow channels. A flow channel can be either a full assembly, a quarter assembly, or the region between four fuel pins, as illustrated by Figure 12.1. The flow through the core, $\dot{m}_{core}$, is given as user input. The flow through each channel is determined by weighting each channel by its inlet area, A_{chan}, as shown in

$$\dot{m}_{chan} = \left(\frac{A_{chan}}{\sum_{c \in channels} A_c} \right) \dot{m}_{core}, \tag{12.1}$$

where A_{chan} is the flow area of a given channel, and the denominator is the total flow area of all channels in the core.

Once the mass flow rate at the inlet of each channel is determined, the solver sequentially progresses up the channel and determines the enthalpy at the outlet of each node in the neutronic solution using

$$h_{out} = h_{in} + \frac{P_n}{\dot{m}_{chan}}, \tag{12.2}$$

where h_{in} and h_{out} are the flow enthalpy at the inlet and outlet of a node, respectively, and P_n is the power generated by all fuel contained in a a given node. The outlet enthalpy becomes the inlet enthalpy for the next node. This marching procedure represents an enthalpy balance for each neutronic node, but the main pieces of information needed are the average quantities within the node. A linear assumption is made to determine the average enthalpy inside the node of interest:

$$\bar{h} = \frac{h_{out} + h_{in}}{2}. \tag{12.3}$$

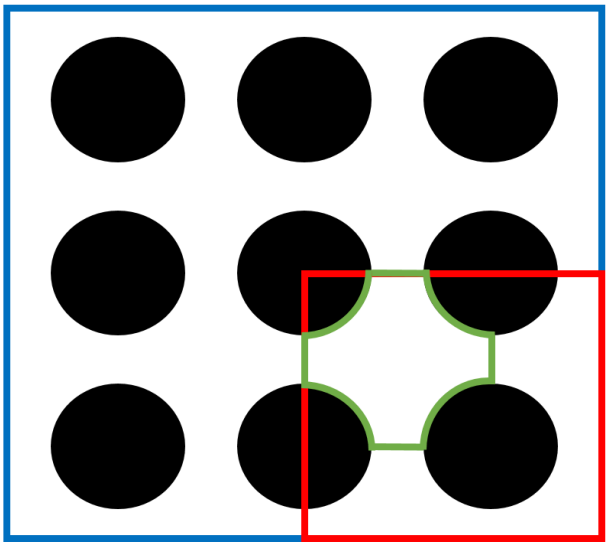


Figure 12.1. Representative flow channels for a 3×3 assembly: full assembly (blue), quarter assembly (red), or closed channel (green).

Once the average enthalpy is determined, equations of state are used to determine the coolant temperature and density:

$$\overline{T_{cool}} = T\left(\overline{h}, P_{sys}\right), \tag{12.4a}$$

$$\overline{\rho_{cool}} = \rho\left(\overline{h}, P_{sys}\right), \tag{12.4b}$$

where P_{sys} is the system pressure.

The equations of state for the simplified TH model are obtained from tables in the sub-channel TH code CTF [75], where the property of interest can be looked up as a function of enthalpy or temperature. The pressure is assumed to be 2250 psia, the nominal operating pressure for most PWRs. The temperature ranges from 279 K to 647 K, which encompasses the operating range of a standard PWR.

Once the coolant properties are determined for each channel, the fuel temperature model (described below) is applied to each pin in the core.

12.3 Fuel Temperature Models

To obtain fuel and clad temperatures, two different methods can be used. The first is a 1D heat conduction solver which can be applied to each pin in the model. The second method uses fuel temperature tables to obtain average fuel and clad temperatures.

12.3.1 1D Heat Conduction

The first method solves the 1D heat conduction equation in every pin in the model. The first step is to determine the cladding surface temperature. Once the clad surface temperature is determined, heat is conducted through the cladding, across the gap through the use of a gap conductivity, and finally through the fuel itself.

12.3.1.1 Heat Transfer from Fluid to Clad

Obtaining the clad surface temperature is achieved through the use of the Dittus-Boelter heat transfer coefficient, which is modified to account for a regular square array of pins [87]:

$$T_{clad,surf} = T_{mod} + \frac{P_l}{A_{surf} h_{mDB}}, \tag{12.5a}$$

$$h_{mDB} = C_0 \frac{k}{D_h} \mathrm{Re}^{0.4} \mathrm{Pr}^{0.2}, \tag{12.5b}$$

$$C_0 = 0.042 \frac{p_{pin}}{2 r_{clad}} - 0.024, \tag{12.5c}$$

where P_l is the local pin power, A_{surf} is the surface area of the clad, h_{mDB} is the modified Dittus-Boelter heat transfer coefficient, k is the fluid thermal conductivity, D_h is the hydraulic diameter of the channel, Re is the Reynolds number of the fluid, Pr is the Prantl number of the fluid, p_{pin} is the pin pitch, and r_{clad} is the clad outer radius. When a pin touches multiple flow channels, h_{mDB} is calculated for each of the channels and then harmonically averaged to obtain a single heat transfer coefficient for the pin. Several properties of the fluid are needed for this calculation, and they are calculated using the same tables discussed in the previous section.

12.3.1.2 Gap Conductance

A region requiring special treatment is the fuel-clad gap, a small region filled with an inert gas (typically He), providing a relatively large thermal resistance. Many complex physical phenomona influence the gap thickness, fuel-clad contact, and the overall thermal resistance of the gap. Consequently, the dynamics of the fuel-clad gap are difficult to model exactly, so instead of modeling the numerous complex physical phenomena through this region, the solver instead relies on an effective heat transfer coefficient for the gap. This value may be set by the user, or it may rely on some semi-emperical correlation or tabulated evaluation. The equation relating the fuel pellet surface temperature to the clad inner surface temperature is given by:

$$T_{fuel,surf} = T_{clad,in} + \frac{P_l}{A_{surf} h_{gap}}, \tag{12.6}$$

where $T_{fuel,surf}$ is the fuel surface temperature, $T_{clad,in}$ is the inner clad temperature, P_l is the local power, A_{surf} is the surface area of heat transfer, and h_{gap} is the gap conductance.

12.3.1.3 Radial Heat Transfer Equation

The flow of heat through a medium is well understood and is modeled using the heat conduction equation,

$$-\boldsymbol{\nabla}\cdot k\left(T\right)\boldsymbol{\nabla}T\left(\boldsymbol{x}\right)=\dot{q}\left(\boldsymbol{x}\right), \tag{12.7}$$

where k is the thermal conductivity, T is the temperature, and $\dot{q}$ is the heat generation rate. This is simplified into a 1D cylindrical equation equation for the solution here,

$$-\frac{1}{r}\frac{d}{dr}rk\left(T\right)\frac{d}{dr}T\left(r\right)=\dot{q}\left(r\right). \tag{12.8}$$

To discretize the 1D heat conduction equation, finite volumes are defined between a ring bounded by r_i and r_{i+1}. Inside these volumes, the heat generation rate and the thermal conductivity are considered constant. Using these two approximations, an analytic expression can be derived for the temperature inside a region. The temperature at r_{i+1} and the heat flux at r_i are used as the boundary conditions for each ring:

$$T\left(r\right)=-\frac{\dot{q}}{4k}r^2+C_0\ln\left(r\right)+C_1, \tag{12.9a}$$

$$T\left(r_{i+1}\right)=T_{out}, \tag{12.9b}$$

$$q_i'\left(r_i\right)=-k_i\frac{dT}{dr}. \tag{12.9c}$$

These conditions result in the following equation for the temperature inside a volume:

$$T\left(r\right)=\frac{\dot{q}_i}{4k_i}\left(r_{i+1}^2-r^2\right)+\left(\frac{\dot{q}_i r_i^2}{2k_i}-\frac{q_i' r_i}{k_i}\right)\ln\left(\frac{r}{r_{i+1}}\right)+T\left(r_{i+1}\right),$$
$$r_i<r<r_{i+1}, \tag{12.10}$$

where k_i, $\dot{q}_i$, and q_i' are the thermal conductivity of the fuel, volumetric heat rate, and linear heat rate in region i, respectively. Using Eq. (12.10), the temperature on the inside of the volume, r_i, can be determined. The volume-averaged temperature of the volume:

$$\overline{T_i}=\frac{1}{\pi\left(r_{i+1}^2-r_i^2\right)}\int_{r_i}^{r_{i+1}}2\pi rT\left(r\right)dr. \tag{12.11}$$

With the equation relating the temperature at r_i and the average temperature $\overline{T_i}$ to the temperature at r_{i+1}, an iterative scheme can be devised starting at the outside of the clad and moving inward. The equation for the cladding is simplified because there is no heat generation:

$$T\left(r\right)=\frac{q_i' r_i}{k_i}\ln\left(\frac{r_{i+1}}{r}\right)+T\left(r_i\right),\quad r_i<r<r_{i+1}. \tag{12.12}$$

Since the thermal conductivity is a function of temperature, each volume is iterated until the average temperature converges. Once the temperature in the clad is obtained, the gap conductance model is used to obtain the fuel surface temperature. Then the same procedure is used to solve for the average temperature in each ring of the fuel. Since the

fuel generates heat, the linear heat rate at r_i changes at each ring. The iteration procedure continues into the the fuel until the innermost region is reached. The major difference about the center core of fuel is that $r_i = 0$ and $q'(0) = 0$ due to the symmetry of the 1D equations.

12.3.1.4 Thermal Properties

The thermal conductivity of the clad and fuel are modeled in units of $\frac{W}{m \cdot K}$ using semi-emprical correlations based on low order polynomials. The thermal conductivity of the clad is modeled using:

$$k_{clad}(T) = 7.51 + 2.09 \cdot 10^{-2} T - 1.45 \cdot 10^{-5} T^2 + 7.67 \cdot 10^{-9} T^3 \,. \tag{12.13}$$

This correlation is recommended in [31], which provides a detailed review of the experimental data which justifies this correlation.

The thermal conductivity of the fuel does not take into account: burnup, porosity, initial plutonium loading, or gadolinium loading, but instead it is only a function of temperature:

$$k_{fuel}(T) = 1.05 + \frac{2150}{T - 73.15} \,. \tag{12.14}$$

12.3.2 Fuel Temperature Tables

In addition to 1D conduction calculations, MPACT can also use fuel temperature tables. These tables return average clad and fuel temperatures as follows:

$$\overline{T_{fuel}} = T_{bulk} + aP_l + bP_l^2, \tag{12.15a}$$

$$\overline{T_{clad}} = f_{clad}\overline{T_{fuel}} + (1 - f_{clad})\overline{T_{fuel}}, \tag{12.15b}$$

where T_{bulk} is the average moderator temperature in the neighboring channel, P_l is the linear power rate in the pin, f_{clad} is a constant value set to 0.2, and a and b are exposure-dependent constants found in the temperature table. The table constants are generated externally by a fuel performance code.

12.4 Discussion

The simplified TH solver described above is useful for predicting leading order effects of TH feedback. The 1D conduction solver can provide radial temperature distributions within each pin, and the temperature tables facilitate depletion calculations by incorporating the fuel temperature's dependence on exposure. However, these capabilities are limited in accuracy and flexibility and should be used as a preliminary scoping capability. It is recommended

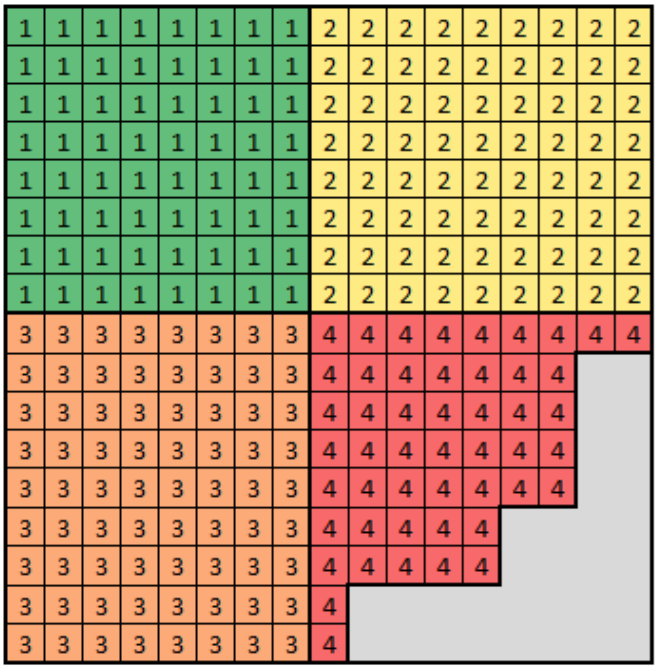
that more advanced TH and fuel performance capabilities such as the subchannel TH code CTF [75] and the fuel performance code BISON [42, 93, 37] be used for production calculations.

13. Miscellaneous Topics

This chapter addresses topics that do not fit into any of the previous chapters, such as the module-based decomposition scheme and rotational symmetry. Both of these topics required modifications to the various solvers (MOC, CMFD, axial transport), so they are best discussed separately.

13.1 Module-Based Decomposition Strategy

One improvement made to the data passing schemes in MPACT was the incorporation of a module-based decomposition scheme, which easily allows for more flexible and load-balanced partitions. In most production-level cases, a quarter assembly is considered to be a module, which is the smallest repeatable geometry component. As such, the partitions are formed on the module basis. Figure 13.1 shows an example 2D quarter-core slice with 257 modules distributed among 4 partitions. With the old partitioning restrictions, which effectively projected a cartesian grid onto the problem, all domains could only have one neighbor partition in each direction (N/S/E/W) and could not have mixed boundary conditions. The data passing for the transport solvers was performed across the entire domain interface at one time. For example, all 9 modules along the east boundary of partition 3 would be used to construct a single buffer of data to be transmitted to partition 4.

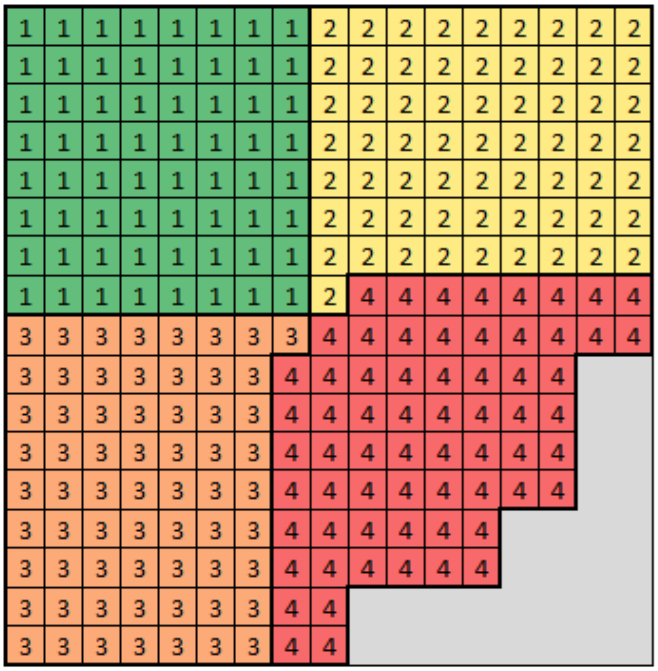


Partition	# Modules
1	64
2	72
3	72
4	49

Figure 13.1. Quarter-core layout with four partitions using old scheme.

There are at least two ways this could have been modified to allow for more flexible partitions. One method would be to preserve this behavior so that all modules along a parallel boundary are used to build a buffer. However, there would be an arbitrary number of parallel boundaries for each partition. Another option, which was pursued in MPACT because of its relative simplicity, is to allow each module to be responsible for its own data communication. This means that each module constructs a buffer and sends it to its parallel neighbor and receives one back, as well. The initial concern with this was that the parallel efficiency would suffer since many more smaller messages would be sent. In practice, however, the efficiency was not observed to suffer, so this approach has been adopted throughout MPACT.

With each module handling its data communication, fairly arbitrary partitions can be obtained. Figure 13.2 shows the comparable 4 domain partition, which allows for nearly perfect load-balancing, as all domains have 64–65 modules. The only primary restriction at this point is that each domain must be convex. However, this restriction is imposed because of MOC ray tracing limitations, not parallel communication itself.



Partition	# Modules
1	64
2	64
3	64
4	65

Figure 13.2. Quarter-core layout with four partitions using new scheme.

13.2 Rotational Symmetry Boundary Treatment

A number of boundary conditions are available in MPACT, but the two that are most commonly used are *mirror* and *rotational symmetry*. This section clarifies rotational symmetry and briefly discusses the parallel communication modifications that were necessary to enable it in MPACT.

Figure 13.3 shows the core layout and control rod bank assignments for Watts Bar Unit 1 [40]. If unfolded, the core layout and many of the control rod banks (A-D and SA-SB) are mirror symmetric. However, it can be seen that SC and SD are not mirror-images, but are rotationally symmetric.

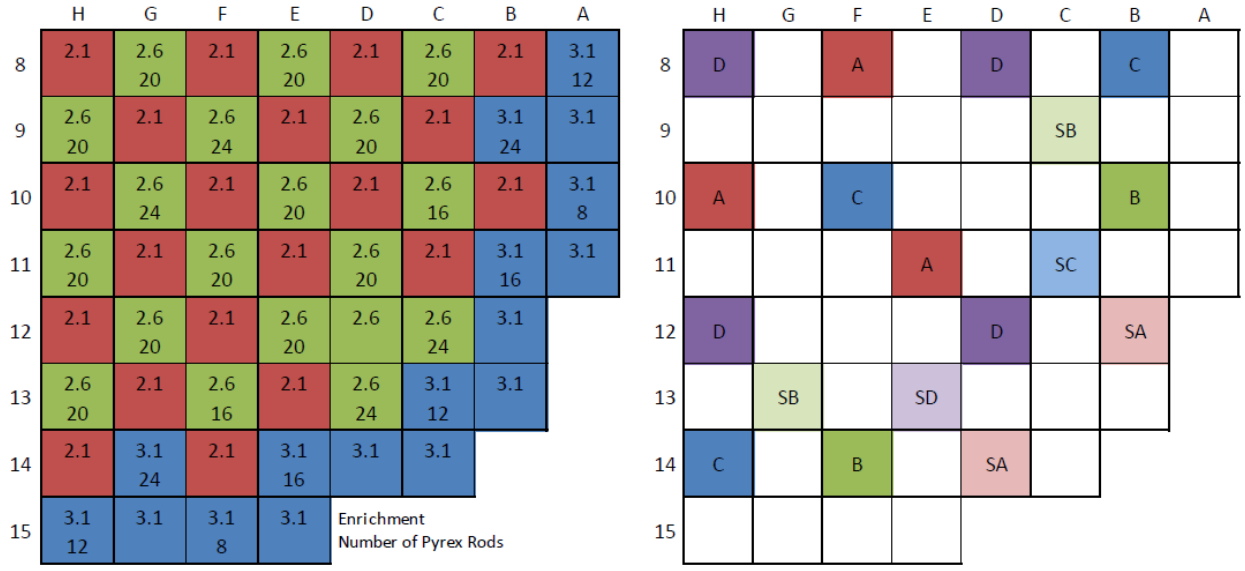


Figure 13.3. Watts Bar Unit 1 - core (left) and rod bank (right) layouts.

To allow for these to be appropriately modelled, a number of modifications were necessary to both the CMFD and MOC solvers, particularly regarding the parallel communication. Along the north and west boundaries, which are considered to be the quarter symmetry boundaries, special accommodations were required. With mirror symmetry, angular fluxes in MOC on these boundaries are simply reflected to a corresponding angle. However, with rotational symmetry, the angular fluxes must be communicated to the corresponding module and angle on the opposing face. For example, a ray coming into contact with the northern boundary of assembly D-8 must be transmitted to the western boundary of H-12, and the angle must be adjusted to reflect this transmission. Similar modifications were necessary to the CMFD solver, which must now construct coupling coefficients using the flux data from its rotational partner instead of using an albedo boundary condition as is done with mirror symmetry.

Rotational symmetry has become the default quarter symmetry boundary condition since this is most representative of operating reactors. When eighth core symmetry exists, rotational and mirror symmetry are equivalent.

Bibliography

[1] J. R. Askew. A characteristics formulation of the neutron transport equation in complicated geometries. *UKAEA Report*, AEEW-M-1108, 1971.

[2] J. R. Askew, F. J. Fayers, and P. B. Kemshell. A general description of the lattice code WIMS. *J. British Nucl. Energy Soc.*, 5:564–585, 1966.

[3] R. S. Baker and R. E. Albouffe. Parallel 3-D S_N performance for DANTSYS/MPI on the Cray T3D. In *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing for Nuclear Applications*, volume 1, page 377, Saratoga Springs, New York, October 5–9 1997.

[4] R. S. Baker, C. Asano, and D. N. Shirley. Implementation of the first-order form of the three-dimensional discrete ordinates equations on a T3D. *Trans. Am. Nucl. Soc.*, 73:170, 1995.

[5] R. S. Baker and K. R. Koch. An S_N algorithm for the massively parallel CM-200 computer. *Nucl. Sci. Eng.*, 128:312, 1998.

[6] S. Balay et al. Petsc User Manual. Technical Report ANL-95/11 - Revision 3.5, Argonne National Laboratory, http://www.mcs.anl.gov/petsc, 2014.

[7] S. J. Ball and R. K. Adams. MATEXP, a general purpose computer program for solving ordinary differential equations by the matrix exponential method. Technical Report ORNL-TM-1933, Oak Ridge National Laboratory, 1967.

[8] I. I. Bondarenko et al. *Group Constants for Nuclear Reactor Calculations*. Consultants Bureau, New York, 1964.

[9] S. M. Bowman. SCALE 6: Comprehensive nuclear safety analysis code system. *Nucl. Technol.*, 174:126–148, 2011.

[10] J. Y. Cho and H. G. Joo. Solution of the C5G7MOX benchmark three-dimensional extension problems by the DeCART direct whole core calculation code. *Prog. Nucl. Energy*, 48:456–466, 2006.

[11] J. Y. Cho, H. G. Joo, K. S. Kim, and S. Q. Zee. Three-dimensional heterogeneous whole core transport calculation employing planar MOC solutions. *Trans. Am. Nucl. Soc.*, 87:234–236, 2002.

[12] J. Y. Cho, K. S. Kim, C. C. Lee, S. Q. Zee, and H. G. Joo. Axial SPN and radial MOC coupled whole core transport calculation. *J. Nuc. Sci. Tech.*, 44:1156–1171, 2007.

[13] N. Z. Cho. Fundamentals and recent developments of reactor physics methods. *Nucl. Eng. Tech.*, 37:25–78, 2005.

[14] N. Z. Cho and S. G. Hong. Neutron transport theory; computational algorithms and applications. In *Chungmoon-Gak, Seoul, Korea*, 2000.

[15] N. Z. Cho, G. S. Lee, and C. J. Park. Fusion method of characteristics and nodal method for 3D whole core transport calculation. *Trans. Am. Nucl. Soc.*, 86:322–324, 2002.

[16] N. Z. Cho, G. S. Lee, and C. J. Park. A fusion technique of 2-D/1-D methods for three-dimensional whole-core transport calculations. In *Proc. Korean Nucl. Soc., Kwangju, Korea*, May 2002.

[17] N. Z. Cho, G. S. Lee, and C. J. Park. Refinement of the 2-D/1-D fusion method for 3-D whole core transport calculation. *Trans. Am. Nucl. Soc.*, 87:417–420, 2002.

[18] N. Z. Cho, G. S. Lee, and C. J. Park. Partial current-based CMFD acceleration of the 2D/1D fusion method for 3D whole core transport calculations. *Trans. Am. Nucl. Soc.*, 88:594–596, 2003.

[19] D. E. Cullen. Application of the probability table method to multigroup calculations of neutron transport. *Nucl. Sci. Eng.*, 55:387, 1974.

[20] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997.

[21] T. Downar, B. Kochunas, Y. Liu, B. Collins, and S. Stimpson. MPACT verification and validation manual (rev 4). Technical report, CASL-U-2018-1641-000, 2018.

[22] T. J. Downar et al. PARCS: Purdue advanced reactor core simulator. In *Proc. PHYSOR 2002, Seoul, South Korea*, 2002.

[23] S. Dulla, E. Mund, and P. Ravetto. The quasi-static method revisited. *Prog. Nucl. Energy*, 50:908, 2008.

[24] A. Santamarina et. al. The JEFF-3.1.1 nuclear data library. validation results from JEF-2.2 to JEFF-3.1.1. Technical report, OECD/NEA Data Bank, 2009.

[25] A. Zhu et al. The implementation and analysis of the MOC and CMFD adjoint capabilities in the 2D-1D code MPACT. In *Proc. M&C 2015, Nashville, TN*, April 19-23 2015.

[26] A. Zhu et al. Transient methods for pin-resolved whole core transport using the 2D-1D methodology in MPACT. In *Proc. M&C 2015, Nashville, TN*, April 19-23 2015.

[27] A. Zhu et al. A multi-level quasi-static kinetics method for pin-resolved transport transient reactor analysis. *Nuclear Science and Engineering*, 182(4), 2016.

[28] A. Zhu et al. Stability analysis of the backward euler time discretization for the pin-resolved transport transient reactor calculation. *Annals of Nuclear Energy*, 87(2):252–266, 2016.

[29] B. Collins et al. Demonstration of full core reactor depletion with MPACT. Technical Report CASL-U-2014-0140-000, CASL, 2014.

[30] C. Shim et al. Application of backward differentiation formula to spatial reactor kinetics calculation with adaptive time step control. *Nuclear Engineering Technology*, 43(6), 2011.

[31] D. T. Hagrman et al. SCDAP/RELAP5/MOD3.1 Code Manual Volume IV: MATPRO – a library of materials properties for light-water-reactor accident analysis. Technical Report NUREG/CR-6150, U.S. Nuclear Regulatory Commission, 1993.

[32] J. Cho et al. Transient capability for a MOC-based whole core transport code DeCART. *Trans. Am. Nucl. Soc.*, 92:721, 2005.

[33] M. Hursin et al. The development and implementation of a one-dimensional SN method in the 2D-1D integral transport solution. *Nuclear Science and Engineering*, 176:186–200, 2014.

[34] R. J. J. Stammler et. al. HELIOS methods. Technical report, Studsvik Scandpower, 2003.

[35] S. Dulla et al. Accuracy of a predictor-corrector quasi-static method for space-time reactor dynamics. In *Proc. PHYSOR 2006, Vancouver, BC*, Sept 10-14 2006.

[36] H. Finnemann, F. Bennewitz, and M. Wagner. Interface nodal current technique for multi-dimensional reactor calculation. *Atomkernenergie*, 30:123, 1977.

[37] D. Gaston et al. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering Design*, 239:1768–1778, 2009.

[38] I. C. Gauld, G. Radulescu, G. Ilas, B. D. Murphy, M. L. Williams, and D. Wiarda. Isotopic depletion and decay methods and analysis capabilities in SCALE. *Nuclear Technology*, 174:169, 2011.

[39] J. Gehin. *A Quasi-Static Polynomial Nodal Method for Reactor Analysis*. PhD thesis, Massachusetts Institute of Technology, 1992.

[40] A. Godfrey. VERA core physics benchmark progression problem specifications. Technical report, CASL-U-2012-0131-004, 2014.

[41] R. Goldstein and E. R. Cohen. Theory of resonance absorption of neutrons. *Nucl. Sci. Eng.*, 13:132–140, 1962.

[42] J. D. Hales et al. BISON theory manual: The equations behind nuclear fuel analysis. Technical report, Idaho National Laboratory, 2015.

[43] M. J. Halsall. CACTUS, a characteristics solution to the neutron transport equations in complicated geometries. *UKAEA Report*, AEEW-R-1291, 1980.

[44] A. Hebert. Applied reactor physics. 2009.

[45] A. Hoffman. *A time-dependent method of characteristics formulation with time derivative propagation*. PhD thesis, University of Michigan, 2013.

[46] S. G. Hong and K. S. Kim. Iterative resonance treatment methods using resonance integral table in heterogeneous transport lattice calculations. *Ann. Nucl. Energy*, 38:32–43, 2011.

[47] M. Hursin. *Full Core, Heterogeneous, Time Dependent Neutron Transport Calculations with the 3D Code DeCART*. PhD thesis, University of California at Berkeley, 2010.

[48] H. G. Joo, J. Y. Cho, K. S. Kim, C. C. Lee, and S. Q. Zee. Methods and performance of a three-dimensional whole core transport code DeCART. In *Proc. PHYSOR 2004, Chicago, IL*, April 25-29, 2004.

[49] Y. S. Jung and H. G. Joo. Direct whole core calculation with thermal feedback using planar moc generated cross sections functions. In *Proc. M&C2011, Rio de Janeiro, Brazil*, May 8-12, 2011.

[50] B. W. Kelley, B. Collins, and E. W. Larsen. 2D/1D approximations to the 3D neutron transport equation. II: Numerical comparisons. In *Proc. M&C 2013, Sun Valley, ID, USA*, May 5-9, 2013.

[51] B. W. Kelley and E. W. Larsen. 2D/1D approximations to the 3D neutron transport equation. I: Theory. In *Proc. M&C 2013, Sun Valley, ID, USA*, May 5-9, 2013.

[52] B. W. Kelly and E. W. Larsen. A consistent 2D/1D approximation to the 3d neutron transport equation. *Nucl. Eng. Design*, 295:598–614, 2015.

[53] K. S. Kim, K. T. Clarno, Y. Liu, X. Wang, and W. R. Martin. Neutron capture energies for flux normalization and approximate model for gamma-smeared power. Technical report, CASL-U-2017-1377-000, Oak Ridge National Laboratory, 2017.

[54] K. S. Kim et al. Development of a multi-group neutron cross section library generation system for pwr. Technical report, KAERI/TR-3634/2007, Korea At. Energy Res. Inst., 2008.

[55] K. S. Kim et al. Development of a new 47-group library for the CASL neutronics simulators. In *Proc. M&C 2015, Nashville, Tennessee*, April 19-23, 2015.

[56] K. S. Kim, C. A. Gentry, A. T. Godfrey, Y. Liu, and S. Palmtag. Development of the multigroup cross section library for the casl neutronics simulator mpact: Verification. *Annals of Nuclear Energy*, 132:1–23, 2019.

[57] K. S. Kim, M. L. Williams, D. Wiarda, and K. T. Clarno. Development of the multigroup cross section library for the casl neutronics simulator mpact: Method and procedure. *Annals of Nuclear Energy*, 133:46–58, 2019.

[58] K. R. Kock, R. S. Baker, and R. E. Alcouffe. A parallel algorithm for 3D S_N transport sweeps. Technical Report LA-CP-92-406, Los Alamos national Laboratory, 1992.

[59] S. Kosaka and T. Takeda. Verification of 3D heterogeneous core transport calculation using non-linear iteration technique. *J. Nucl. Sci. Tech.*, 41:645–654, 2004.

[60] L. Lapidus and R. Luus. *Optimal Control of Engineering Processes*. Blaisdell Publishing Company, Waltham, Massachusetts, 1967.

[61] E. W. Larsen and B. W. Kelly. The relationship between the coarse-mesh finite difference and the coarse-mesh diffusion synthetic acceleration methods. *Nucl. Sci. Eng.*, 178:1–15, 2014.

[62] D. Lee, T. J. Downar, and Y. Kim. Convergence analysis of the nonlinear coarse mesh finite difference method for one-dimensional fixed source neutron diffusion problem. *Nucl. Sci. Eng.*, 147:127–147, 2004.

[63] G. S. Lee and N. Z. Cho. 2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX. *Prog. Nucl. Energy*, 48:410–423, 2006.

[64] F. Leszczynski. Neutron resonance treatment with details in space and energy for pin cells and rod clusters. *Ann. Nucl. Energy*, 14:589–601, 1987.

[65] Y. Liu et al. Modeling resonance interference by 0-D slowing-down solution with embedded self-shielding method. In *Proc. M&C 2013, Sun Valley, ID, USA*, May 5-9, 2013.

[66] Y. Liu et al. Development of explicit heat calculation and coupling between MPACT and CTF. Technical report, CASL-U-2019-1807-000, 2019.

[67] Y. Liu., B. Kochunas, W. Martin, and T. Downar. Delayed fission energy effect on LWR normal operation and transients. *Annals of Nuclear Energy*, 128:84–93, 2019.

[68] Y. Liu, W. R. Martin, M. L. Williams, and K. S. Kim. A full core resonance self-shielding method using a continuous energy quasi-1D slowing-down solution that accounts for temperature-dependent fuel subregions and resonance interference. *Nucl. Sci. Eng.*, 180:247–272, 2015.

[69] Y. Liu, S. Stimpson, K. S. Kim, B. Collins, and B. Kochunas. Runtime improvements to the cross section calculation in MPACT. Technical report, CASL-X-2016-1105-000, Oak Ridge National Laboratory, 2016.

[70] R. G. McClarren. Theoretical aspects of the simplified P_n equations. *Transport Theory and Statistical Physics*, 39:73–109, 2011.

[71] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2006.

[72] K. Ott and R. Neuhold. *Introductory nuclear reactor dynamics*. American Nuclear Society, La Grange Park, IL, 1985.

[73] A. K. Prinja and E. W. Larsen. General principles of neutron transport. 1, 2010.

[74] M. Ryu, Y. S. Jung, H. H. Cho, and H. G. Joo. Solution of the BEAVRS benchmark using the ntracer direct whole core calculation code. *Journal of Nuclear Science and Technology*, 52(7-8):961–969, 2015.

[75] R. K. Salko and M. N. Avramova. CTF Theory Manual. Technical report, The Pennsylvania State University.

[76] SCALE. A modular code system for performing standardized computer analyses for licensing evaluation. Technical report, ORNL-TM/2005/39, Version 6.1, Vols. I-III, Oak Ridge National Laboratory, Oak Ridge, Tenn., 2011.

[77] S. Shaner, B. Forget, and K. Smith. Sensitivity analysis and performance of the adiabatic, theta, and multigrid amplitude function kinetics methods in 2D MOC neutron transport. In *Proc. M&C 2013, Sun Valley, ID*, May 5–9 2013.

[78] K. S. Smith. Nodal method storage reduction by nonlinear iteration. *Trans. Am. Nucl. Soc.*, 44:265, 1983.

[79] R. J. J. Stammler and M. J. Abbate. *Methods of Steady-state Reactor Physics in Nuclear Design*. Academic Press, London, 1983.

[80] S. Stimpson, B. Collins, and B. Kochunas. Improvement of transport-corrected scattering stability and performance using a Jacobi inscatter algorithm for 2D-MOC. *Annals of Nuclear Energy*, 2017 [submitted].

[81] S. Stimpson, Y. Liu, B. Collins, and K. Clarno. MPACT subgroup self shielding efficiency improvements. Technical report, CASL-U-2016-1063-001, Oak Ridge National Laboratory, 2016.

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

[83] S. G. Stimpson, B. S. Collins, B. M. Kochunas, and T. J. Downar. Boundary acceleration techniques for CMFD-Accelerated 2D-MOC. In *Proc. PHYSOR 2014*, Kyoto, Japan, September 28 - October 3 2014.

[84] S. G. Stimpson, B. S. Collins, A. Zhu, and Y. Xu. A hybrid P_3/SP_3 axial transport solver for the MPACT 2D/1D scheme. In *Proc. PHYSOR 2016, Sun Valley, Idaho*, May 1–5, 2016.

[85] M. Tabuchi, A. Yamamoto, T. Endo, and N. Sugimura. Convergence analysis of moc inner iterations with large negative self-scattering cross-section. *Journal of Nuclear Science and Technology*, 50:493–502, 2013.

[86] A. Talamo. Numerical solution of the time dependent neutron transport equation by the method of the characteristics. *Journal of Computational Physics*, 240:248–267, 2013.

[87] L. S. Tong and J. Weisman. *Thermal Analysis of Pressurized Water Reactors (2nd edition)*. American Nuclear Society, Le Grange Park, Illinois, 1979.

[88] T. J. Trahan and E. W. Larsen. An asymptotic homogenized neutron diffusion approximation. I. theory. In *Proc. PHYSOR 2012, Knoxville, TN*, April 15-20 2012.

[89] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.

[90] Eugene L. Wachspress. *Iterative Solution of Elliptic Systems*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1966.

[91] M. L. Williams and M. Asgari. Computation of continuous-energy neutron spectra with discrete ordinates transport theory. *Nucl. Sci. Eng.*, 121:173–201, 1974.

[92] M. L. Williams and K. S. Kim. The embedded self-shielding method. In *Proc. PHYSOR 2012, Knoxville, TN*, April 15–20, 2012.

[93] R. Williamson et al. Multidimensional multiphysics simulation of nuclear fuel behavior. *Nuclear Materials*, 423:149–163, 2012.

[94] A. Yamamoto, Y. Kitamura, and Y. Yamane. Simplified treatments of anisotropic scattering in LWR core calculators. *Journal of Nuclear Science and Technology*, 45(3):271 – 279, 2008.

[95] B. C. Yee, B. Kochunas, E. W. Larsen, and Y. Xu. Space-dependent wielandt shifts for multigroup diffusion eigenvalue problems. *Nuclear Science and Engineering*, 2017.

[96] B. C. Yee, E. W. Larsen, and B. Kochunas. An analytical derivation of transport-corrected p0 cross sections and diffusion coefficients. In *Proc. PHYSOR 2016, Sun Valley, Idaho*, May 1–5, 2016.

[97] J. I. Yoon and H. G. Joo. Two-level coarse mesh finite difference formulation with multigroup source expansion nodal kernels. *Journal of Nuclear Science and Technology*, 45:668–682, 2008.

[98] M. T. H. Young, B. S. Collins, and W. R. Martin. Corrected diamond difference method for coupling from the method of characteristics to discrete ordinates. In *Proc. PHYSOR 2014, Kyoto, Japan*, September 28–October 3, 2014.

[99] A. Zhu et al. An optimally diffusive coarse mesh finite diference method to accelerate neutron transport calculations. *Annals of Nuclear Energy*, 95:116–124, 2016.