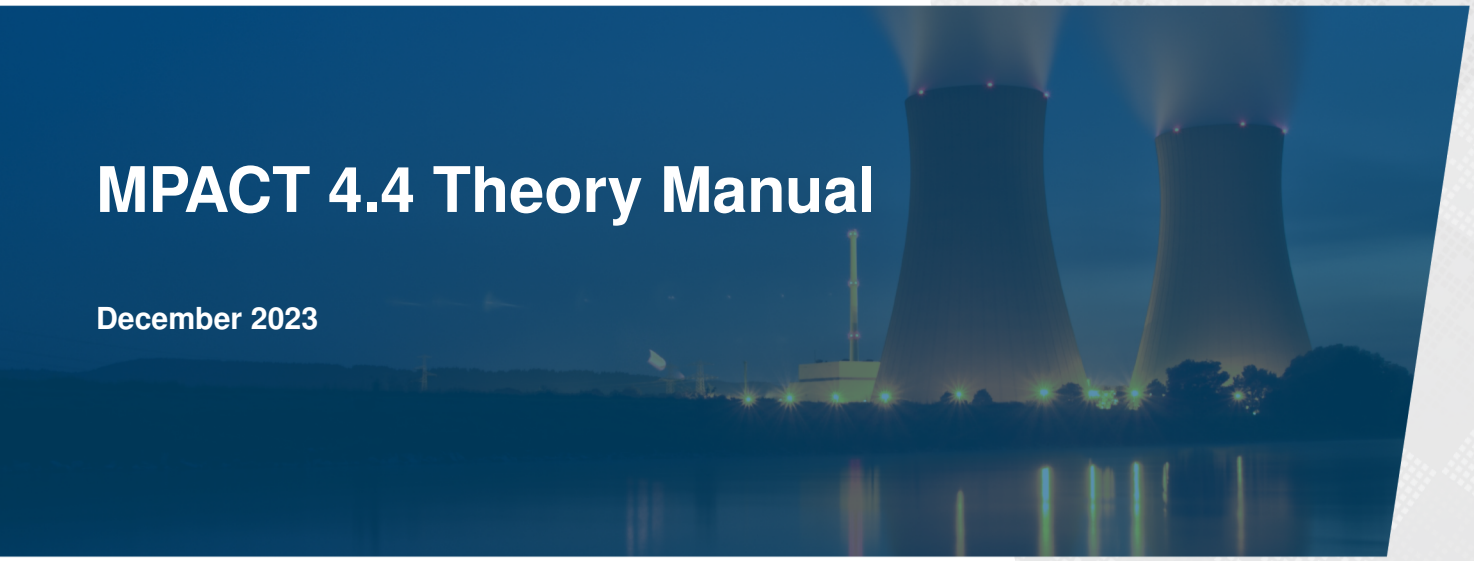
VERA Virtual Environment for Reactor Applications

ORNL/SPR-2023/3090



MPACT 4.4 Theory Manual

December 2023

Editors: Aaron Graham[1], Edward Larsen[2], Benjamin Collins[3], Brendan Kochunas[2], and Shane Stimpson[1]

[1]Oak Ridge National Laboratory
[2]University of Michigan
[3]Veracity Nuclear, LLC.

OAK RIDGE National Laboratory

U.S. DEPARTMENT OF ENERGY

ORNL is managed by UT-Battelle LLC for the US Department of Energy.

ORNL/SPR-2023/3090

MPACT 4.4 THEORY MANUAL

Editors: Aaron Graham[1], Edward Larsen[2], Benjamin Collins[3], Brendan Kochunas[2], and Shane Stimpson[1]

[1]Oak Ridge National Laboratory
[2]University of Michigan
[3]Veracity Nuclear, LLC.

Date Published: December 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

MPACT 4.4 Theory Manual

Revision Log

Revision	Date	Affected Pages	Revision Description
0	9/30/2023	51–52	Minor updates to acronyms and nomenclature
		74–77	New section on on-the-fly energy condensation;
		89	New section on Wigner-Seitz approximation;
		99–102	New section on xenon and samarium treatments;
		131–143	Minor revisions to section 12.2.1, added sections on User TH, bypass flow calculation, natural circulation, and decay heat coupling;
		152–156	added missing nodal data variables;
		176	added section on critical searches;
		177–end	added new references;
			this document supercedes ORNL/SPR-2021/2330

Document pages that are:

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

MPACT 4.4 Theory Manual

Approvals:

Aaron Graham, Product Software Manager	**Date**
Bob Salko, Independent Reviewer	**Date**

Contributors (in alphabetical order)

- Dr. Mehdi Asgari
- Dr. Kevin Clarno
- Dr. Benjamin Collins
- Dr. Zackary Dodson
- Prof. Thomas J. Downar
- Dr. Andrew Fitzgerald
- Dr. Jess Gehin
- Dr. Cole Gentry
- Dr. Tarek Ghaddar
- Andrew Godfrey
- Dr. Aaron Graham
- Shane Henderson
- Daniel Jabaay
- Dr. Jesse Jones
- Dr. Blake Kelley
- Dr. Michael Jarrett
- Dr. Kang Seog Kim
- Dr. Brendan Kochunas
- Dr. Joel Kulesza
- 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
- Dr. Qicang Shen
- Dr. Shane Stimpson
- Dr. Travis Trahan
- Dr. Erik Walker
- Dr. Jipu Wang
- Dr. Will Wieselquist
- 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.

The development of MPACT has also been supported by the US Department of Energy Nuclear Energy Advanced Modeling and Simulation (NEAMS) program.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

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. Development of the underlying theory of the 2D/1D method is presented, 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 complete and finalized. Future versions of this manual will address current deficiencies.

CHAPTER 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 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 [89,90], 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) [4], [68], 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 warranted and if 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, and 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 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 describes the multiphysics coupling in MPACT, including an internal simplified TH model, coupling with external codes, and energy deposition treatments.
- Chapter 13 describes MPACT's additional capability to generate nodal cross sections to drive nodal simulator calculations.
- Chapter 14 provides brief discussions of some miscellaneous topics:
  - Parallel Decomposition Mechanics
  - Boundary Condition Treatment
  - Rod Cusping Treatment

CHAPTER 2. 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 the work by Prinja and Larsen (2010) [123].

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, then 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:

$\Sigma_t(\boldsymbol{x}, E)ds$ = the incremental probability that a neutron at point $\boldsymbol{x}$, with energy E, traveling an incremental distance ds, will experience a collision with a nucleus, (2.2a)

$\Sigma_s(\boldsymbol{x}, E)ds$ = the incremental probability that a neutron at point $\boldsymbol{x}$, with energy E, traveling an incremental distance ds, will scatter off a nucleus, (2.2b)

$\Sigma_\gamma(\boldsymbol{x}, E)ds$ = the incremental probability that a neutron at point $\boldsymbol{x}$, with energy E, traveling an incremental distance ds, will be captured by a nucleus, (2.2c)

$\Sigma_f(\boldsymbol{x}, E)ds$ = the incremental probability that a neutron at point $\boldsymbol{x}$, with energy E, traveling an incremental distance ds, will initiate a fission event with a nucleus. (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

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

This cross section satisfies

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

Also, the *fission spectrum* is defined by

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

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, 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 $E + dE$ are considered. The *angular neutron density* $N(\boldsymbol{x}, \boldsymbol{\Omega}, E, t)$, a function of six independent variables, is defined by

$$\begin{aligned} N(\boldsymbol{x}, \boldsymbol{\Omega}, E) dV d\Omega dE = {} & \text{the incremental number of neutrons in} \\ & dV d\Omega dE \text{ about the phase space point } (\boldsymbol{x}, \boldsymbol{\Omega}, E) \; . \end{aligned} \tag{2.8}$$

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

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) = v N(\boldsymbol{x}, \boldsymbol{\Omega}, E) \; , \tag{2.9}$$

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

$$\begin{aligned} \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) dV d\Omega dE = {} & \text{the incremental rate at which path length is} \\ & \text{generated by neutrons in } dV d\Omega dE \text{ about } (\boldsymbol{x}, \boldsymbol{\Omega}, E) \; . \end{aligned} \tag{2.10}$$

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

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

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

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

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

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.) Furthermore, the solution, $\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E)$, of the eigenvalue problem in Eqs. (2.11a) and (b) is unique up to an arbitrary constant. Therefore, an additional equation is needed to define a normalization condition for $\psi(\boldsymbol{x}, \boldsymbol{\Omega}, E)$. Because MPACT is used to model reactors, this normalization condition is usually given in terms of total power:

$$P_{th} = \int_V \int_0^\infty \int_{4\pi} \kappa \Sigma_f(\boldsymbol{x}, E) \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) d\Omega dE d\boldsymbol{x} \ , \tag{2.11c}$$

where P_{th} is the thermal power of the reactor, and κ is a factor that represents the average energy release per fission. The normalization condition can be expressed in other ways, such as in terms of the total neutron population or total neutron production rate. However, the method through which the normalization is expressed is less important its existence.

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 below.

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. The multigroup transport equations are derived below, 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 such 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:

$$\begin{aligned} \psi_g(\boldsymbol{x}, \boldsymbol{\Omega}) &= \int_{E_g}^{E_{g-1}} \psi(\boldsymbol{x}, \boldsymbol{\Omega}, E) dE \\ &= \text{ angular flux for group } g, \text{ and} \end{aligned} \tag{2.12a}$$

$$\begin{aligned} \chi_g(\boldsymbol{x}) &= \int_{E_g}^{E_{g-1}} \chi(\boldsymbol{x}, E) dE \\ &= \text{ multigroup fission spectrum for group } g \ . \end{aligned} \tag{2.12b}$$

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

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

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:

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

where:

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

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

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

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, based on 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 lieu of the hatted exact cross sections in Eq. (2.14), we obtain the *multigroup transport equations*:

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

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':

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

and:

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

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 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 $\boldsymbol{\Omega}_m$ in the quadrature set, with $1 \le m \le M_N$. An angular weight w_m is associated with each $\boldsymbol{\Omega}_m$, which represents the area of the cone on the unit sphere with a central axis of $\boldsymbol{\Omega}_m$. A function $\psi(\boldsymbol{\Omega})$ is now represented as a discrete quantity:

$$\psi(\boldsymbol{\Omega}_m) = \psi_m, 1 \le m \le M_N \,,$$

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

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

Historically, 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 Chebyshev 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:

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

where

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

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.

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

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

$$P_{th} = \int_V \sum_{g=1}^{G} \sum_{m=1}^{M_N} \kappa \Sigma_{f,g}(\boldsymbol{x}) \psi_{m,g}(\boldsymbol{x}) w_m \, d\boldsymbol{x} \,, \tag{2.23b}$$

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

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 manner 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 of the algorithms used in MPACT in Chapter 3, the 2D/1D method is discussed in Chapter 4.

CHAPTER 3. SUMMARY OF ALGORITHMS

This chapter summarizes the algorithms MPACT uses to solve 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 1 describes the process by which MPACT solves the steady-state eigenvalue problem for k_{eff} and ϕ. In Algorithm 1, the user can choose (1) between 2D/1D and 2D, (2) whether or not to use CMFD, Gauss-Siedel, or Jacobi iteration in energy, and (3) 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 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 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, and a graphical flow chart of the iteration (see Figure 11.7) is provided.

Algorithm 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 equation, Eq. (4.10), 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. (4.12), as described in Algorithm 6.
  else if Jacobi, **then**
    Obtain $\phi^{(n+1)}$ by performing transport sweep(s) on the 2D radial transport equations, Eq. (4.12), as described in Algorithm 7.
  end if
end while

Algorithm 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 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 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: 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. 4).
    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

CHAPTER 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 [28], [23], [29], [27], [30], [79], [98], [25], [104], [22], [24].

- The group at the Korea Advanced Institute of Science and Technology (KAIST) (N. Z. Cho, G. S. Lee, C. J. Park, and colleagues) developed the "2D/1D fusion" method for the CRX code [28], [29], [27], [30], [25], [104]. 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, cladding, and the outer edges of the coarse cell.) Unlike 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 at the Korea Atomic Energy Research Institute (KAERI) (J. Y. Cho, H. G. Joo, K. S. Kim, and S. Q. Zee and colleagues), developed a different planar MOC solution-based 3D heterogeneous core method for the DeCART code [23], [79], [22], [24]. 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:

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

The buckling approximation occurs naturally in diffusion theory, thus suggesting 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 (i) transport physics is used to treat the radial variables, and (ii) 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* [79]. The 2D/1D moniker is used to identify that method and 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. However, 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, monoenergetic ($G = 1$) problems with isotropic scattering are considered. However, 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:

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

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

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

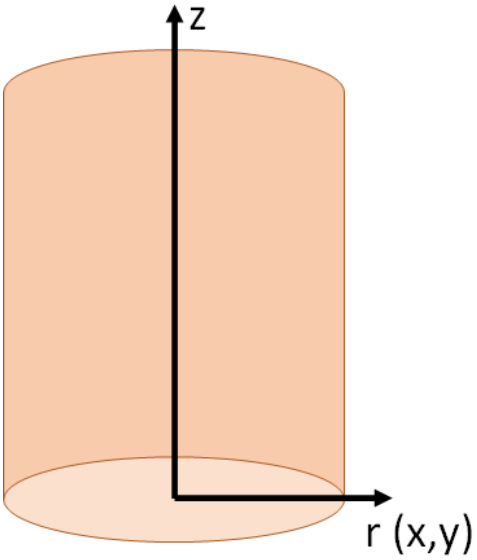


Figure 4.1. The 3D System V.

where R is a 2D region in the (x, y)-plane that is not necessarily circular.

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 are *radial*, and the z variable is *axial*.

Finally, a normalization condition is required. The solution to the transport equation is unique only in its shape, not in its magnitude. To address this, MPACT uses the following normalization condition:

$$\int_V \frac{\kappa\Sigma_f}{4\pi} \int_{4\pi} \Psi\left(\boldsymbol{x}, \boldsymbol{\Omega}'\right) d\Omega' dV = P \, , \tag{4.3}$$

where P is a specified constant, and $\kappa\Sigma_f$ is the energy released per fission. In practice, P is set equal to 1.0 at the beginning of the calculation. When post-processing the solution into data useful for reactor analysts, P can be set to the reactor's total power to obtain the magnitude of the flux as it exists in a real reactor.

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:

$$\begin{aligned} \Omega_x \frac{\partial \psi}{\partial x}(\boldsymbol{x}, \boldsymbol{\Omega}) + \Omega_y \frac{\partial \psi}{\partial y}(\boldsymbol{x}, \boldsymbol{\Omega}) + \Sigma_t \psi(\boldsymbol{x}, \boldsymbol{\Omega}) = \frac{\Sigma_s}{4\pi} \int_{4\pi} \psi(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega' \\ + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}} \int_{4\pi} \psi(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega' - \frac{1}{4\pi} \left[\frac{\partial J_z}{\partial z}(\boldsymbol{x}) \right]. \end{aligned} \tag{4.4}$$

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

$$J_z(\boldsymbol{x}) \approx \int_{4\pi} \Omega_z \psi(\boldsymbol{x}, \boldsymbol{\Omega}) d\Omega \; ; \tag{4.5}$$

the exact definition of J_z is discussed in Chapter 6 because it depends on the method used to solve the 1D axial equation. The boundary condition for Eq. (4.4) consists of a vacuum boundary condition on the cylindrical portion of ∂V:

$$\psi(\boldsymbol{x}, \boldsymbol{\Omega}) = 0 \text{ for } (x, y) \in \partial R \; , \; 0 \leq z \leq Z \; . \tag{4.6}$$

Operating on the solution of Eq. (4.4) by $\int_{4\pi} \Omega_x(\cdot) d\Omega$ and $\int_{4\pi} \Omega_y(\cdot) d\Omega$ yields the following radial currents:

$$J_x(\boldsymbol{x}) = \int_{4\pi} \Omega_x \psi(\boldsymbol{x}, \boldsymbol{\Omega}) d\Omega \; , \tag{4.7a}$$

$$J_y(\boldsymbol{x}) = \int_{4\pi} \Omega_y \psi(\boldsymbol{x}, \boldsymbol{\Omega}) d\Omega \; . \tag{4.7b}$$

The method MPACT uses to solve the radial equation is MOC, which is described in detail in Chapter 5.

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

$$\begin{aligned} \mu \frac{\partial \hat{\psi}}{\partial z}(\boldsymbol{x}, \boldsymbol{\Omega}) + \Sigma_t \hat{\psi}(\boldsymbol{x}, \boldsymbol{\Omega}) = \frac{\Sigma_s}{4\pi} \int_{4\pi} \hat{\psi}(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega' \\ + \frac{\nu \Sigma_f}{4\pi k_{\text{eff}}} \int_{4\pi} \hat{\psi}(\boldsymbol{x}, \boldsymbol{\Omega}') d\Omega' - \frac{1}{4\pi} \left[\frac{\partial J_x}{\partial x}(\boldsymbol{x}) + \frac{\partial J_y}{\partial y}(\boldsymbol{x}) \right] . \end{aligned} \tag{4.8}$$

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

$$\hat{\psi}(\boldsymbol{x}, \mu) \equiv \int_0^{2\pi} \hat{\psi}(\boldsymbol{x}, \boldsymbol{\Omega}) d\omega \; , \tag{4.9}$$

results in

$$\mu \frac{\partial \hat{\psi}}{\partial z}(\boldsymbol{x}, \mu) + \Sigma_t \hat{\psi}(\boldsymbol{x}, \mu) = \frac{\Sigma_s}{2} \int_{-1}^{1} \hat{\psi}(\boldsymbol{x}, \mu') d\mu' + \frac{\nu \Sigma_f}{2k_{\text{eff}}} \int_{-1}^{1} \hat{\psi}(\boldsymbol{x}, \mu') d\mu' - \frac{1}{2} \left[\frac{\partial J_x}{\partial x}(\boldsymbol{x}) + \frac{\partial J_y}{\partial y}(\boldsymbol{x}) \right] . \tag{4.10}$$

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

$$\hat{\psi}(\boldsymbol{x}, \mu) = 0 \text{ for } \begin{cases} z = Z & , \; -1 \leq \mu < 0 \; , \\ z = 0 & , \; 0 < \mu \leq 1 \; . \end{cases} \tag{4.11}$$

MPACT can use a variety of methods to solve the axial equation. These methods are detailed in Chapter 6.

The 2D/1D equations constitute a 2D radial transport equation coupled to 1D axial transport equations. The 2D radial transport and 1D axial transport 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 equations.

4.4. DISCRETIZATIONS

4.4.1 Energy Discretization

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 are 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 as a result of the neutrons that scatter into a given group g from other groups g'.

4.4.2 Angle Discretization

The direction-of-flight variable $\mathbf{\Omega}$ 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 2D 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 $1 \le n \le N_q$. The modular quadrature sets are used to angularly discretize the 2D radial transport Eq. (4.4) and to construct the radial currents in Eqs. (4.7).

4.4.3 Spatial Discretization

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. (4.4) and the 1D axial transport Eq. (4.10) are now integrated axially over the k^{th} slice $z_{k-1/2} < z < z_{k+1/2}$. The integration of Eq. (4.4) produces a 2D radial transport equation for the axially integrated ψ:

$$\begin{aligned}\Omega_x \frac{\partial \psi_{n,k}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{n,k}}{\partial y}(x,y) + \Sigma_t \psi_{n,k}(x,y) = \frac{\Sigma_s}{4\pi} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m \\ + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m - \frac{1}{4\pi \Delta z_k} [J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y)],\end{aligned} \tag{4.12}$$

where

$$\psi_{n,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} \psi(x,y,z,\mathbf{\Omega}_n) dz \, , \tag{4.13}$$

is the axially integrated ψ, and

$$J_{z,k\pm1/2}(x,y) = J_z(x,y,z_{k\pm1/2}) \, , \tag{4.14}$$

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

The axial integration of the 1D axial transport equation in Eq. (4.10) produces a similar result:

$$\begin{aligned}&\frac{\mu}{\Delta z_k} \left[\hat{\psi}(x,y,z_{k+1/2},\mu) - \hat{\psi}(x,y,z_{k-1/2},\mu) \right] + \Sigma_t \hat{\psi}_k(x,y,\mu) \\ &= \frac{\Sigma_s}{2} \int_{-1}^{1} \hat{\psi}_k(x,y,\mu') \, d\mu' + \frac{\nu\Sigma_f}{2k_{\text{eff}}} \int_{-1}^{1} \hat{\psi}_k(x,y,\mu') d\mu' - \frac{1}{2} \left[\frac{\partial J_{x,k}}{\partial x}(x,y) + \frac{\partial J_{y,k}}{\partial y}(x,y) \right],\end{aligned} \tag{4.15}$$

where

$$\hat{\psi}_k(x,y,\mu) = \frac{1}{\Delta z_k}\int_{z_{k-1/2}}^{z_{k+1/2}} \hat{\psi}(x,y,z,\mu)dz \,, \tag{4.16}$$

is the axially integrated $\hat{\psi}$, and

$$J_{x,k}(x,y) = \frac{1}{\Delta z_k}\int_{z_{k-1/2}}^{z_{k+1/2}} J_x(x,y,z)dz \,, \tag{4.17a}$$

$$J_{y,k}(x,y) = \frac{1}{\Delta z_k}\int_{z_{k-1/2}}^{z_{k+1/2}} J_y(x,y,z)dz \,, \tag{4.17b}$$

are the axially integrated radial currents.

Operating on Eq. (4.12) by

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

we obtain

$$\frac{\partial J_{x,k}}{\partial x}(x,y) + \frac{\partial J_{y,k}}{\partial y}(x,y) + \Sigma_a\phi_k(x,y) = \frac{\nu\Sigma_f}{k_{\text{eff}}}\phi_k(x,y) - \frac{1}{\Delta z_k}[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y)], \tag{4.18}$$

where

$$\phi_k(x,y) = \sum_{m=1}^{N_q}\psi_{m,k}(x,y)w_m \,. \tag{4.19}$$

Similarly, operating on Eq. (4.15) with $\int_{-1}^{1}(\cdot)\,d\mu$ gives

$$\begin{aligned}\frac{1}{\Delta z_k}\left[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y)\right] + \Sigma_a\hat{\phi}_k(x,y) \\ = \frac{nu\Sigma_f}{k_{\text{eff}}}\hat{\phi}_k(x,y) - \left(\frac{\partial J_{x,k}}{\partial x}(x,y) + \frac{\partial J_{y,k}}{\partial y}(x,y)\right) \,.\end{aligned} \tag{4.20}$$

Because the leakage terms in Eqs. (4.18) and (4.20) are identical, it follows that

$$\phi_k(x,y) = \hat{\phi}_k(x,y) \,. \tag{4.21}$$

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

Equation (4.12) 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. (4.12) completely determines $\psi_{n,k}(x,y)$. However, Eq. (4.15) alone cannot determine both $\hat{\psi}_k(x,y,\mu)$ and $J_{z,k\pm1/2}(x,y)$; at least one extra equation is needed. This requires a solver for Eq. (4.15) that can determine the relationship between the flux and current terms. MPACT has a variety of transport- and diffusion-based methods to accomplish this task. These methods are discussed in greater detail in Chapter 6.

The discretizations of Eqs. (4.12) and (4.15) 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.

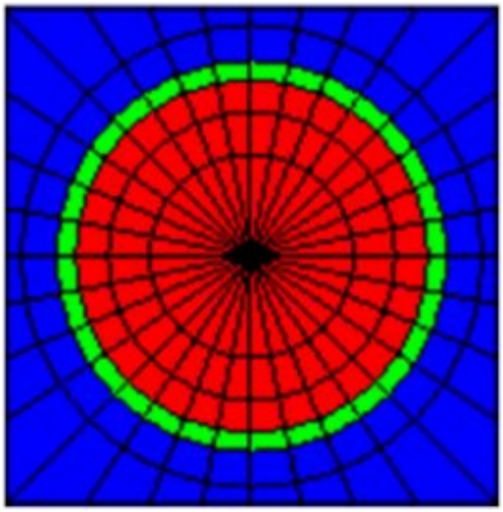


Figure 4.2. A typical coarse spatial cell with its fine spatial cells.

Second, the axial transport Eq. (4.15) 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 transport 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. (4.12) is discretized on the fine radial grid for each coarse cell $V_{j,k}$ using MOC. However, the axial current terms in this equation are obtained from the axial transport 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. (4.12) utilizes balance equations obtained by integrating Eq. (4.12) over each fine cell $v_{i,j,k}$. 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 Eq. (4.15) over the cell. The flux-weighted coarse-cell homogenized cross sections used in the axial transport 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 transport 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 transport 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 transport equation, and (iii) applying 3D coarse mesh finite difference (CMFD) acceleration (described in Chapter 7).

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.)

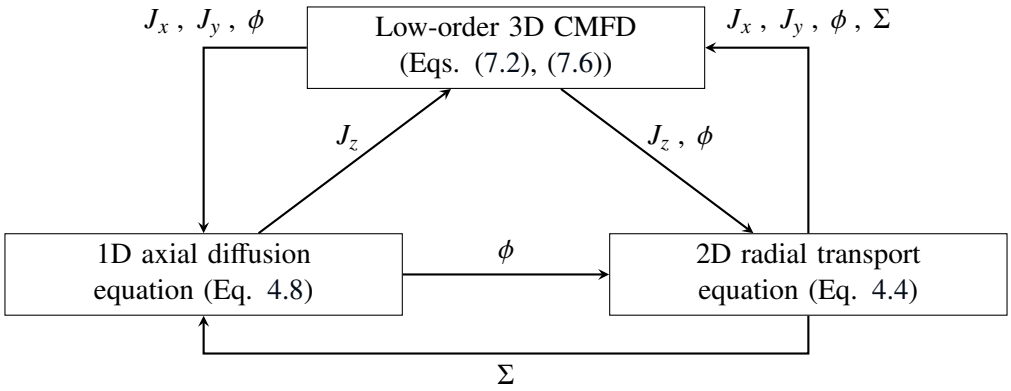


Figure 4.3. MPACT 2D/1D iteration strategy.

(i) First, the $\hat{D}$ terms are calculated from Eq. (7.8) using coarse grid scalar flux and current estimates obtained from the previous 1D axial transport 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 transport equations is performed using the coarse-grid edge current estimates from step (i). This solve can include 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, then 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, then the iteration process is terminated. Otherwise, the computer returns to step (i), and another iteration is initiated.

In Chapters 5, 6, and 7, each component of the above iteration strategy will be described in much greater detail.

CHAPTER 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 highlighting important approximations at each step is provided, 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. (4.12) and does not assume isotropic scattering. It does assume that the incoming angular flux on the top and bottom of the 2D slice is isotropic, allowing the currents $J_{z,k+1/2,g}(x,y)$ and $J_{z,k-1/2,g}(x,y)$ to be used instead of the angular fluxes.

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

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

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

going forward:

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

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 dependent variables 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 characteristic form of Eq. (5.3),

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

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,

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

where

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

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 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.

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

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

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

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

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

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 is 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:

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

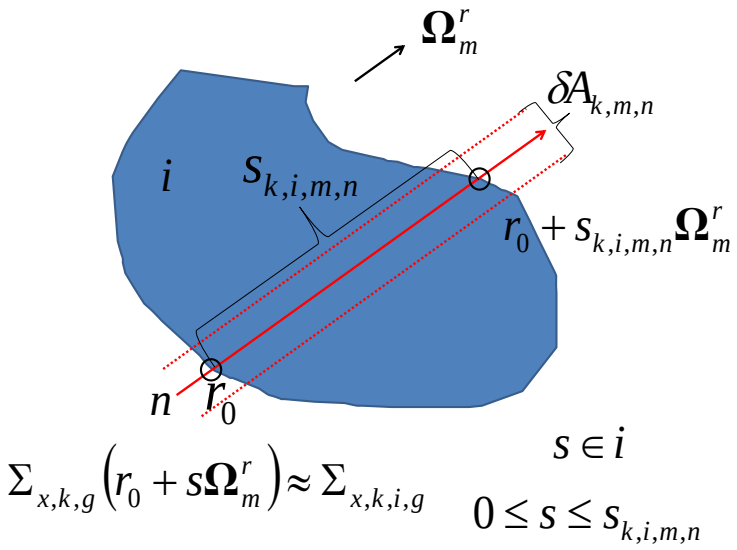


Figure 5.1. Spatial discretization with constant properties.

and q:

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

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 Eqs. (5.7) and (5.14) are identical for their respective spatial dimensionality. In the following, Section 5.3, approximations to allow for the evaluation 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, Eqs. (5.6) and (5.7) reduce to the following equations for each characteristic ray k passing through each discrete region i:

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

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

In Eq. (5.15), the shorthand notations,

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

and

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

are 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 very widely used. It is accurate in the fine limit of the spatial mesh. Recently, a linear source approximation was also implemented in MPACT. The details of the linear source approximation are described at length in Section 5.3.4.

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:

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

Equations (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. This source is usually represented by expanding the differential scattering cross section as an infinite series of Legendre polynomials of the cosine of the scattering angle to account for anisotropic scattering. For LWR analysis, linearly anisotropic 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 omitted). The approximation described here is also identical to the flat component of the linear source approximation described in subsection 5.3.4.4.

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

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

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

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

where $q_{0,g}$ is the total isotropic source including 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:

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

where

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

and the elements of $\boldsymbol{J}$ are defined by Eqs. (4.7) and (4.5) 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* [157]. In the outscatter method, the quantity $\bar{\mu}_0\Sigma_s$ is subtracted from 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

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

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

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

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

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

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 approximate an anisotropic scattering operator with an isotropic scattering operator. More information regarding these methods can be found in a publication by Yee (2016) [159].

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:

$$q_g^{tr} = \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}\phi_{g'} + \frac{1}{4\pi} \sum_{g'=1}^{G} \Sigma_{s0,g'\to g}^{tr}\phi_{g'} \ , \tag{5.27}$$

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.

5.3.4 Linear Source Approximation

5.3.4.1 Overview

The *linear-source approximation* (LSA) in the MOC assumes that the shape of the source along a characteristic track-segment is linear. Development of LSA for the MOC has been a long-time goal, because previous work [101] indicated that a spatially linear source would be able to achieve more accurate computational performance in S_N calculations. Many different variants of this approximation have been introduced. The first instance of the LSA was the *gradient source approximation* introduced by Halsall [69]. This early linear-source MOC (LSMOC) was based on the averaging of the angular flux gradient along tracks and was implemented in the WIMS [69] and PEACH [147] MOC transport codes. These averaged gradients were used as estimates to the gradient of the scalar flux to compute the source shape as spatially linear.

Petkov devised an LSA that estimated the gradient of the scalar flux based on the P_1 approximation in the MARIKO code [118, 119]. In this approximation, the gradient of the scalar flux is computed from the neutron current, the total cross section, and the linearly anisotropic scattering matrix:

$$\boldsymbol{\nabla}\phi_i^g \approx -3\left(\Sigma_{t,i}^g \boldsymbol{J}_i^g - \sum_{g'} \Sigma_{s,1,i}^{g'\to g} \boldsymbol{J}_i^{g'}\right) . \tag{5.29}$$

A similar approach using the diffusion approximation to compute the scalar flux gradient was used in the so called "quasi-linear" source implemented by Rabiti [124]. In this approach, the $\Sigma_{s,1,i}^{g'\to g}$ matrix is diagonalized, turning the P_1 approximation into the diffusion approximation—similar to the outflow approximation described in 5.3.3. Because they were based on the P_1 and diffusion approximations, these early LSA methods were eventually assessed to be inaccurate for situations with more transport-like effects. (It can be shown, even in simple cases, that this approximation can be predict the opposite direction for the scalar flux gradient).

Santandrea [129] introduced the positive linear and nonlinear surface characteristics scheme, which constructed a linear source by interpolating between source values on the surfaces of cell regions. Various improvements have been made to this surface characteristics scheme for conservation [129], as well as coupling in APOLLO2 [130]. Tellier introduced a simplification to the linear characteristics scheme for conservation by using a diamond-differencing scheme [102]. This work was extended by Hébert [71] to include higher order diamond difference schemes and to allow for acceleration [72].

The most recent LSA examined in this work was introduced as a 2D general high-order method for unstructured meshes by Masiello [113]. The approximation uses track-based integration to compute spatial moments of the angular flux. This LSA was shown to reduce memory and computation times in 3D MOC calculations [17]. The general method was simplified in the case of the isotropic and anisotropic LSA by Ferrer [53]; this also introduced the LS-P0 method, in which the isotropic source is spatially linear, but the anisotropic source components are spatially uniform within each cell. This LSA was also shown to be consistent with particle conservation under certain constraints and to be compatible with CMFD acceleration [54].

The LSA in MPACT is based on the derivation of the moment-based LSA from Ferrer [53,54]. The original derivation faced inefficiencies in problems with non-constant cross sections. Multiphysics applications were not of interest to the original authors [53]. The inefficiencies arose from the use of pre-computed coefficients that must be re-computed if the cross sections change. The computation of these coefficients was approximately 10–20% of the run-time cost of a MOC sweep. In multiphysics calculations, such as those with thermal hydraulic feedback, cross sections typically change every iteration, leading to significant overhead from re-computing these terms. The next section presents an equivalent formulation that eliminates the need to re-compute these terms if cross sections change, without additional operations.

Various different LSAs to the MOC have been developed and implemented in transport codes [15,53,57,69,118,124,129,147]. Results have indicated that by using an LSA, the spatial mesh discretization can be made coarser relative to the flat source approximation (FSA) while maintaining transport accuracy. Although each segment calculation is more expensive when using a LSA, the number of calculations (resulting from the coarser spatial mesh) can be significantly reduced, leading to reduced run-times. Additionally, the reduction in spatial mesh elements generally reduces the amount of memory required by the calculation. This is possible because the LSA produces a more accurate solution than the FSA for any given spatial mesh.

5.3.4.2 Derivation

The moment-based LSA assumes that the shape of the source, $q^g_{mi}(\boldsymbol{x})$, is spatially linear within each cell, $\mathcal{R}_i$. This can be expressed as

$$q^g_{mi}(\boldsymbol{x}) \approx q^g_{mi} + \boldsymbol{x} \cdot \underline{\widehat{\boldsymbol{q}}}^g_{mi} \,, \tag{5.30a}$$

where $\underline{\widehat{\boldsymbol{q}}}^g_{mi}$ is a column vector of source spatial expansion coefficients,

$$\underline{\widehat{\boldsymbol{q}}}^g_{mi} \equiv \begin{bmatrix} \widehat{q}^g_{mi,x} \\ \widehat{q}^g_{mi,y} \\ \widehat{q}^g_{mi,z} \end{bmatrix} \,, \tag{5.30b}$$

and $\boldsymbol{x}$ is the position in *local* coordinates. A similar spatial expansion of the angular moments of the flux can be performed,

$$\phi^{\ell,g}_{i,n}(\boldsymbol{x}) = \overline{\phi}^{\ell,g}_{i,n} + \boldsymbol{x} \cdot \underline{\widehat{\boldsymbol{\phi}}}^{\ell,g}_{i,n}, \tag{5.31}$$

the source can then be expressed as

$$q^g_{mi}(\boldsymbol{x}) = \sum_{g'} \sum_{\ell=0}^{L} \sum_{n=-\ell}^{\ell} R^n_\ell(\widehat{\boldsymbol{\Omega}}_m) \Sigma^{g' \to g}_{s,\ell,i} \phi^{\ell,g'}_{i,n}(\boldsymbol{x}) + \frac{\chi^g_i}{k_{\text{eff}}} \sum_{g'} \nu \Sigma^{g'}_{f,i} \phi^{g'}_i(\boldsymbol{x}) \,, \tag{5.32}$$

and the linear expansion coefficients are explicitly given by

$$\widehat{\underline{q}}^{g}_{mi} = \sum_{g'} \sum_{\ell=0}^{L} \sum_{n=-\ell}^{\ell} R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}_m) \Sigma^{g'\to g}_{s,\ell,i} \widehat{\underline{\boldsymbol{\phi}}}^{\ell,g'}_{i,n} + \frac{\chi^{g}_{i}}{k_{\text{eff}}} \sum_{g'} \nu\Sigma^{g'}_{f,i} \widehat{\underline{\boldsymbol{\phi}}}^{g}_{g'} \,. \tag{5.33}$$

In the spatial moment-based LSA, it is convenient to define the spatially linear source (and flux) in terms of a cell-local coordinate system. We allow $\boldsymbol{X}$ to be the position variable in the global coordinate system, and the local coordinates are then defined as

$$\boldsymbol{x} = \boldsymbol{X} - \boldsymbol{X}^{\text{c}}_{mi} \,, \tag{5.34}$$

where $\boldsymbol{X}^{\text{c}}_{mi}$ is the numerical centroid of the cell i.

These numerical centroids can be defined as either direction-dependent or direction-independent, which will have implications on particle conservation. The direction-dependent centroids are defined by

$$\boldsymbol{X}^{\text{c}}_{mi} \equiv \langle \boldsymbol{X} \rangle_{mi} = \frac{1}{V_i} \sum_{k} \delta A_{mki} t_{mki} \boldsymbol{X}^{\text{c}}_{mki} \,, \tag{5.35}$$

where $\boldsymbol{X}^{\text{c}}_{mki}$ is the global coordinate vector of the segment mid-point. Similarly, the direction-independent centroids are defined by

$$\boldsymbol{X}^{\text{c}}_{i} \equiv \frac{1}{4\pi} \langle \boldsymbol{X} \rangle_{i} = \frac{1}{V_i} \sum_{m} w_m \sum_{k} \delta A_{mki} t_{mki} \boldsymbol{X}^{\text{c}}_{mki} \,. \tag{5.36}$$

Following the same approach as that used for the flat source MOC (FSMOC) derivation, computing the source requires the region-averaged flux moments, $\overline{\phi}^{\ell,g}_{i,n}$, and the flux expansion coefficients, $\widehat{\underline{\boldsymbol{\phi}}}^{\ell,g}_{i,n}$. The region-averaged flux moment can be found using the same definition as was used previously:

$$\overline{\phi}^{\ell,g}_{i,n} \equiv \left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \psi^{g} \right\rangle_{i} = \frac{4\pi}{V_i} \sum_{m} w_m R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}_m) \sum_{k} \delta A_{mki} t_{mki} \langle \psi^{g} \rangle_{mki} \,. \tag{5.37a}$$

To determine the flox moment spatial expansion coefficients, Eq. (5.31) is operated on by $\left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x}(\cdot) \right\rangle_{i}$. Recognizing that this should be directly proportional to angular flux operated on by $\left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x} \psi^{g} \right\rangle_{i}$, a system of equations is found

$$\boldsymbol{M}_i \widehat{\underline{\boldsymbol{\phi}}}^{\ell,g}_{i,n} = \left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x} \psi^{g} \right\rangle_{i} \,, \tag{5.37b}$$

where

$$\boldsymbol{M}_i \equiv \left\langle \boldsymbol{x} \boldsymbol{x}^{T} \right\rangle_{i} \,. \tag{5.37c}$$

The spatial angular flux moments, $\left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x} \psi^{g} \right\rangle_{i}$, are then defined as

$$\left\langle R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x} \psi^{g} \right\rangle_{i} = \frac{4\pi}{V_i} \sum_{m} w_m R^{n}_{\ell}(\widehat{\boldsymbol{\Omega}}_m) \sum_{k} \delta A_{mki} t_{mki} \left(\boldsymbol{x}^{\text{in}}_{mki} \langle \psi^{g} \rangle_{mki} + \widehat{\boldsymbol{\Omega}}_m \langle t_m \psi^{g} \rangle_{mki} / \xi_{mi} \right) \,. \tag{5.37d}$$

To evaluate the flux moments defined in Eq. (5.37), the track-averaged angular flux values, $\langle \psi^{g} \rangle_{mki}$, and $\langle t_m \psi^{g} \rangle_{mki}$, must be determined. First, the transport equation must be put into characteristic form. The spatially expanded source, Eq. (5.30a), can be defined along the characteristic. The characteristic transport equation becomes

$$\left[\frac{\text{d}}{\text{d}t_m} + \Sigma^{g}_{t,i} \right] \psi^{g}_{mki}(s) = \overline{q}^{g}_{mki} + \widehat{q}^{g}_{mi} \left(t_m - \frac{t_{mki}}{2} \right) \,, \tag{5.38a}$$

where

$$\overline{q}^g_{mki} \equiv \frac{1}{4\pi}\left[q^g_{mi} + \boldsymbol{x}^{\mathrm{c}}_{mki} \cdot \underline{\widehat{q}}^g_{mi}\right] , \tag{5.38b}$$

$$\widehat{q}^g_{mi} \equiv \frac{1}{4\pi}\left[\frac{\widehat{\boldsymbol{\Omega}}_m \cdot \underline{\widehat{q}}^g_{mi}}{\xi_{mi}}\right] , \tag{5.38c}$$

and $\boldsymbol{x}^{\mathrm{c}}_{mki}$ is the local-coordinate centroid of the track-segment. The angular flux along a track-segment is found to be

$$\psi^g_{mki}(s) = \psi^{g,\mathrm{in}}_{mki} + \left(\frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}} - \psi^{g,\mathrm{in}}_{mki}\right) F_1(\tau^g_m) + \frac{\widehat{q}^g_{mi}}{2(\Sigma^g_{t,i})^2} F_2(\tau^g_m) , \tag{5.39a}$$

where

$$F_1(\tau^g_m) \equiv 1 - \exp(-\tau^g_m) , \tag{5.39b}$$

and

$$F_2(\tau^g_m) \equiv 2[\tau^g_m - F_1(\tau^g_m)] - \tau^g_{mki} F_1(\tau^g_m) . \tag{5.39c}$$

To evaluate the moments defined in Eq. (5.37), the track-average flux and first spatial moment of the angular flux must be determined. The track-average flux is determined as it was before, by operating on Eq. (5.38a) by $t_{mki}\langle(\cdot)\rangle_{mki}$, yielding

$$\langle\psi^g\rangle_{mki} = \frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}} + \frac{\Delta\psi^g_{mki}}{\tau^g_{mki}} . \tag{5.40a}$$

At this point, the derivation deviates from that described in [53]. In the previous formulation, the integral in $\langle t_m\psi^g\rangle_{mki}$ is *explicitly* evaluated by substituting in the solution of the angular flux along the track (Eq. (5.39)). Here, the moment will be found implicitly by operating on Eq. (5.38a) by $t_{mki}\langle t_m(\cdot)\rangle_{mki}$, just as was done for the 0th moment. This results in

$$\langle t_m\psi^g\rangle_{mki} = \frac{\langle\psi^g\rangle_{mki} - \psi^{g,\mathrm{out}}_{mki}}{\Sigma^g_{t,i}} + \frac{t_{mki}}{2}\left[\frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}} + \frac{\widehat{q}^g_{mi}}{\Sigma^g_{t,i}}\frac{t_{mki}}{6}\right] . \tag{5.40b}$$

Unlike the original form by Ferrer [53], no new exponential functions are introduced in this form.

Previously, the average track flux, $\langle\psi^g\rangle_{mki}$, was expanded to find a simpler final form. This is the case here, as well; however, the outgoing flux, $\psi^{g,\mathrm{out}}_{mki}$, will not be expanded, as this must necessarily be computed during transmission. Eq. (5.37) are then simplified into

$$\overline{\phi}^{\ell,g}_{i,n} = \frac{4\pi}{V_i\Sigma^g_{t,i}} \sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \sum_k \delta\mathcal{A}_{mki}\left(t_{mki}\overline{q}^g_{mki} + \Delta\psi^g_{mki}\right) , \tag{5.41a}$$

and

$$\begin{aligned}\left\langle R^n_\ell(\widehat{\boldsymbol{\Omega}})\boldsymbol{x}\psi^g\right\rangle_i =& \frac{1}{V_i}\sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \sum_k \delta\mathcal{A}_{mki} t_{mki}\left[\boldsymbol{x}^{\mathrm{c}}_{mki} q^g_i + \left(\boldsymbol{x}^{\mathrm{c}}_{mki}(\boldsymbol{x}^{\mathrm{c}}_{mki})^T + \frac{s^2_{mki}}{12}\widehat{\boldsymbol{\Omega}}_m\widehat{\boldsymbol{\Omega}}^T_m\right)\underline{\widehat{q}}^g_i\right] \\ &+ \frac{4\pi}{V_i\Sigma^g_{t,i}}\sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \sum_k \delta\mathcal{A}_{mki}\left[\boldsymbol{x}^{\mathrm{in}}_{mki}\Delta\psi^g_{mki} + \widehat{\boldsymbol{\Omega}}_m s_{mki}\left(\frac{\Delta\psi^g_{mki}}{\tau^g_{mki}} - \psi^{g,\mathrm{out}}_{mki} + \frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}}\right)\right] .\end{aligned} \tag{5.41b}$$

Eq. (5.41b) contains $\Delta\psi^g_{mki}/\tau^g_{mki}$ terms rather than only $\Delta\psi^g_{mki}$ terms. For numerical stability, it is beneficial to compute this quantity directly rather than $\Delta\psi^g_{mki}$ and performing division. This can be found by evaluating the transmission equation, Eq. (5.39a), giving

$$\frac{\Delta\psi^g_{mki}}{\tau^g_{mki}} = \left(\psi^{g,\mathrm{in}}_{mki} - \frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}}\right) E_1(\tau^g_{mki}) - \frac{t_{mki}}{2}\frac{\widehat{q}^g_{mi}}{\Sigma^g_{t,i}} T_2(\tau^g_{mki}) , \tag{5.42a}$$

where

$$E_1(\tau^g_{mki}) \equiv \frac{F_1(\tau^g_{mki})}{\tau^g_{mki}} \,, \tag{5.42b}$$

$$T_2(\tau^g_{mki}) \equiv 2E_2(\tau^g_{mki}) - E_1(\tau^g_{mki}) \,, \tag{5.42c}$$

and

$$E_2(\tau^g_{mki}) \equiv \frac{1 - E_1(\tau^g_{mki})}{\tau^g_{mki}} \,. \tag{5.42d}$$

Here, $E_2(\tau^g_{mki})$ is defined as an intermediate function with smaller derivative terms, meaning higher accuracy with fewer interpolation intervals. Only the $E_2(\tau^g_{mki})$ function must be tabulated. Then this is used to compute the other two exponential functions, $E_1(\tau^g_{mki})$, and $T_2(\tau^g_{mki})$. The outgoing flux can then be evaluated as

$$\psi^{g,\text{out}}_{mki} = \psi^{g,\text{in}}_{mki} - \tau^g_{mki} \frac{\Delta\psi^g_{mki}}{\tau^g_{mki}} \,. \tag{5.43}$$

5.3.4.3 Particle Conservation

When considering particle conservation, use of the LSA results in additional constraints on the calculations. The track-based integration of the source must exactly integrate to the spatial and angular moments of the source. The conservation of spatial moments is the basis of this LSA [54], so this constraint is satisfied without additional constraints on the method. The angular moment constraint is expressed as

$$\frac{1}{4\pi}\left\langle R^n_\ell(\widehat{\boldsymbol{\Omega}}) q^g_{mi}(\boldsymbol{x}) \right\rangle_i = q^{g,n}_{i,\ell} \,. \tag{5.44}$$

In addition to these constraints for the direction-dependent track renormalization (see Eq. (5.50)) and directional quadrature restrictions, there is a constraint on the definition of the local coordinate system:

$$\langle \boldsymbol{x} \rangle_{mi} = 0 \,. \tag{5.45}$$

This is equivalent to stating that the local coordinate system must be defined with respect to the direction-dependent global centroids. If these constraints are satisfied, then Eq. (5.41) can be simplified into

$$\overline{\phi}^{\ell,g}_{i,n} = \sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \frac{q^g_i}{\Sigma^g_{t,i}} + \frac{4\pi}{V_i \Sigma^g_{t,i}} \sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \sum_k \delta A_{mki} \Delta\psi^g_{mki} \,, \tag{5.46a}$$

and

$$\begin{aligned} \left\langle R^n_\ell(\widehat{\boldsymbol{\Omega}}) \boldsymbol{x} \psi^g \right\rangle_i = &\sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \boldsymbol{M}_{mi} \frac{\widehat{\boldsymbol{q}}^g_i}{\Sigma^g_{t,i}} \\ &+ \frac{4\pi}{V_i \Sigma^g_{t,i}} \sum_m w_m R^n_\ell(\widehat{\boldsymbol{\Omega}}) \sum_k \delta A_{mki} \left[\boldsymbol{x}^{\text{in}}_{mki} \Delta\psi^g_{mki} + \widehat{\boldsymbol{\Omega}}_m s_{mki} \left(\frac{\Delta\psi^g_{mki}}{\tau^g_{mki}} - \psi^{g,\text{out}}_{mki} + \frac{\overline{q}^g_{mki}}{\Sigma^g_{t,i}} \right) \right] \,, \end{aligned} \tag{5.46b}$$

where

$$\boldsymbol{M}_{mi} \equiv \frac{1}{V_i} \sum_k \delta A_{mki} \left[\boldsymbol{x}^{\text{c}}_{mki} (\boldsymbol{x}^{\text{c}}_{mki})^T + \frac{s^2_{mki}}{12} \widehat{\boldsymbol{\Omega}}_m \widehat{\boldsymbol{\Omega}}^T_m \right] \,. \tag{5.46c}$$

Although the above form seems to be more computationally efficient, it would be remiss to not include a more mathematically elegant form of Eq. (5.41b). The terms in the second summation may be rewritten as

$$\begin{aligned}\left\langle R_\ell^n(\widehat{\boldsymbol{\Omega}})\boldsymbol{x}\psi^g\right\rangle_i &= \sum_m w_m R_\ell^n(\widehat{\boldsymbol{\Omega}})\boldsymbol{M}_{mi}\frac{\widehat{\overline{\boldsymbol{q}}}_i^g}{\Sigma_{t,i}^g} \\ &+ \frac{4\pi}{V_i\Sigma_{t,i}^g}\sum_m w_m R_\ell^n(\widehat{\boldsymbol{\Omega}})\sum_k \delta A_{mki}\left[\boldsymbol{x}_{mki}^{\text{in}}\left(\psi_{mki}^{g,\text{in}}-\overline{\psi}_{mki}^g\right)-\boldsymbol{x}_{mki}^{\text{out}}\left(\psi_{mki}^{g,\text{out}}-\overline{\psi}_{mki}^g\right)\right],\end{aligned} \tag{5.47a}$$

where $\overline{\psi}_{mki}^g$ is the average track flux,

$$\overline{\psi}_{mki}^g \equiv \langle\psi^g\rangle_{mki}\,. \tag{5.47b}$$

In this form, the interior summation over track-segments becomes a spatially weighted average of the incident and outlet flux differences from the average flux.

5.3.4.4 Isotropic Simplifications

Ferrer's work suggests that allowing only the flat source components to consider anisotropic scattering has performance benefits, although it does not significantly affect accuracy [53]. It was demonstrated for the B&W experiments [18] that considering anisotropic scattering only with spatially flat flux moments resulted in an error of approximately 10 pcm. Furthermore, by making this simplification, run-times can be reduced significantly (up to 45%), and memory savings were even more significant (up to 89%) [53].

If the so-called linear-isotropic-flat-anisotropic (LIFA) scheme is used, then the spatially flat moment equations do not change. However, Eq. (5.41b) can be simplified to

$$\begin{aligned}\left\langle R_\ell^n(\widehat{\boldsymbol{\Omega}})\boldsymbol{x}\psi^g\right\rangle_i &= \boldsymbol{M}_i\frac{\widehat{\overline{\boldsymbol{q}}}_i^g}{\Sigma_{t,i}^g} \\ &+ \frac{4\pi}{V_i\Sigma_{t,i}^g}\sum_m w_m\sum_k \delta A_{mki}\left[\boldsymbol{x}_{mki}^{\text{in}}\Delta\psi_{mki}^g+\widehat{\boldsymbol{\Omega}}_m s_{mki}\left(\frac{\Delta\psi_{mki}^g}{\tau_{mki}^g}-\psi_{mki}^{g,\text{out}}+\frac{\overline{q}_{mki}^g}{\Sigma_{t,i}^g}\right)\right].\end{aligned} \tag{5.48}$$

If isotropic scattering is used, then the scalar flux moments can be found as

$$\begin{aligned}\langle\boldsymbol{x}\psi^g\rangle_i &= \boldsymbol{M}_i\frac{\widehat{\overline{\boldsymbol{q}}}_i^g}{\Sigma_{t,i}^g} \\ &+ \frac{4\pi}{V_i\Sigma_{t,i}^g}\sum_m w_m\sum_k \delta A_{mki}\left[\boldsymbol{x}_{mki}^{\text{in}}\Delta\psi_{mki}^g+\widehat{\boldsymbol{\Omega}}_m s_{mki}\left(\frac{\Delta\psi_{mki}^g}{\tau_{mki}^g}-\psi_{mki}^{g,\text{out}}\right)\right].\end{aligned} \tag{5.49}$$

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 because they are not specific to MOC. The spatial discretization is generally 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 particle flight paths (characteristic tracks) 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

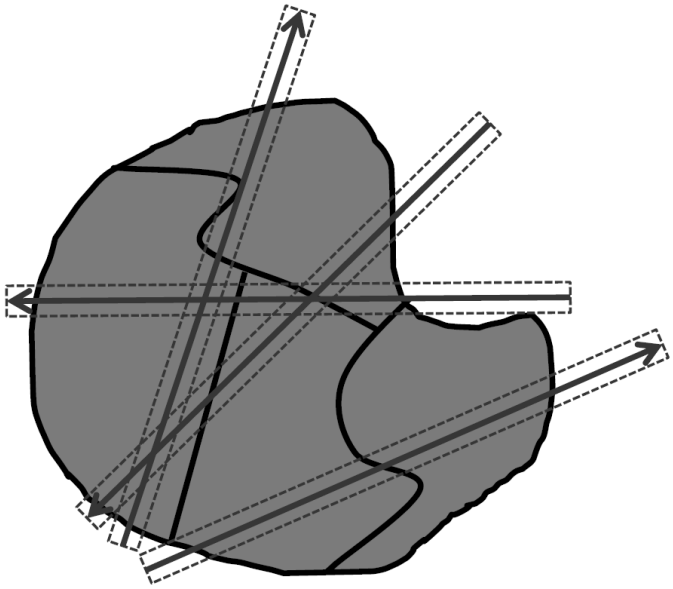


Figure 5.2. Characteristic rays intersecting a set of discrete spatial regions.

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*.

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 solution accuracy. Several possible choices may be made in the design of the ray-tracing algorithm. Choices specific to the algorithm in MPACT are discussed in the remainder of this section.

For LWR analysis, when developing the ray-tracing algorithm, it is possible to take advantage of the fact that reactors generally have a high degree of regularity in their geometry. 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 depicts nine *ray-tracing modules* denoted as black squares. The *modular rays*, which are depicted as blue lines, are defined only within the ray-tracing module, and they 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.

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 for a given angle, although this may not be the case. Furthermore, when modular ray-tracing is used, an additional setup step is required

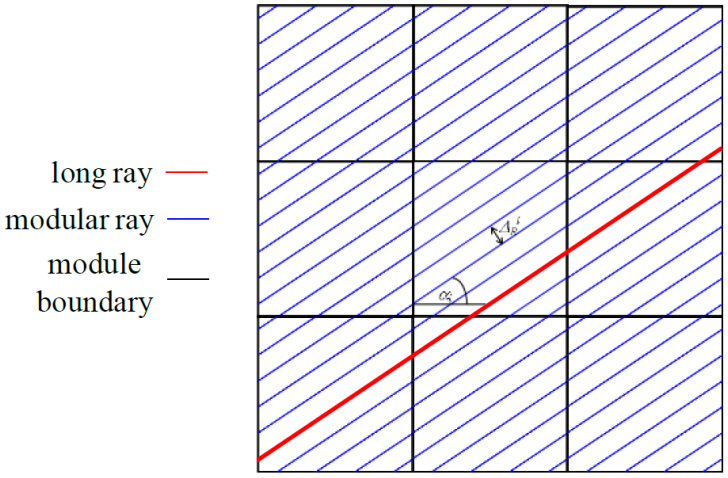


Figure 5.3. Modular ray-tracing concept in 2D.

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, then 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 to 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 above. Therefore, the approach in MPACT is to start from an existing angular quadrature that introduces small perturbations to the discrete angles so that they satisfy the modularity condition. This is analogous to 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 to 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.

In addition to storing the cross sectional area of the rays, a choice must be made as to whether other ray-tracing data should be stored, 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

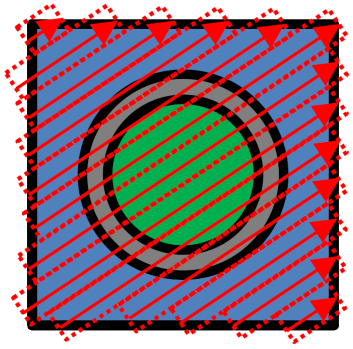


Figure 5.4. Uniformly spaced characteristic rays through a pin cell.

stored and accessed. In this case, a trade-off between increased memory storage and repeated computations must be optimized. For any 2D MOC implementation, the choice is invariably to compute the ray-tracing information once and store it. However, for 3D MOC, some combination may be optimal. Storing the ray-tracing data has the benefit of decoupling the sweep algorithm from the ray-trace algorithm, which allows each algorithm 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 is an approximation of the region volume. This is shown in Figure 5.5. This is generally a problem of numerical

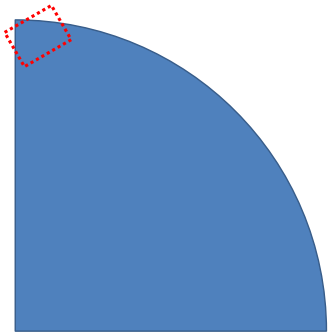


Figure 5.5. Numerical integration of a region volume by ray segments.

integration stemming from the discrete 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 is sometimes referred to as *segment renormalization*, and it can be implemented several ways. Two common approaches are to renormalize the segment lengths where (i) the renormalization is performed

for each angle separately, and (ii) the renormalization is computed for all angles. These renormalizations are given by Eqs. (5.50) and (5.51), respectively:

$$\bar{s}_{k,i,m,n} = s_{k,i,m,n}\left(\frac{V_i}{\sum\limits_{n\in i} s_{k,i,m,n}\delta A_{k,m}}\right), \tag{5.50}$$

$$\bar{s}_{k,i,m,n} = s_{k,i,m,n}\left(\frac{4\pi V_i}{\sum\limits_{m}\sum\limits_{n\in i} s_{k,i,m,n}\omega_m\delta A_{k,m}}\right), \tag{5.51}$$

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. Although this approach is common, it is certainly questionable in this regard. As an alternative to renormalizing the segment lengths, one can choose to use no renormalization, although there may be a slightly higher error from an inadequate spatial discretization. It is important to recall here that angle-dependent renormalization is required to have particle conservation with the LSA, as described in Section 5.3.4.3.

Experience suggests that the normalization of Eq. (5.50) 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. (5.51), rather than once for each angle, as in Eq. (5.50), or it can be neglected entirely. The Eq. (5.51) 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. (5.50) may lead to calculation stability issues. The renormalization method of Eq. (5.51) 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 particle balance. The default method in MPACT performs no volume correction and leaves the segment lengths unadjusted. Based on an examination of 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 are described in detail in other parts of this manual. However, this section describes the source iteration procedure and 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 *inner* and *outer* iterations. The inner iteration assumes a fixed source and converges the self-scattering term, whereas the outer iteration performs an update of the eigenvalue. To facilitate this iteration scheme, the computation of the source is divided into two parts:

$$\bar{q}^{tr}_{g,i} = \frac{1}{4\pi}\left(Q_{ext,i,g} + \Sigma^{tr}_{s0,i,g\rightarrow g}\phi_{i,g}\right). \tag{5.52}$$

where

$$Q_{ext,i,g} = \sum_{g'=1,g'\neq g}^{G} \Sigma^{tr}_{s0,i,g'\rightarrow g}\phi_{i,g'} + \frac{\chi_{i,g}}{k}\sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'} \ . \tag{5.53}$$

In the 1-group fixed source problem, the known external source $Q_{ext,i,g}$ is the source from fission and *in-scatter*, or scattering from group $g' \neq g$ to group g, and is a function of the scalar flux. Equation (5.52) is updated during the inner iteration, and Eq. (5.53) is updated during the outer iteration. The inner iteration for a 1-group problem is given by Algorithm 4.

Algorithm 4: 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. (5.51) for all regions.
8: **end while**

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

$$\left(\boldsymbol{\psi}_g^{in,(l+1)}, \boldsymbol{\phi}_g^{(l+1)}\right) = f\left(\boldsymbol{\psi}_g^{in,(l)}, \bar{\boldsymbol{q}}_g^{(l)}\right) \ , \tag{5.54}$$

In Eq. (5.54), $\boldsymbol{\psi}_g^{in,(l)}$ is a vector of the discrete incoming angular flux boundary conditions in all space and angle for a single group g. $\bar{\boldsymbol{q}}_g^{(l)}$ is a vector of the group sources computed using Eq. (5.52) for all regions, $\boldsymbol{\phi}_g^{(l)}$ is a vector of the scalar fluxes for all regions, and l is the inner iteration index. Although this description assumes a 1-group MOC kernel, a multigroup kernel for solving this fixed source problem is also straightforward and more efficient, as discussed later in this chapter.

In Eq. (5.53), 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*, which is the basis of the *outer iteration*. The general form of the eigenvalue problem in reactor physics can be written in operator notation as

$$\mathbf{T}\boldsymbol{\phi} = \frac{1}{k_{\text{eff}}}\mathbf{F}\boldsymbol{\phi} \ , \tag{5.55}$$

where $\mathbf{F}$ represents the fission operator, and $\mathbf{T}$ represents the streaming-plus-collision operator. The power method to solve Eq. (5.55) consists of the following iterative scheme:

$$\boldsymbol{\phi}^{(\ell+1)} = \mathbf{T}^{-1}\frac{1}{k_{\text{eff}}^{(\ell)}}\mathbf{F}\boldsymbol{\phi}^{(\ell)} \ , \tag{5.56}$$

$$k_{\text{eff}}^{(\ell+1)} = \frac{||\mathbf{F}\boldsymbol{\phi}^{(\boldsymbol{\ell+1})}||_1}{\frac{1}{k_{\text{eff}}^{(\ell)}}||\mathbf{F}\boldsymbol{\phi}^{(\ell)}||_1} \ . \tag{5.57}$$

For the outer iteration, denoted by the index ℓ, the total fission source is computed as shown in Eq. (5.58), and Eq. (5.53) is rewritten as shown in Eq. (5.59).

$$\Psi_i^{(\ell)} = \frac{1}{k_{\text{eff}}^{(\ell)}}\sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}^{(\ell)} \ , \tag{5.58}$$

$$Q_{ext,i,g}^{(\ell)} = \chi_{i,g}^{(\ell)} \Psi_i^{(\ell)} + \sum_{g'=1, g' \neq g}^{G} \Sigma_{s0,i,g' \rightarrow g} \phi_{i,g'}^{(\ell)} \ . \tag{5.59}$$

The equation to update the eigenvalue based on the power method is shown in Eq. (5.60), where V_i is the region volume:

$$k_{\text{eff}}^{(\ell+1)} = \frac{\sum_{i=1}^{I} V_i \sum_{g'=1}^{G} \nu \Sigma_{f,i,g'} \phi_{i,g'}^{(\ell+1)}}{\sum_{i=1}^{I} V_i \Psi_i^{(\ell)}} \ . \tag{5.60}$$

This overall iterative procedure for solving the eigenvalue problem is shown in Algorithm 5.

Algorithm 5: 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. (5.53) for all regions.
4: **for** all groups **do**
5: Compute source for group g by evaluating Eq. (5.52) for all regions.
6: Solve fixed source problem for group g with Algorithm 4.
7: **end for**
8: Update k_{eff} by evaluating Eq. (5.60) for all regions.
9: **end while**

The 1-group iteration scheme presented here has the advantage of reducing memory usage by limiting 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, the inscatter source of Eq. (5.53) is also updated in a Gauss-Seidel fashion. This helps 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 in which the scattering source is updated by a Jacobi iteration, a speed-up of approximately 2× over a 1-group kernel is expected. Additionally, better stability properties for the transport correction used in the isotropic scattering approximation can be observed.

5.5.2 Convergence Criterion for Source Iteration

In MPACT, the solutions 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.

$$\left| k_{\text{eff}}^{(\ell)} - k_{\text{eff}}^{(\ell-1)} \right| < \epsilon_{k_{\text{eff}}} \ . \tag{5.61}$$

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

$$\sqrt{\frac{1}{N_{fi}} \sum_i \left[\sum_g \left(\frac{F_{i,g}^{(l)}}{\bar{F}^{(l)}} - \frac{F_{i,g}^{(l-1)}}{\bar{F}^{(l-1)}} \right) \right]^2} < \epsilon_\phi \ . \tag{5.62}$$

Here, $\epsilon_{k_{\text{eff}}}$ 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 $O(\epsilon_{k_{\text{eff}}})$, and the root-mean-square (RMS) error in ϕ can be expected to be $O(\epsilon_\phi)$.

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 swept sequentially for a bidirectional sweep.

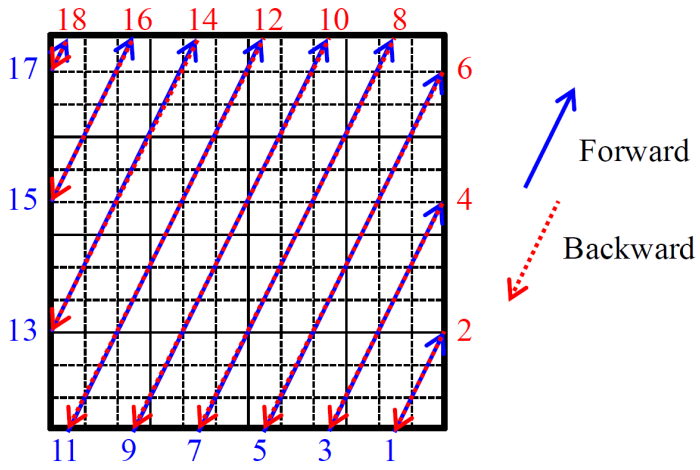


Figure 5.6. Sequential sweep algorithm.

This type of sweep ordering requires increased memory storage for 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 for Multigroup Problems

This section presents the theory of the 2D MOC sweeping algorithms in MPACT for two inscatter-source iteration schemes: Gauss-Seidel and Jacobi. As the MOC solver loops over groups, the Gauss-Seidel approach, uses the flux solution from the previous group to construct the inscatter source for the next group. Alternatively, the Jacobi approach only uses 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 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 6 outlines a more detailed description of the looping structure of a Gauss-Seidel transport sweep and the tasks performed by each 1-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 6: Pseudocode for Gauss-Seidel sweeping with group on outermost loop

```
1:  for each group(g from 1 to N_groups) do
2:      Setup source for group g (using updated flux solution from previous groups)
3:      for each inner iteration (i from 1 to N_inners) do
4:          for each azimuthal angle (a from 1 to N_angles) do
5:              for each longray in angle a (l from 1 to N_longrays(a)) do
6:                  Connect modular rays and determine coarse mesh surfaces
7:                  Calculate exponential values for each segment in longray l and group g
8:                  for each polar angle (p from 1 to N_polar) do
9:                      Gather incoming angular flux (φ^in_{l,p,g}) at both ends of longray l
10:                     for each segment (s from 1 to N_segments(l)) do
11:                         Evaluate forward direction:
12:                             Calculate outgoing angular flux (ψ^out_{l,p,g})
13:                             Tally contribution to fine mesh scalar flux in region r1 (φ_{g,r1})
14:                         Evaluate backward direction:
15:                             Calculate outgoing angular flux (ψ^out_{l,p,g})
16:                             Tally contribution to fine mesh scalar flux in region r2 (φ_{g,r2})
17:                     end for
18:                     Store outgoing angular flux (ψ^out_{l,p,g})
19:                     for each coarse mesh surface intersection (c from 1 to N_cm) do
20:                         Tally surface flux and currents for coarse mesh surface c, group g
21:                     end for
22:                 end for
23:             end for
24:         end for
25:     end for
26: end for
```

Alternatively, the multigroup flux used to construct the inscatter sources can be lagged, creating a Jacobi-like algorithm in energy. This could be implemented in a manner similar to the scheme outlined in Algorithm 6 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 7. With the scheme in Algorithm 7, 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.

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.

Although it is 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 [139]. Results also indicate that the

Algorithm 7: 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
```

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. [145].

CHAPTER 6. 1D AXIAL SOLUTION METHODOLOGY

Previous chapters present 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, whereas 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 [103], 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 are available in the appendices of Stimpson's dissertation [142].

6.1. ONE-DIMENSIONAL AXIAL GOVERNING EQUATIONS

6.1.1 Transport-Based

The axial equations can be derived by averaging the 3D 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):

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

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

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

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

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

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

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

Although 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, although 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 to obtain Eq. (6.3), so that the equations maintain only polar dependence. This formulation was used by Hursin in previous work with DeCART [48, 77]:

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

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

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

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

where

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

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):

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

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):

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

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

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

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

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

6.1.2 Diffusion-Based Solvers

The diffusion-based axial sweepers (NEM/SENM) are formulated by radially averaging the 3D diffusion equation. The details of obtaining the diffusion equation from the transport equation are deferred to Chapter 7 to avoid duplication, but it is important to present the 1D diffusion equation to allow for better understanding the resulting extensions.

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

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

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):

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

This is numerically equivalent to Eq. (6.7c).

6.2. ONE-NODE VS. TWO-NODE

As previously mentioned, MPACT defines a single axial mesh region of a single rod (fuel, guide tube, control, or burnable poison rod) as a node. 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, reference to a "left" or "right" node, as shown in the figures, directly relates 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 distributions are output.

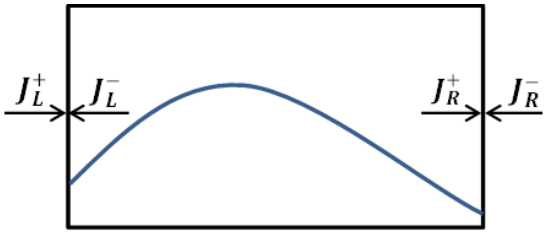


Figure 6.1. One-node kernel.

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.

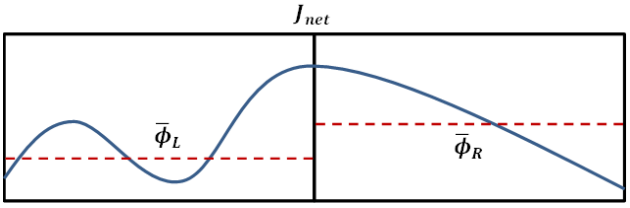


Figure 6.2. Two-node kernel.

It is beneficial that the two-node kernels do not allow the mesh-average scalar flux (zeroth moment) to change because it is considerably more stable. When incorporating the transverse leakage terms, which is discussed in more detail below, negative sources may be encountered 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 stored only 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, then mesh-averaged angular fluxes would need to be preserved, which would be prohibitively expensive to store for the existing 2D/1D scheme. However, it is possible that a two-node transport kernel could be used for other related methods, such as the 2D/3D scheme, in which 2D-MOC and 3D-S_N are coupled [163].

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 [144]. This capability has demonstrated some usefulness, but it has been shown 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 [80]. 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 [55], 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 Eq. 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 nodes being simulated, as in Eq. (6.13). Here, 1 and 2 indicate the indices of the two nodes:

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

Although more explicit details of how this matrix is formed can be found in Stimpson's dissertation [142], 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, because 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:

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

where *alpha* 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.

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 [162]:

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

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

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

In the previous set of equations, the flux has a homogeneous solution, $A\sinh(\kappa\xi) + B\cosh(\kappa\xi)$, as well as 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 that used for 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) [114]:

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

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:

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

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 or SENM kernels by normalizing spatially (ξ) and rearranging the boundary conditions slightly. With this setup, P_N handles the angular distribution, and NEM or SENM 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

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

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

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

Second-Moment

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

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

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

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 two-node diffusion-based 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 and can then be used to inform the incoming boundary conditions for the surrounding nodes. 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 is provided. Historically, P_N methods within the context of 2D/1D have been referred to as *simplified* P_N, or SP_N [79, 80]. 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, the terminology could be misleading because the 2D/1D approximation does not preserve the 3D SP_N approximation when using these axial solvers. To avoid further confusion, these methods are now being referred to as P_N; when referring to the combined angle and space methods, the terms NEM-P_N or SENM-P_N are used.

6.6. DISCRETE ORDINATES (S_N)

Principles similar 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, the azimuthal dependence can be represented in several ways. 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 described by Stimpson [142] (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 $f_{out,i}$, $f_{j,in}$, and $f_{j,i}$ 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 gain a better understanding of what these condensed 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

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

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

Complete descriptions of the condensed functions up through a cubic source are available in Stimpson's dissertation [142].

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:

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

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

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

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

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

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, by 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:

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

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

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

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 propagate 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.

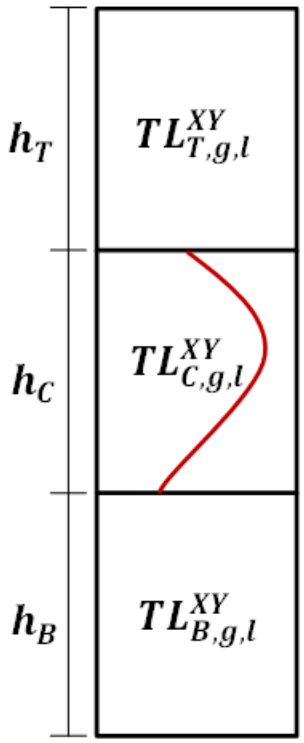


Figure 6.3. Transverse leakage interpolation.

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

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

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

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

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

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

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.

Cho and Joo have shown that the quadratic interpolation can introduce larger errors in cases with more severe axial profiles [22], particularly when using a refined axial mesh. It is likely that higher order expansions would alleviate these errors, but the results from Cho and Joo 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 from Cho and Joo and those shown here, which can likely avoid these issues.

CHAPTER 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 [135]. The fundamental concept of CMFD also applies to the transport equation and has been shown to be effective at accelerating the convergence of transport problems. In general, a linearized form of CMFD is algebraically equivalent to a coarse-mesh diffusion synthetic acceleration (DSA) scheme. [100]

The 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 14.8 discusses the performance of CMFD for 3D problems using 2D/1D and 3D MOC. In Section 7.6, CMFD is discussed using red-black successive over-relaxation. Section 14.5 describes subplane CMFD, and Section 7.5 presents the convergence criteria used in MPACT to determine when to discontinue performing power iterations.

7.1. THE CMFD METHOD

The lower order equations used in the CMFD acceleration scheme are derived by integrating the multigroup neutron transport equation (2.19) over all angles. This yields the multigroup neutron balance equation:

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

CMFD also introduces the concept of a coarse mesh on which the low-order equations are solved compared to the fine mesh of the transport solution. Integrating Eq. (7.1) over a coarse cell j gives the discretized neutron balance equation

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

where 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. The integration over a coarse cell also defines the restriction (homogenization) operator from the fine mesh solution onto the coarse mesh and is shown below:

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

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

$$J^{\text{net}}_{j,g,s} = \frac{\sum_{s'\in s} J^{\text{net}}_{i,g,'}A_{i,s'}}{\sum_{s'\in s} A_{i,s'}} , \tag{7.3c}$$

$$\chi_{j,g} = \frac{\sum_{i\in j} \chi_{i,g}\nu\Sigma_{f,i,g}\phi_{i,g}V_i}{\sum_{i\in j} \nu\Sigma_{f,i,g}\phi_{i,g}V_i} . \tag{7.3d}$$

Here the index i denotes the fine cells of the transport mesh.

In classic diffusion theory, the net current is approximated by Fick's Law:

$$J^{\text{net,diff}}_{g} = -D_{j,g,s}\boldsymbol{\nabla}\cdot\phi_g , \tag{7.4}$$

where the diffusion coefficient is defined via the transport-corrected cross section, as follows:

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

Σ_{tr} is defined in Section 5.3.3. Applying a finite-difference spatial discretization to Fick's Law on the coarse mesh gives Eqs. (7.6), where $h_{j,s}$ is the distance between the center of cell j and its neighboring cell on surface s:

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

$$\tilde{D}_{j,g,s} = \frac{2D_{j,g}D_{j_s,g}}{h_{j,s}\left(D_{j,g} + D_{j_s,g}\right)} . \tag{7.6b}$$

Here, the subscript j_s is the index of the cell bordering cell j via surface s. This approximation yields a set of low-order equations that are not consistent with the transport equations, so CMFD introduces a correction coefficient, $\hat{D}_{j,g,s}$, to the expression for the net current:

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

The correction factor, $\hat{D}_{j,g,s}$, is defined by Eq. (7.8), where the flux, $\phi^{trans}_{j,g}$, and the net current $J^{\text{net},trans}_{j,g,s}$ are determined by the homogenized fine mesh transport solution as defined in Eqs. (7.3b) and (7.3c), respectively:

$$\hat{D}_{j,g,s} = \frac{J^{\text{net},trans}_{j,g,s} + \tilde{D}_{j,g,s}\left(\phi^{trans}_{j,g} - \phi^{trans}_{j,g,s}\right)}{\left(\phi^{trans}_{j,g} + \phi^{trans}_{j,g,s}\right)} . \tag{7.8}$$

This correction factor, along with cross section homogenization, creates equivalence between the solution of the fine mesh transport equations and the coarse mesh diffusion equations, Eqs. (7.2) and (7.7), at convergence. The homogenization process preserves all the cell-volume integrated reaction rates based on the fine

mesh solution. The correction factor of Eqs. (7.7) and (7.8) 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.

Since low-order CMFD equations are solved on a coarser mesh than the transport mesh, it also requires development of prolongation operators to transfer the solution between the CMFD coarse mesh and the transport fine mesh regions. Because the CMFD problem is an eigenvalue problem, the prolongation equation for the scalar flux takes a nonlinear form given by

$$\phi_{i,g}^{(l+1)} = \phi_{i,g}^{(l+\frac{1}{2})} \frac{\phi_{j,g}^{(l+1)}}{\phi_{j,g}^{(l+\frac{1}{2})}} \ , \ i \in j \ . \tag{7.9}$$

This projection preserves the coarse mesh cell-averaged flux and reaction rates from the CMFD solution and preserves the intra-cell flux shape calculated by the transport calculation. The iterative solution algorithm with CMFD is described in Algorithm 8, in which the superscript l denotes the iteration index. Given a

Algorithm 8: Iterative algorithm for the CMFD acceleration of steady-state eigenvalue problem

while not converged **do**
  Compute cell-averaged values for CMFD coefficients from Eqs. (7.3) and (7.7):
  $\left\{\phi_{j,g}^{(l+\frac{1}{2})}, \Sigma_{x,j,g}, \chi_{j,g}, \tilde{D}_{j,g,s}, \hat{D}_{j,g,s}\right\} \leftarrow f_{hom}\left(\phi_{i,g}^{(l+\frac{1}{2})}, \Sigma_{x,i,g}, \chi_{i,g}\right)$
  Solve CMFD eigenvalue problem given by Eqs. (7.2) for cell-averaged scalar flux and k_{eff}:
  $\phi_{j,g}^{(l+1)} \leftarrow f_{\text{CMFD}}\left(\phi_{j,g}^{(l+\frac{1}{2})}, \Sigma_{x,j,g}, \chi_{j,g}, \tilde{D}_{j,g,s}, \hat{D}_{j,g,s}\right)$
  Update fine mesh solution given by Eq. (7.9) and (7.10):
  $\boldsymbol{\phi}_g^{(l+1)} \leftarrow f_{\text{pro}}\left(\boldsymbol{\phi}_g^{(l+1)}, \boldsymbol{\phi}_g^{(l+\frac{1}{2})}\right)$
  Perform transport sweep:
  $\left\{\boldsymbol{\phi}_g^{(l+\frac{1}{2})}, \boldsymbol{\varphi}_g^{in,(l+\frac{1}{2})}\right\} \leftarrow f_{\text{transport}}\left(\boldsymbol{\varphi}_g^{in,(l)}, \boldsymbol{\phi}_g^{(l)}\right)$
  Update fine-mesh fission source
  Check if solution is converged
end while

flat initial guess to the transport solution, the CMFD correction coefficient becomes zero so the CMFD problems effectively become a diffusion problem for the first iteration. The eigenvalue problem is solved in the low-order CMFD equations to reduce the computational cost of the eigenvalue iterations.

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 [6–8, 97] 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. Spatial domain decomposed (SDD)-CMFD extends the prolongation operator to also provide an update to the angular flux boundary condition on the decomposed spatial subdomains [143]. This update equation can take a number of forms, so it is written as

$$\varphi^{in,(l+1)}_{i,g,m,k} = \varphi^{in,(l)}_{i,g,m,k} f^{(l+1)}_{j,g,s} \;, \quad k \in s, \quad i \in j \;. \tag{7.10}$$

In MPACT, the form of f is taken to be

$$f^{(l+1)}_{j,g,s} = \frac{\phi^{S,(l+1)}_{j,g,s}}{\phi^{S,\left(l+\frac{1}{2}\right)}_{j,g,s}} \;. \tag{7.11}$$

Here, $\phi^{S}_{j,g,s}$ is the surface-averaged scalar flux for surface s. Previous work [143] compares several definitions of f, showing that Eq. (7.11) provides reasonable additional speed-up and guarantees positivity of the update factor. To obtain the coarse mesh surface fluxes required by Eq. (7.11), additional radial coupling coefficients must be introduced to reconstruct the transport surface flux from the coarse mesh cell-averaged flux. These are given below and can be derived in a manner similar to that used for the current coupling coefficients:

$$\phi^{S}_{j,g,s} = \tilde{s}_{j,g,s}\phi_{j,g} - \left(1 - \tilde{s}_{j,g,s}\right)\phi_{j,g,s} + \hat{s}_{j,g,s}\left(\phi_{j,g} + \phi_{j,g,s}\right) \;, \tag{7.12a}$$

$$\tilde{s}_{j,g,s} = \frac{1}{1 + h_{j,s}\frac{\Sigma_{tr,j,g}}{\Sigma_{tr,j,g,s}}} \;, \tag{7.12b}$$

$$\hat{s}_{j,g,s} = \frac{\phi^{S,transport}_{j,g,s} - \tilde{s}_{j,g,s}\phi_{j,g} + \left(1 - \tilde{s}_{j,g,s}\right)\phi_{j,g,s}}{\left(\phi_{j,g} + \phi_{j,g,s}\right)} \;. \tag{7.12c}$$

Here, Σ_{tr} is the transport-corrected total cross section defined in Section 5.3.3. It is expected that SDD-CMFD will have performance similar to that of 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 $\tilde{D}$:

$$\tilde{D}_{g,s} = \frac{2}{3\left(\Sigma_{tr,g,s^-}h_{s^-} + \Sigma_{tr,g,s^+}h_{s^+}\right)} + \frac{1}{4} \;, \tag{7.13}$$

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 as the result of 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 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 [165]. This is done by adding a variable amount to the diffusion coefficient on a surface, based on the optical thickness of its neighboring cells:

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

The variable θ in Eq. (7.14) 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 (7.14) demonstrates the relationship between adCMFD and pCMFD. The actual implementation of adCMFD is slightly different:

$$\tilde{D}_{g,s} = \frac{2}{3\left(\frac{\Sigma_{tr,g,s-}h_{s-}}{1+3\theta_{g,s-}} + \frac{\Sigma_{tr,g,s+}h_{s+}}{1+3\theta_{g,s+}}\right)} \, , \tag{7.15}$$

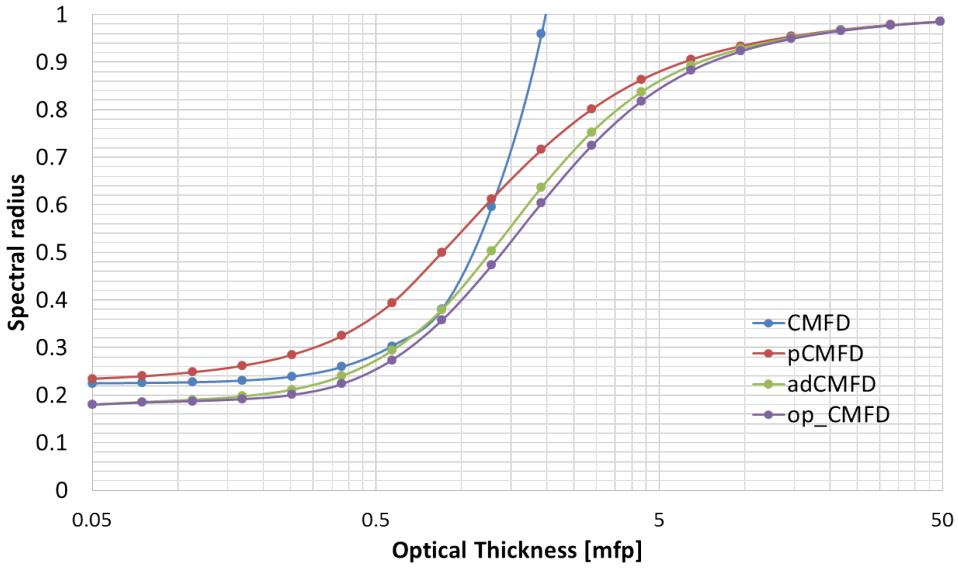


Figure 7.1. Fourier analysis convergence behavior of CMFD-accelerated transport method for CMFD variants (1D IHM S_N, 1 transport sweep per CMFD update, 3 fine cells per coarse cell).

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:

$$\theta_g = \begin{cases} 0 \ , & \Sigma_{tr,g}h < 1 \ , \\ \sum_{k=0}^{6} a_k \left(\Sigma_{tr,g}h\right)^{k+1} \ , & 1 \leq \Sigma_{tr,g}h \leq 14 \ , \\ 0.254\Sigma_{tr,g}h \ , & \Sigma_{tr,g}h > 14 \ . \end{cases} \tag{7.16}$$

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

$$\Lambda_{j,g} = \frac{\phi_{j,g}^{l+1}}{\phi_{j,g}^{l+\frac{1}{2}}} \ . \tag{7.17}$$

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

$$\Lambda_{i,g} = f_i \Lambda_{j,g}, i \in j \ , \tag{7.18a}$$

$$\frac{1}{V_j} \sum_{i \in j} f_{i,g} V_i = 1 \ . \tag{7.18b}$$

The curve in Fig. 7.1 was generated by running many different cases while varying f for a 1D S_N problem, with three 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. SOLVING THE CMFD EIGENVALUE PROBLEM

7.5.1 Power Iteration

Algorithm 8 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. (7.2) can be described as follows:

$$\left[-\boldsymbol{\nabla}\cdot D_g(\boldsymbol{x})\boldsymbol{\nabla}+\Sigma_{t,g}(\boldsymbol{x})\right]\phi_g^{(l+1)}(\boldsymbol{x})-\sum_{g'=1}^{G}\Sigma_{s0,g'\rightarrow g}(\boldsymbol{x})\phi_{g'}^{(l+1)}(\boldsymbol{x})=\frac{\chi_g}{k_{\text{eff}}^{(l)}}F^{(l)}(\boldsymbol{x})\,, \tag{7.19a}$$

$$F^{(l+1)}(\boldsymbol{x})=\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{x})\phi_{g'}^{(l+1)}(\boldsymbol{x})\,, \tag{7.19b}$$

$$k_{\text{eff}}^{(l+1)}=k_{\text{eff}}^{(l)}\frac{\int F^{(l+1)}(\boldsymbol{x})F^{(l+1)}(\boldsymbol{x})d^3\boldsymbol{x}}{\int F^{(l+1)}(\boldsymbol{x})F^{(l)}(\boldsymbol{x})d^3\boldsymbol{x}}\,. \tag{7.19c}$$

The domain of integration for the integrals in Eq. (7.19c) 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. (7.19) can be rewritten as follows:

$$M\underline{\phi}^{(l+1)}=\frac{1}{k_{\text{eff}}^{(l)}}F\underline{\phi}^{(l)}\,, \tag{7.20a}$$

$$k_{\text{eff}}^{(l+1)}=k_{\text{eff}}^{(l)}\frac{\left\langle F\underline{\phi}^{(l+1)},F\underline{\phi}^{(l+1)}\right\rangle}{\left\langle F\underline{\phi}^{(l+1)},F\underline{\phi}^{(l)}\right\rangle}\,. \tag{7.20b}$$

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 non-negative scalar flux).

7.5.2 Wielandt Shift

7.5.2.1 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 [152]. 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. (7.19) is modified as follows:

$$\begin{aligned}[-\boldsymbol{\nabla}\cdot D_g(\boldsymbol{x})\boldsymbol{\nabla}+\Sigma_{t,g}(\boldsymbol{x})]\phi_g^{(l+1)}(\boldsymbol{x})-\sum_{g'=1}^{G}\left[\Sigma_{s0,g'\rightarrow g}(\boldsymbol{x})+\frac{\chi_g(\boldsymbol{x})}{k_s^{(l)}}\nu\Sigma_{f,g'}(\boldsymbol{x})\right]\phi_{g'}^{(l+1)}(\boldsymbol{x})&\\ =\left[\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}}\right]\chi_g F^{(l)}(\boldsymbol{x})\,,&\end{aligned} \tag{7.21a}$$

$$F^{(l+1)}(\boldsymbol{x})=\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{x})\phi_{g'}^{(l+1)}(\boldsymbol{x})\,, \tag{7.21b}$$

$$k_{\text{eff}}^{(l+1)}=\left[\left(\frac{1}{k_{\text{eff}}^{(l)}}-\frac{1}{k_s^{(l)}}\right)\frac{\int F^{(l+1)}(\boldsymbol{x})F^{(l)}(\boldsymbol{x})d^3\boldsymbol{x}}{\int F^{(l+1)}(\boldsymbol{x})F^{(l+1)}(\boldsymbol{x})d^3\boldsymbol{x}}+\frac{1}{k_s^{(l)}}\right]^{-1}\,. \tag{7.21c}$$

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 $k_s \leq k_{\text{eff}}$, then 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 [37], an iteration-dependent or adaptive Wielandt shift is employed, defined as

$$\frac{1}{k_{s,\text{PARCS}}^{(l)}} \equiv \max\left\{\frac{1}{k_{\text{eff}}^{(l)}} - c_1\left|\frac{1}{k_{\text{eff}}^{(l)}} - \frac{1}{k_{\text{eff}}^{(l-1)}}\right| - c_0, \frac{1}{k_{\max}}\right\} . \tag{7.22}$$

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, because doing so would yield a singular system in Eq. (7.19a). 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 remainder 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.

7.5.2.2 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. (7.21) are modified as follows:

$$\left[-\boldsymbol{\nabla}\cdot D_g(\boldsymbol{x})\boldsymbol{\nabla} + \Sigma_{t,g}(\boldsymbol{x})\right]\phi_g^{(l+1)}(\boldsymbol{x}) - \sum_{g'=1}^{G}\left[\Sigma_{s0,g'\to g}(\boldsymbol{x}) + \frac{1}{k_s^{(l)}(\boldsymbol{x})}\nu\Sigma_{f,g'}(\boldsymbol{x})\right]\phi_{g'}^{(l+1)}(\boldsymbol{x}) = \left[\frac{1}{k_{\text{eff}}^{(l)}} - \frac{1}{k_s^{(l)}(\boldsymbol{x})}\right]\chi_g F^{(l)}(\boldsymbol{x}) , \tag{7.23a}$$

$$F^{(l+1)}(\boldsymbol{x}) = \sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{x})\phi_{g'}^{(l+1)}(\boldsymbol{x}) , \tag{7.23b}$$

$$k_{\text{eff}}^{(l+1)} = \frac{\int F^{(l+1)}(\boldsymbol{x})F^{(l+1)}(\boldsymbol{x})d^3\boldsymbol{x}}{\int F^{(l+1)}(\boldsymbol{x})\left[\left(\frac{1}{k_{\text{eff}}^{(l)}} - \frac{1}{k_s^{(l)}(\boldsymbol{x})}\right)F^{(l)}(\boldsymbol{x}) + \frac{1}{k_s^{(l)}(\boldsymbol{x})}F^{(l+1)}(\boldsymbol{x})\right]d^3\boldsymbol{x}} . \tag{7.23c}$$

This reduces to Eqs. (7.21) when $k_s^{(l)}(\boldsymbol{x})$ 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:

$$k^{(l)}_{s,\mathrm{LE}}(\boldsymbol{x}) \equiv k_\infty(\boldsymbol{x}) \,, \tag{7.24}$$

$$k^{(l)}_{s,\mathrm{LEPS}}(\boldsymbol{x}) \equiv \max\{k^{(l)}_{s,\mathrm{LE}}(\boldsymbol{x}), k^{(l)}_{\mathrm{eff}}\} \,, \tag{7.25}$$

$$k^{(l)}_{s,\mathrm{ILEPS}}(\boldsymbol{x}) \equiv \min\{k^{(l)}_{s,\mathrm{LEPS}}(\boldsymbol{x}), k^{(l)}_{s,\mathrm{PARCS}}\} \,. \tag{7.26}$$

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

$$\Sigma_{t,g}(\boldsymbol{x})\phi_{0,g}(\boldsymbol{x}) - \sum_{g'=1}^{G} \Sigma_{s0,g'\to g}(\boldsymbol{x})\phi_{0,g'}(\boldsymbol{x}) = \frac{\chi_g}{k_\infty(\boldsymbol{x})} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\boldsymbol{x})\phi_{0,g'}(\boldsymbol{x}) \,. \tag{7.27}$$

In matrix notation, this can be expressed as

$$\left[\Sigma_t(\boldsymbol{x}) - \Sigma_{s0}(\boldsymbol{x})\right] \underline{\phi_0}(\boldsymbol{x}) = \frac{1}{k_\infty(\boldsymbol{x})} \underline{\chi} \left[\underline{\nu\Sigma_f}(\boldsymbol{x})\right]^T \underline{\phi}_0(\boldsymbol{x}) \,. \tag{7.28}$$

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

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

The LE shift provides a physically motivated shift which may have better performance than 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. (7.23a) 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. (7.25), bounds the value of $k_s(\boldsymbol{x})$ 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. Whereas 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. (7.26) 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. [158].

7.5.2.3 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. (7.23a). 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) be 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.

7.5.3 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

$$M\underline{\phi} = \frac{1}{k_{\text{eff}}} F\underline{\phi} \,, \tag{7.30}$$

then the second condition can be expressed as follows:

$$\frac{\|r^{(l)}\|}{\|r^{(0)}\|} \equiv \frac{\left\| M\underline{\phi}^{(l)} - \frac{1}{k_{\text{eff}}^{(l)}} F\underline{\phi}^{(l)} \right\| / \left\| \underline{\phi}^{(l)} \right\|}{\left\| M\underline{\phi}^{(0)} - \frac{1}{k_{\text{eff}}^{(0)}} F\underline{\phi}^{(0)} \right\| / \left\| \underline{\phi}^{(0)} \right\|} < 100 \,, \tag{7.31}$$

$$\left| k_{\text{eff}}^{(l)} - k_{\text{eff}}^{(l-1)} \right| < 10^{-6} \,. \tag{7.32}$$

7.6. RED-BLACK SUCCESSIVE OVER-RELAXATION CMFD

7.6.1 Red-Black Gauss-Seidel

The linear systems in CMFD can be solved by a variety of techniques. A PETSc-based [9] 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 [35], which tags each coarse cell as red or black to produce a checkerboard pattern, as in Fig. 7.2 [138]. This shows the layout for a small 2×2 core of assemblies with 3×3 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

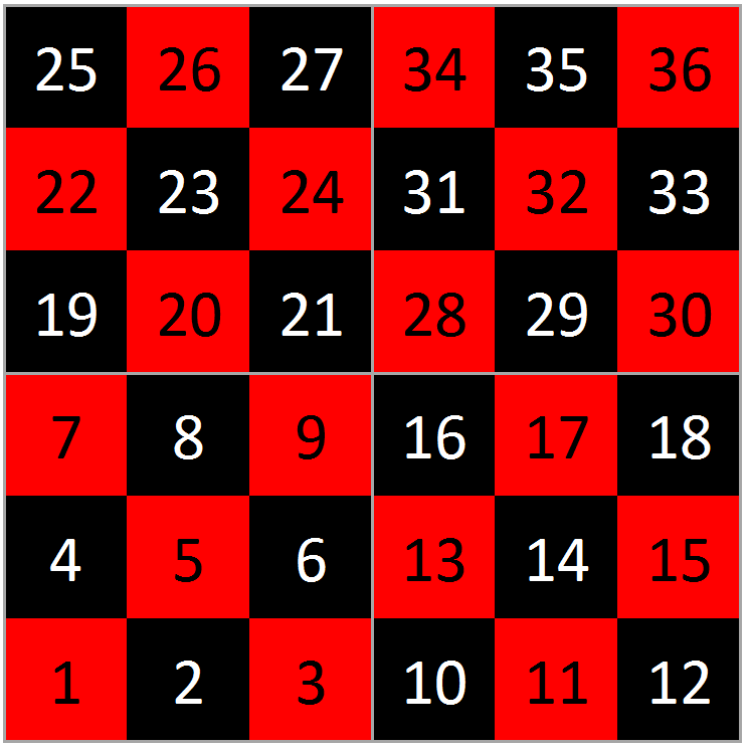


Figure 7.2. Red-black indexing.

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 [9], although a hybrid MPI/OpenMP approach is being considered by the PETSc development team.

7.6.2 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 [35, 151],

$$\phi_{n,g}^{(l)} = \omega_{R/B}^{(l-1)} \phi_{n,g}^{(l-\frac{1}{2})} + (1 - \omega_{R/B}^{(l-1)}) \phi_{n,g}^{(l-1)} \,, \tag{7.33}$$

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 [61, 151], in which the red and black fluxes use different relaxation factors—Eqs. (7.34) and (7.35)—that eventually converge to the same value. Equation (7.34) shows the initial relaxation factors, and Eq. (7.35) shows the relaxation factor for subsequent iterations:

$$\omega_R^{(0)} = 1 \,, \tag{7.34a}$$

$$\omega_B^{(0)} = \frac{1}{1 - \frac{1}{2}\rho_J^2} \,, \tag{7.34b}$$

$$\omega_R^{(l+1)} = \frac{1}{1 - \frac{1}{4}\rho_J^2 \omega_B^{(l)}} \,, \quad l \geq 1 \,, \tag{7.35a}$$

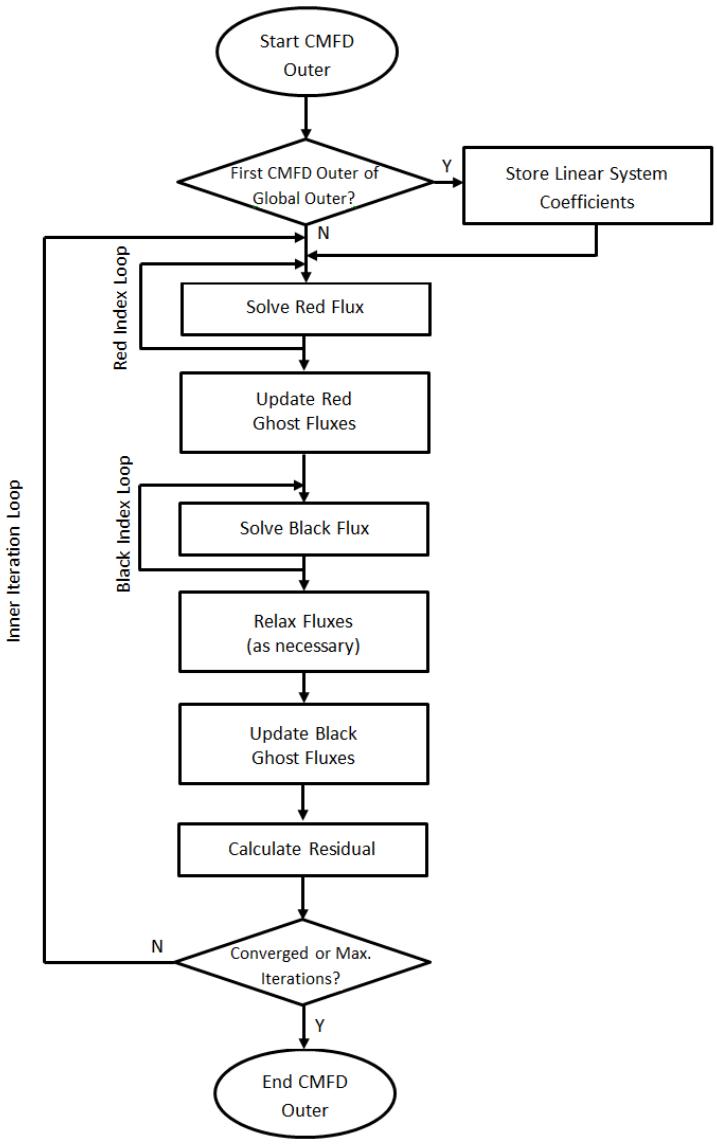


Figure 7.3. Red-black SOR CMFD iteration strategy.

$$\omega_B^{(l+1)} = \frac{1}{1 - \frac{1}{4}\rho_J^2 \omega_R^{(l+1)}} \ , \quad l \geq 1 \ . \tag{7.35b}$$

These equations make use of the Jacobi spectral radius, which in this work is estimated during the first 10 inner iterations without relaxation. The pseudo-error (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 reduction of roughly 2× in CMFD solve time is observed, with a reduction of roughly 40% overall.

7.7. 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 14.5. Finally, the power iteration scheme for solving the CMFD eigenvalue problem is described in Section 7.5. 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 7.6.

CHAPTER 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: that is, the *macroscopic cross sections*. In Section 2.2, the steady-state Multigroup (MG) transport equation has been derived,

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

Unlike nodal diffusion codes, in which 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 calculations are discussed. 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 are 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 [90]. 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 [89]. Validations for the MPACT cross section library and transport solvers are presented in Ref. [36].

The native MPACT cross section library format, which is based on the HELIOS [50] and DeCART library [87] 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 intermediate resonance (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 caused by 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 the report by Downar et al. [90].

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. Because the absorption (and fission) cross sections are modified through the resonance treatment and are needed for the depletion calculation, these cross sections should also be included. 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
- heat released from each fission κ
- scattering cross sections $\sigma_{s,l,g}$
- (n,2n) cross section $\sigma_{n\to 2n,g}$
- (n,3n) cross section $\sigma_{n\to 3n,g}$
- P_{0-3} scattering matrices $\sigma_{s,l,g'\to g}$
- 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 [90] 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, the scattering matrices are expanded by Legendre polynomials, and the angular fluxes in the scattering source term are expanded by spherical harmonics [70]:

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

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

If Eqs. (8.5) and (8.6) are inserted into the scattering source term, then the spherical harmonic addition theorem can be applied. When this is performed, the resultant scattering source term is shown in Eq. (8.7):

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

In this equation, $R_{l,m}$ are the real components of the spherical harmonics. The microscopic cross sections $\sigma_{s,l,g'\to g,iso}$ 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 realistic calculations when using low-order scattering.

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

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

Consistently, the total cross section in Eq. (8.1) is 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 is discussed in Chapter 11, but the time-dependent MG neutron transport equation and the precursor equation are given here to inform the discussion on data needs for transient calculation:

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

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

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 to define 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 [36].

The MPACT library provides the delayed group decay constants $\lambda_{\tau,iso}$ for every fissionable isotope. However, another common approximation to save memory is to neglect 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 of a concern for the 8-group delayed data [39], because all the isotopes have the same set of decay constants.

8.4. ON-THE-FLY ENERGY CONDENSATION

8.4.1 MPACT Iteration Scheme

MPACT uses the method of characteristics (MOC), a geometry-agnostic method useful for resolving the complex heterogeneity of tens of thousands of fuel rods in light-water reactors (LWRs), to solve the neutron transport equation. This method is applied in the radial direction and is coupled to a faster running, lower-order transport method in the axial direction—an approach called 2D/1D. The 2D/1D iteration scheme is then accelerated by the coarse mesh finite difference (CMFD) acceleration method. The MOC calculations are the most expensive part of the calculation because they require sweeping the length of a very large number of rays drawn across the geometry. The CMFD calculations are close behind because they involve solving a nonlinear system of equations; the number of rows in the matrix for this system is equal to the number of coarse mesh cells multiplied by the number of energy groups. The MPACT iteration scheme is illustrated in Figure 8.1.

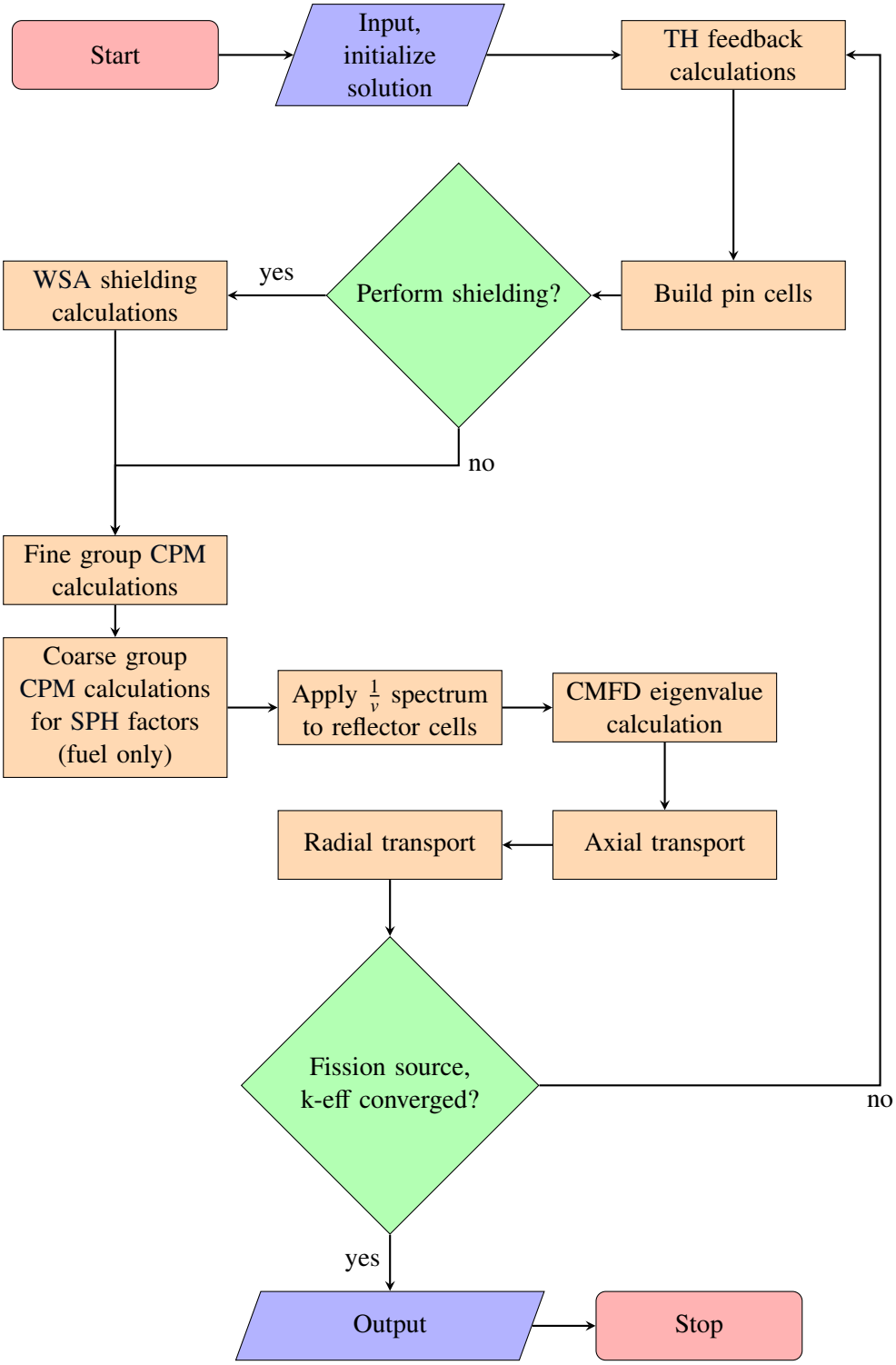


Figure 8.1. Illustration of MPACT iteration scheme.

8.4.2 Condensation Procedure

To accurately condense cross sections, it is important that an accurate energy spectrum be used. Because this spectrum is not known before solving the problem, it must be accurately approximated. To do so, the 3D reactor is decomposed into constituent pin cells that can be solved efficiently. A fine group spectrum is approximated for each region of each pin cell and is used to collapse the cross sections for each region. These collapsed cross sections are then used for the whole-core transport calculations.

8.4.3 Pin Cell Identification

Several categories of pin cells can be identified:

- fuel pins,
- gadolinia-bearing (gad) fuel pins,
- guide tubes,
- control rods,
- discrete burnable poison inserts,
- control blades,
- cells comprising the channel box, bypass flow, and control blade in boiling water reactors (BWRs), and
- everything else (including axial and radial reflector cells).

Cells are grouped into these categories because some of them require special treatments. These treatments are discussed in the sections in which they are applicable.

8.4.4 Pin Cell Spectrum Calculation

For the pin cell calculations, fine group cross sections are calculated using the results of the Wigner-Seitz approximation (WSA) shielding calculations, discussed in Section 9.6. These cross sections are used to set up a 1D collision probabilities method (CPM) calculation. The CPM calculations are one-group calculations, so the groups must be swept starting with the fastest group and ending with the most thermal group. Additionally, the presence of up-scatter and more than one region in the pin cell requires iterating over all the groups to resolve the up-scattering source and the spatial distribution of the fission source. For each iteration over the groups, the total fission source in the pin cell is fixed to a constant value. This ensures that the calculation eventually converges even though the pin cell will likely not be perfectly critical. Note that the spectrum calculation should use the original radius of the pin cell that preserves volume, not the Dancoff-equivalent radius. The results of the WSA calculations are incorporated into the fine group cross sections, so there is no need to use the Dancoff-equivalent radius for the spectrum calculation too.

8.4.5 Cross Section Generation

Once the fine group spectrum ϕ_g is obtained for each region, the spectrum and the cross sections can be collapsed:

$$\phi_G = \sum_{g \in G} \phi_g \,, \tag{8.17a}$$

$$\Sigma_{x,G} = \frac{\sum_{g \in G} \phi_g \Sigma_{x,g}}{\phi_G} \,, \tag{8.17b}$$

$$\Sigma_{s,G \to G'} = \sum_{g' \in G'} \sum_{g \in G} \frac{\phi_g \Sigma_{s,g \to g'}}{\phi_G} \,, \tag{8.17c}$$

$$\chi_G = \sum_{g \in G} \chi_g \,, \tag{8.17d}$$

where G indicates a coarse group quantity and $g \in G$ refers to all fine groups g that are inside the coarse group G.

8.4.6 SPH Factors

Energy condensation always results in some error in the collapsed cross sections if angle-dependent cross sections are not used [16, 117]. To mitigate these effects, SuPerHomogéneísation (SPH) factors are used. To calculate these factors, a second CPM calculation is set up for each pin cell, this time using the coarse group structure and cross sections. The converged source from the fine group CPM calculation is condensed to the coarse group structure and used to drive the coarse group CPM calculation. After the calculation is converged, an SPH factor is calculated:

$$f_{i,G}^k = \frac{\phi_{fine,i,G}}{\phi_{coarse,i,G}^k} \,, \tag{8.18}$$

where $\phi_{fine,i,G}$ is the condensed flux from the fine group CPM calculation in region i and coarse group G and $\phi_{coarse,i,G}^k$ is the flux from the coarse group CPM calculation. The cross sections are updated using this factor:

$$\Sigma_{x,i,G}^k = f_{i,G}^k \Sigma_{x,i,G}^0 \,, \tag{8.19}$$

where $\Sigma_{x,i,G}^0$ is calculated with Eqs. (8.17b) and (8.17c) and $f_{i,G}^0 = 1$.

8.4.7 Geometry Considerations

The 1D cylindrical CPM calculations fit naturally with LWR fuel rods. However, there are several regions that do not fit well in this type of calculation. First, the baffle and radial reflector regions are not well suited for cylindrical calculations. For this reason, a $\frac{1}{\nu}$ spectrum is used in those regions instead of CPM calculations. The cylindrical CPM method was tested for those regions but gave worse results than a simple predefined spectrum. Several other more complicated hard-coded spectra were attempted but did not make a significant difference.

Second, the channel box and bypass regions of BWRs do not fit well in cylindrical geometry. These regions are still solved using the cylindrical CPM calculations, but this represents an area for improvement.

Third, the axial reflector regions do not necessarily fit well into the cylindrical approach. It is expected that this is less important than the radial reflectors. This also represents an area for improvement.

CHAPTER 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 transport 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,

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

It is not feasible to accurately determine the neutron spectra of a specific problem before the 3D whole-core transport calculations are performed, 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 [154] and RMET21 [105] 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 [137], different methods can be derived to determine the equivalent cross sections for consideration of spatial self-shielding effects. The Bondarenko background cross section method [13] is the conventional method, incorporating Dancoff factors to account for the spatial self-shielding. The subgroup method [34] 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 [76, 155]. Compared to the conventional Bondarenko method, which 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 [110], 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 calculations based on multigroup libraries with subgroup parameters such as the HELIOS library [50]. The MPACT production library is processed with the Evaluated Nuclear Data File (ENDF)-BVII.1 cross sections using SCALE-6.2 [14], as well as a few auxiliary codes for generating subgroup parameters, resonance

upscattering data, transport cross section data for hydrogen, and superhomogenization (SPH) factors for ^{238}U.

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 [105],

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

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,

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

In Eq. (9.2), the neutron energy E has been transformed to lethargy u, as is conventionally done for slowing-down equations. 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 $u-\epsilon_i$ to u, the intermediate resonance (IR) approximation [63] is employed to achieve

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

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:

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

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

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

The equivalence theory [50] correlates the flux of a homogeneous material with a heterogeneous material by introducing the equivalence cross section Σ_e,

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

where $\Sigma_b = \sum\limits_i \lambda_i \Sigma_{p,i} + \Sigma_e$ 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, and 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 caused by 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 [50]:

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

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,

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

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 $\sigma_{x,g,0}$ [50].

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),

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

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. 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 the flux is first converted into an equivalence cross section. This conversion and the detailed subgroup equations are discussed in Section 9.5.

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

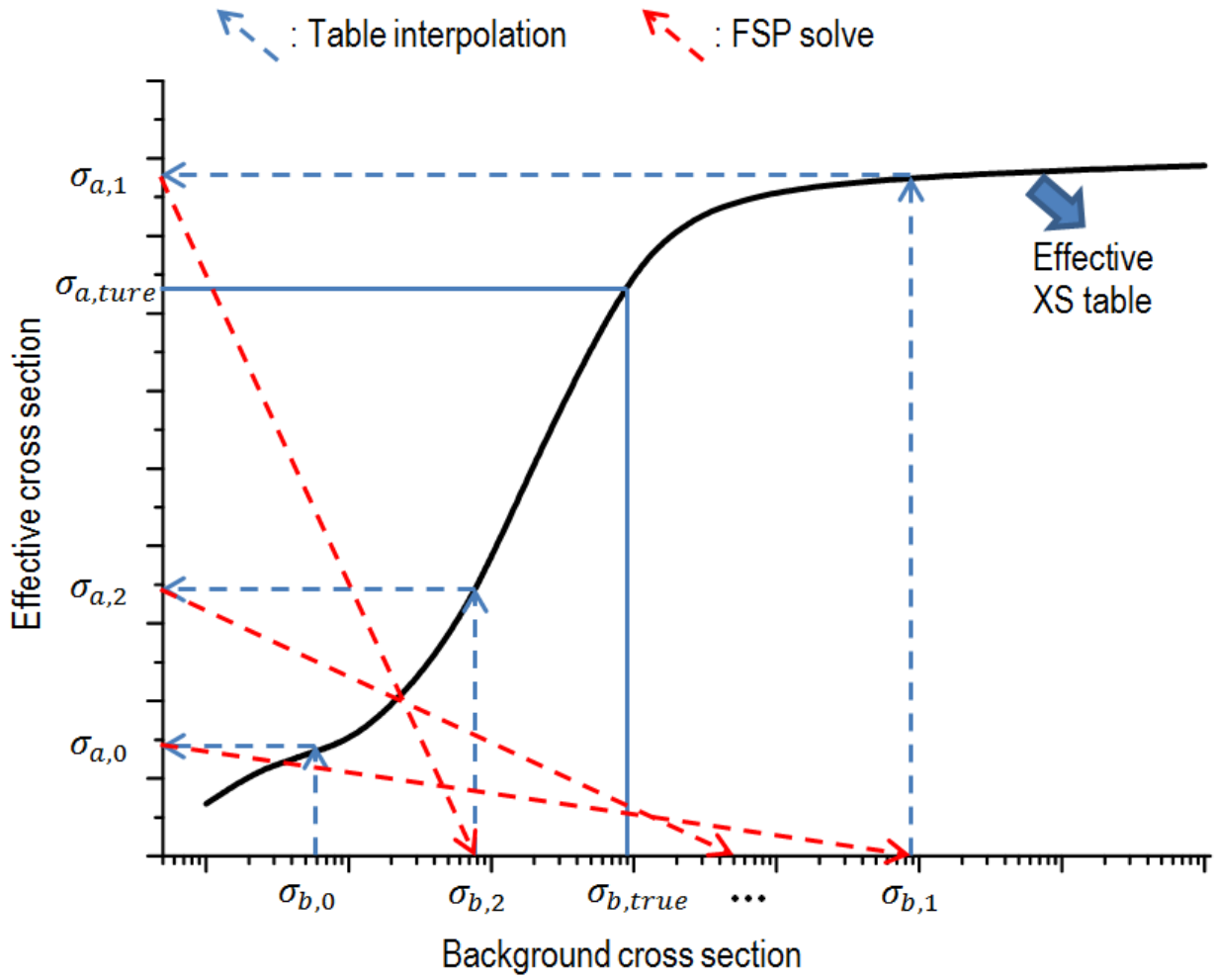


Figure 9.1. Conceptual illustration of ESSM procedures.

calculations, and (2) the computational time of the subgroup method is predetermined by the number of subgroup levels used in the fixed source calculations, whereas 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 [110] 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 0D form, but 1D 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 descriptions of these terms can be found in the paper by Liu et al. [110].

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 considered afterward on the multi-group level using Bondarenko iteration described in WIMS code [5]. In MPACT, Bondarenko iteration is also used for the subgroup method and ESSM. More accurate models for resonance interference require on-the-fly slowing-down calculations [106], which are not implemented into the subgroup method and ESSM because of 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 2D 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

$$f_{\mathrm{iso},g} = \frac{\sigma^{\mathrm{shielded}}_{s,\mathrm{iso},g}}{\sigma^{\mathrm{unshielded}}_{s,\mathrm{iso},g}}\ . \tag{9.13}$$

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

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

The corrections for high-order scattering terms are performed using the same ratios that are used 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. As can be seen, inside these loops, there is an iteration loop in which transport sweeps are performed for each resonant group, category, and level. 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: Pseudocode for pre-existing subgroup scheme with group on outermost loop.

```
for each resonant group (g from g_res,beg to g_res,end)
    for each subgroup category (c from 1 to N_cat(g))
        for each subgroup level (l from 1 to N_levels)
            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)
                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 is 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 [88] 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 10 shows the pseudocode for the refactored scheme, taking advantage of the multigroup kernel concept.

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 11 shows the pseudocode for the batched approach, where each batch contains a starting and stopping pseudogroup index:

In addition to using multigroup kernels, a lumped parameter MOC approach has been applied to the subgroup self-shielding problem [141]. Because the self-shielding calculation is a purely absorbing fixed source

Algorithm 10: Pseudocode for subgroup scheme using the multigroup transport kernel.

for each pseudogroup (pg from 1 to $N_{pseudogroups}$)
    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})
    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 $N_{pseudogroups}$)
    Calculate equivalence cross section $\Sigma_{eq,pg}$
end for

Algorithm 11: Pseudocode for subgroup scheme using the multigroup transport kernel and batching.

for each batch (b from 1 to N_{batch})
    for each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$)
        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})
        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 $pg_{end}(b)$)
        Calculate equivalence cross section $\Sigma_{eq,pg}$
    end for
end for

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 and backward) at the same time, effectively, two equations are needed:

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

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

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.

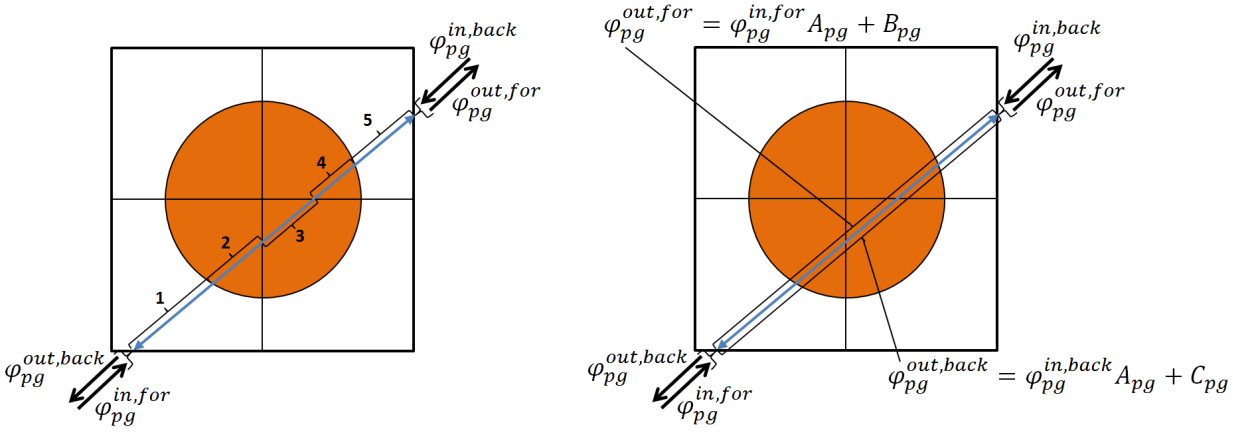


Figure 9.2. Visualization of MOC ray tracing (left) and lumped parameter (right) on a pin cell.

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 12 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 12: Pseudocode for lumped parameter subgroup scheme using the multigroup transport kernel and batching.

1: **for** each batch (b from 1 to N_{batch})
2: **for** each pseudogroup (pg from $pg_{beg}(b)$ to $pg_{end}(b)$)
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 an initial sweep accumulating the $A_{pg}/B_{pg}/C_{pg}$ lumped parameters
7: **for** each iteration (i from 2 to N_{iters})
8: Perform a transport sweep applying $A_{pg}/B_{pg}/C_{pg}$ parameters for this batch
9: Compare residual based on boundary angular fluxes (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 $pg_{end}(b)$)
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, because 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. The concept of the 1-group/multigroup in this section should be interpreted as the energy integration of the 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),

$$\Omega \cdot \nabla \varphi_{g,c,n}(r,\Omega) + \left[\Sigma_{a,g,c,n}(r) + \lambda_g \Sigma_p(r)\right] \varphi_{g,c,n}(r,\Omega) = \frac{1}{4\pi} \lambda_g \Sigma_p(r) \Delta u_g \ . \tag{9.17}$$

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

$$\Sigma_{a,g,c,n} = \frac{\sum\limits_{i \in c} N^i I^i_{a,g,\infty}(T_{loc})}{I^r_{a,g,\infty}(T_{loc})} \sigma^r_{a,g,n}(T_{loc}) \ ,$$

and

$$\lambda_g \Sigma_p = \sum_i \lambda^i_g \Sigma^i_p \ . \tag{9.18}$$

$I^i_{a,g,\infty}(T_{loc})$ 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 $\sigma_{a,g}(T_{ave})$ [112], 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:

$$\phi_{g,c,n} = \frac{\Sigma_{b,g,c,n} \Delta u_g}{\Sigma_{b,g,c,n} + \Sigma_{a,g,c,n}} = \frac{\left(\sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq,g,c,n}\right) \Delta u_g}{\left(\sum_i \lambda_i \Sigma_{p,i} + \Sigma_{eq,g,c,n}\right) + \Sigma_{a,g,c,n}} \ . \tag{9.20}$$

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:

$$\sigma^{i,\mathrm{arg}}_{a,g,m} = \frac{I^r_{a,g,\infty}(T_{loc})}{I^i_{a,g,\infty}(T_{loc})} f(T_{loc}) \bar{\sigma}^i_{a,g,m} \,. \tag{9.21}$$

Here, $f(T_{loc})$ is defined as the same as Eq. (9.19), but it is about isotope i. Using $\sigma^{i,\mathrm{arg}}_{a,g,m}$ as the argument to obtain $\Sigma_{eq,g,c,m}(\sigma^{i,\mathrm{arg}}_{a,g,m})$, the background cross section can be computed as

$$\Sigma^i_{b,g,m} = \Sigma_{eq,g,c,m}(\sigma^{i,\mathrm{arg}}_{a,g,m}) + \lambda_g \Sigma_p \,. \tag{9.22}$$

With all the subgroup parameters, the effective cross section is computed as

$$\sigma^i_{a,g} = \frac{\sum\limits_m \bar{\sigma}^i_{a,g,m} \frac{\Sigma^i_{b,g,m}}{\Sigma^i_{a,g,m}(T_{loc})+\Sigma_{x,g}+\Sigma^i_{b,g,m}} w^i_{a,g,m}(T_{loc})}{\sum\limits_m \frac{\Sigma^i_{b,g,m}}{\Sigma^i_{a,g,m}(T_{loc})+\Sigma_{x,g}+\Sigma^i_{b,g,m}} w^i_{a,g,m}(T_{loc})} \,, \tag{9.23}$$

where $\Sigma^i_{a,g,m}(T_{loc}) = N^i f(T_{loc}) \bar{\sigma}^i_{a,g,n}$ and $\Sigma_{x,g} = \sum\limits_{j \neq i} \Sigma^j_{a,g}$. 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

$$\Omega \cdot \nabla \varphi_{c,n}(r,\Omega) + [\Sigma_{a,c,n}(r) + \lambda \Sigma_p(r)] \varphi_{c,n}(r,\Omega) = \frac{1}{4\pi} \lambda \Sigma_p(r) \Delta u \,. \tag{9.24}$$

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. Specifically, 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.

9.6. WIGNER-SEITZ APPROXIMATION FOR SELF-SHIELDING CALCULATIONS

MPACT relies on the subgroup approach to shielding calculations discussed in Section 9.2. Historically, it performed subgroup calculations for each category of isotopes in each resonance group. Recently, the WSA approximation was implemented to accelerate the subgroup calculations [91]. In this approach, the core is decomposed into pin cells. For each pin cell category, a one-group transport calculation is performed to obtain a Dancoff factor for each pin cell and each subgroup isotope category. The CPM is then used on each pin cell to determine a Dancoff factor for the isolated pin cell. The outer radius of the pin cell is then adjusted and the CPM calculation repeats until the isolated pin cell Dancoff factor equals that obtained from the whole-core calculation. Once this iteration scheme is complete, the new Dancoff-equivalent cell can be used to accurately calculate the spatial and energy self-shielding effects for each region in the pin cell.

This WSA approach has two major benefits. First, it is more computationally efficient than performing whole-core shielding calculations. The 51-group library has 22 resonance groups that have subgroup levels and isotope categories for each resonance group. For a typical pressurized water reactor (PWR) calculation, this results in 268 "pseudogroups" that must be solved using MOC. For the WSA approach, this number is reduced to fewer than 20 for most problems. When using the 252-group library, there are 133 resonance groups that lead to more than 1,000 pseudogroups to solve. The CPM calculations require nontrivial amounts of time, but the overall shielding calculation is still significantly faster than the whole-core approach since the number of MOC calculations is no longer a function of the number of resonance groups. Second, the constituent pin cells required for the on-the-fly (OTF) energy condensation, discussed in Section 8.4, are already constructed for the WSA approach to shielding, allowing them to be reused to generate an energy spectrum as well.

CHAPTER 10. NUCLIDE DEPLETION AND DECAY

The composition of fuels varies during irradiation in a nuclear reactor core, which is referred to as *depletion*. In the simulation of depletion, there are two main computational components: (i) the point depletion solver and (ii) the time integration algorithm. A good summary of the conventional methods for the two components can be found in Chapter 9 of the Handbook of Nuclear Engineering [93]. With regards to the point depletion solver, MPACT can call the ORIGEN code [60], which is included in the SCALE package [131], or it can use an 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 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:

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{N} f_{j\rightarrow i}\lambda_j X_j(t) + \bar{\phi}\sum_{k=1}^{N} g_{k\rightarrow i}\sigma_k X_k(t) - (\lambda_i + \sigma_i\bar{\phi})X_i(t)\ , \quad 1 \le i \le N\ , \tag{10.1}$$

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 phenomena 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,

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_i, \ldots, X_N)^T \ ,$$

then Eq. (10.1) can be written in matrix form as

$$\frac{d\boldsymbol{X}(t)}{dt} = \boldsymbol{A} \cdot \boldsymbol{X}(t) \ , \tag{10.2}$$

where $\boldsymbol{A}$ 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

$$\boldsymbol{X}(t) = \exp(\boldsymbol{A}t)\boldsymbol{X}(0) \ , \tag{10.3}$$

where the vector $\boldsymbol{X}(0)$ 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.

$$\boldsymbol{X}(t_n + \Delta t) = \exp(\boldsymbol{A}\Delta t)\boldsymbol{X}(t_n) \ . \tag{10.4}$$

Obtaining $\boldsymbol{X}(t_n + \Delta t)$ then becomes a matter of calculating $\exp(\boldsymbol{A}\Delta t)$.

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 [115]. 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 many 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:

$$\exp(\boldsymbol{A}\Delta t) = I + \boldsymbol{A}\Delta t + \frac{(\boldsymbol{A}\Delta t)^2}{2!} + \cdots = \sum_{m=0}^{\infty} \frac{(\boldsymbol{A}\Delta t)^m}{m!} \ . \tag{10.5}$$

However, if all the nuclides are included in the transition matrix $\boldsymbol{A}$, then $\boldsymbol{A}$ 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 $\boldsymbol{A}$.

10.1.2.1 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, as in

$$\Delta t \leq 10 t_{r,1/2} \quad \text{where} \quad t_{r,1/2} = \left(\frac{\ln 2}{\lambda_i + \sigma_i \bar{\phi}} \right) . \tag{10.6}$$

Given that the transition matrix only considers long-lived nuclides, the Ball and Adams [10] suggestion that the transitions involving short-lived nuclides with large removal rates be considered instantaneous is adopted. For a decay chain of $A \rightarrow B \rightarrow C$ and a small removal half-life for B, 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, which has been removed. Similarly, if the removal half-life for A is very small, then 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 the process of removing these short-lived nuclides to create a *reduced transition matrix* (or preconditioned transition matrix), special care must be taken to ensure that 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}, then 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 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.

10.1.2.2 Calculation of the Long-lived Nuclide Number Densities

As a result of splitting up and preconditioning the 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

$$\exp(\tilde{\boldsymbol{A}}\Delta t) = I + \tilde{\boldsymbol{A}}\Delta t + \frac{(\tilde{\boldsymbol{A}}\Delta t)^2}{2!} + \cdots = \sum_{m=0}^{\infty}\frac{(\tilde{\boldsymbol{A}}\Delta t)^m}{m!}. \tag{10.9}$$

However, even with the removal of the short-lived nuclides from the transition matrix $\boldsymbol{A}$, it is still quite large and sparse, and it is computationally inefficient to store the reduced transition matrix, $\tilde{\boldsymbol{A}}$, 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 $\tilde{\boldsymbol{A}}$, 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, so 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 [99] is

used:

$$\|\tilde{\boldsymbol{A}}\| = \min\left[\max_{1\le j\le N}\left(\sum_{i=1}^{N}|\tilde{a}_{i,j}|\right).\max_{1\le i\le N}\left(\sum_{j=1}^{N}|\tilde{a}_{i,j}|\right)\right] . \tag{10.12}$$

10.1.2.3 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. Because secular equilibrium is assumed, the system of differential equations of the short-lived nuclides can be written as 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, because 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 describes how the matrix exponential is formulated and calculated. In the description above, the overall procedure is mentioned, but it is 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 [131].

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.

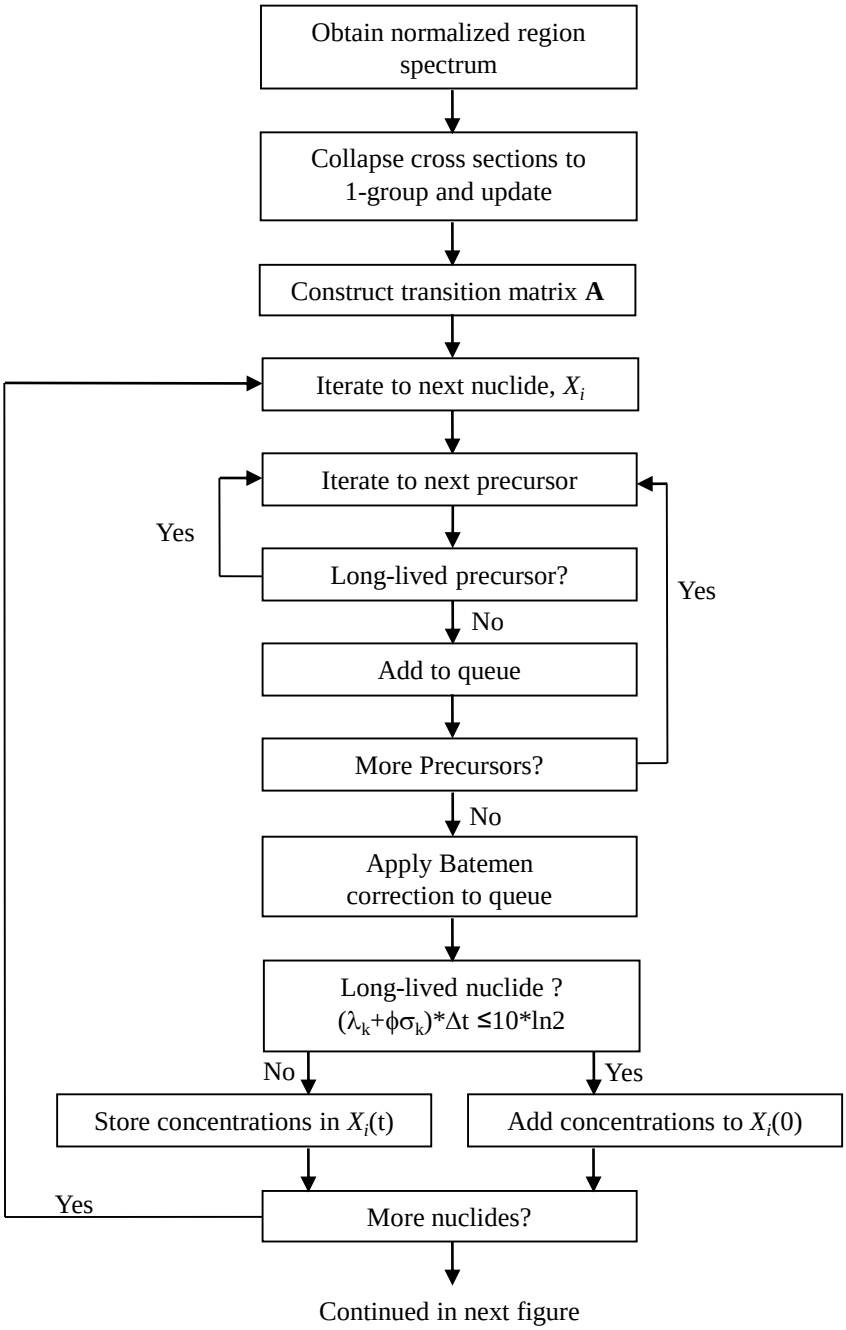


Figure 10.1. Point depletion algorithm (1 of 2).

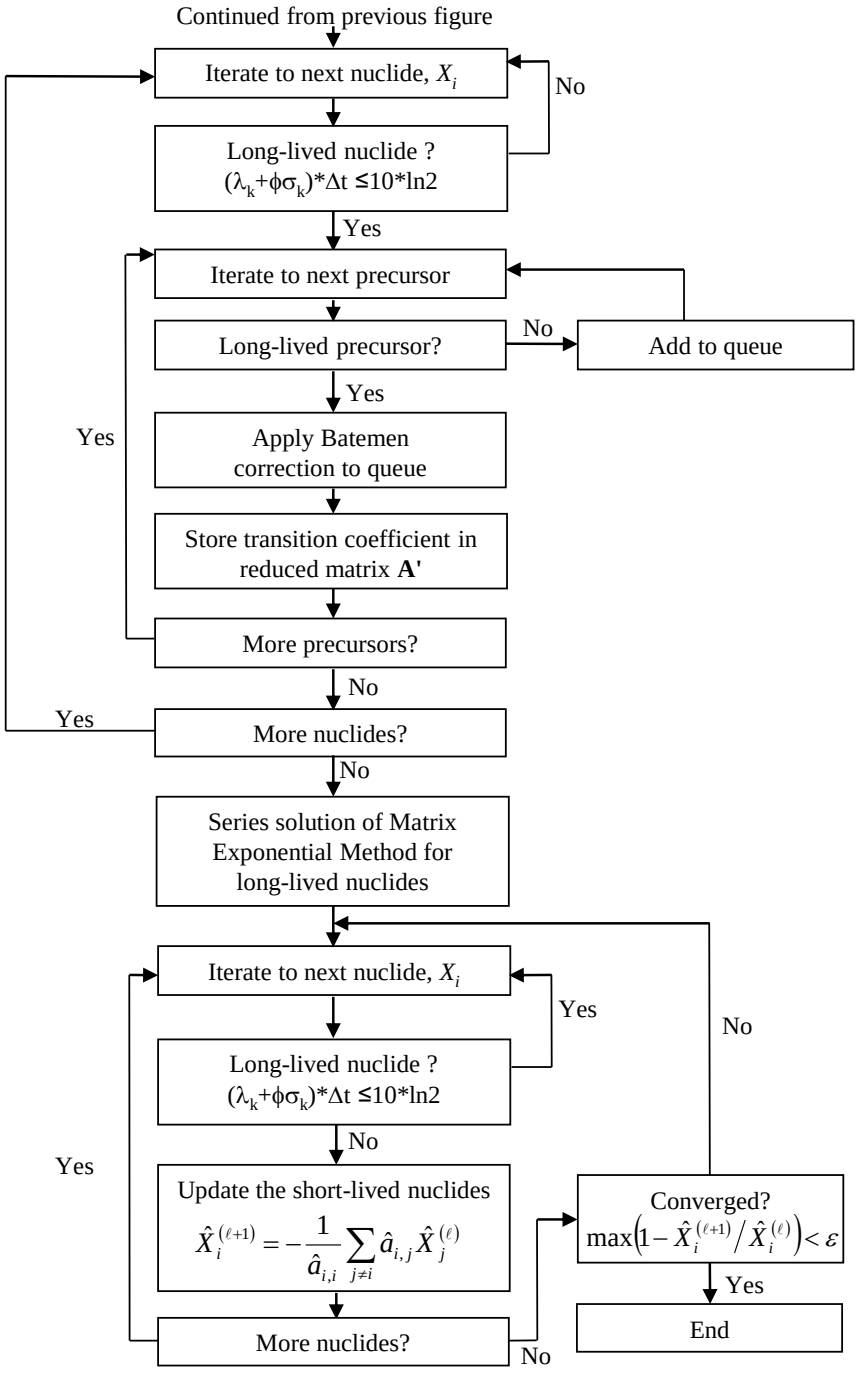
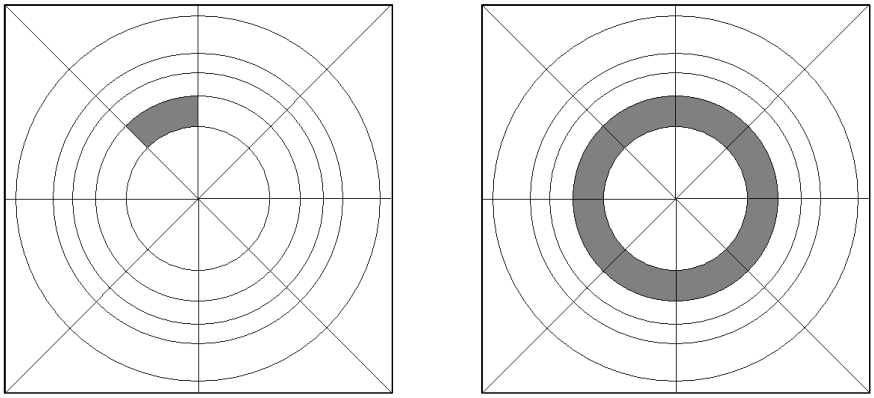


Figure 10.2. Point depletion algorithm (2 of 2).

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 only requires a simple modification, but it will considerably increase memory requirements.



Flat Scalar Flux Region (grey)

Uniform Cross Section Region (grey)

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 undesirable longer computation time. Therefore, to reduce computation time and allow for longer burnup step sizes, MPACT adopts two commonly used techniques: (i) the predictor-corrector [137] and (ii) sub-step methods [93]. 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 in Figure 10.4 and is given in Eq. (10.15):

$$X_{t2} = \frac{X^P_{t2}(\phi_{t1}, \sigma_{t1}) + X^C_{t2}(\phi^P_{t2}, \sigma^P_{t2})}{2} . \tag{10.15}$$

The predictor step uses 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^p_{t2}) and cross sections (σ^p_{t2}) are obtained through a transport calculation using the predicted concentration, $X^P_{t2}(\phi_{t1}, \sigma_{t1})$. Next, the corrector step performs a depletion calculation using the new 1-group flux (ϕ^p_{t2}) and cross section (σ^p_{t2}), and the new corrected particle number densities, $X^C_{t2}(\phi^P_{t2}, \sigma^P_{t2})$, are obtained. The final particle number densities for t_2 are then taken to be the arithmetic mean of the

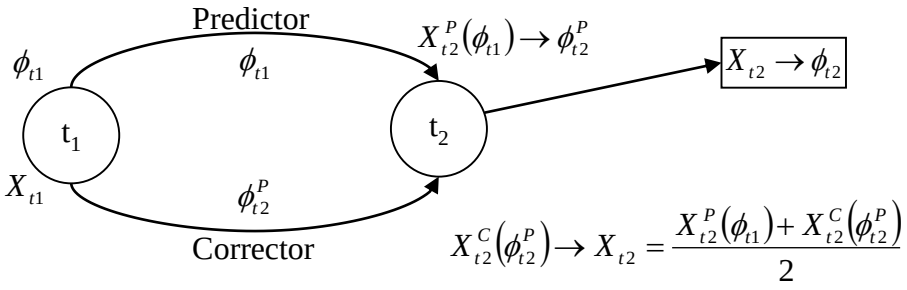


Figure 10.4. Illustration of depletion predictor-corrector.

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

$$\phi_m = \phi_{t1} f_{m-1} \quad \text{where} \quad f_{m-1} = \frac{P_{t1}}{\sum_j \sum_i X^{i,j}_{m-1} \kappa\sigma^{f,i,j}_{t1} \phi^j_{t1}}, \tag{10.16}$$

where P_{t1} is the total power at t_1, ϕ^j_{t1} and $\kappa\sigma^{f,i,j}_{t1}$ 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 $X^{i,j}_{m-1}$ 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.

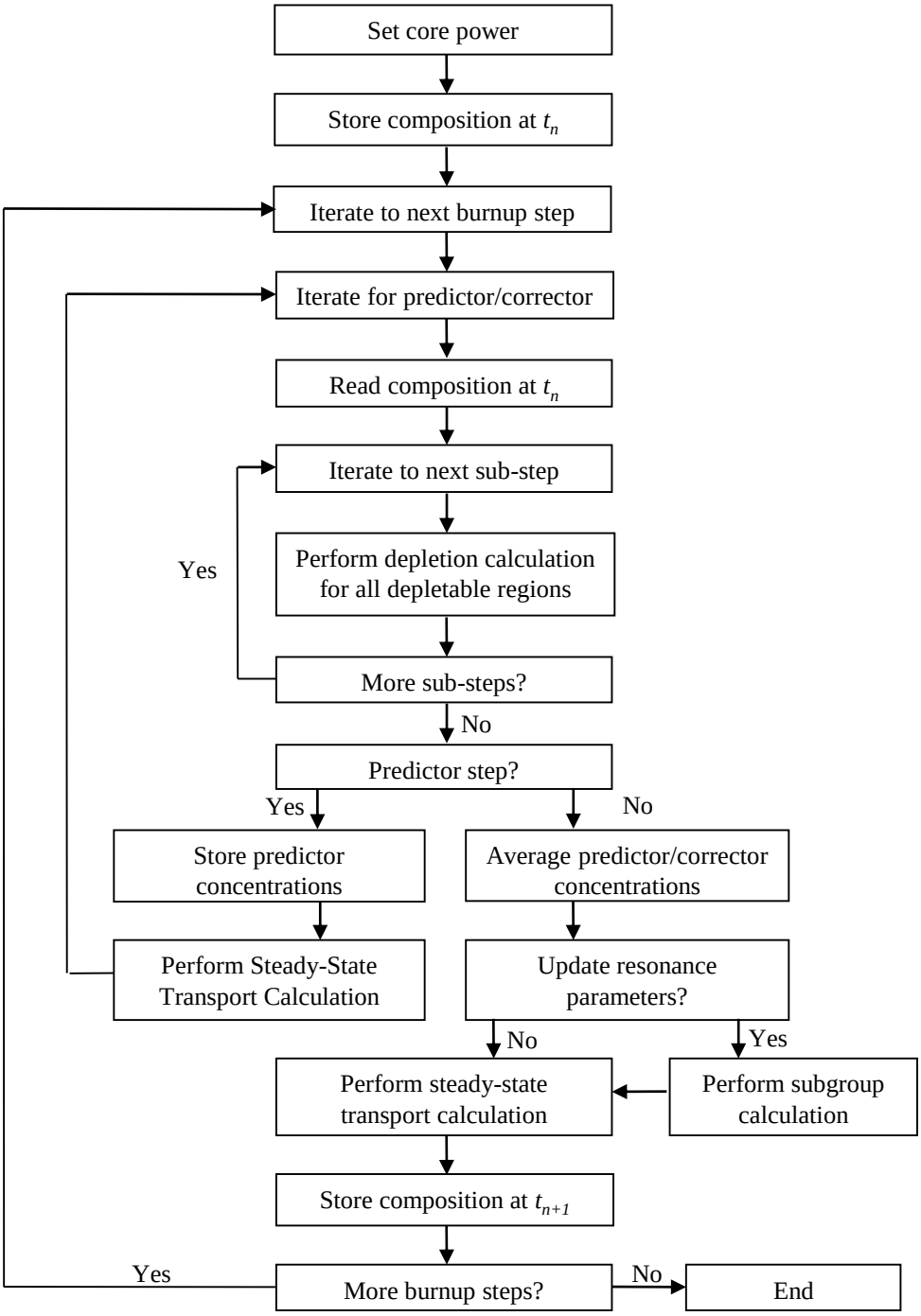


Figure 10.5. MPACT depletion algorithm.

10.3. XENON AND SAMARIUM TREATMENTS

Certain isotopes that are highly important to accurately predicting reactivity of the reactor are also tightly coupled with thermal hydraulic (TH) feedback and therefore require special treatments that are embedded in the MPACT iteration scheme for a single state point. The most important example of these is ^{135}Xe, but

^{149}Sm is also treated similarly. The Bateman equations for ^{135}Xe can be written as follows:

$$\frac{dN_I}{dt} = \gamma_I \Sigma_f(t)\phi(t) - \lambda_I N_I(t) \ , \tag{10.17a}$$

$$\frac{dN_{Xe}}{dt} = \gamma_{Xe}\Sigma_f(t)\phi(t) + \lambda_I N_I(t) - \lambda_{Xe}N_{Xe}(t) - \sigma_{a,Xe}(t)\phi(t)N_{Xe}(t) \ , \tag{10.17b}$$

where subscripts I and Xe represent ^{135}I and ^{135}Xe, N is the number density, γ is the direct yield from fission, Σ_f is the macroscopic fission cross section, σ_a is the microscopic absorption cross section, ϕ is the scalar flux, and λ is the decay constant. Note that the yields in this case are cumulative yields, meaning the direct yield of the nuclide from fission plus the indirect yield of the nuclide via radioactive decay of other fission products. For ^{149}Sm, the equations are the same except that (1) Xe is replaced by Sm, (2) I is replaced by Pm for ^{149}Pm, (3) $\gamma_{Sm} = 0$, and (4) $\lambda_{Sm} = 0$. Treatments for these isotopes are split into two categories: steady state (or pseudo-steady state during depletion calculations) and transient.

10.3.1 Steady State

Several treatments can be applied to these equations for either ^{135}Xe ^{149}Pm: *zero*, *dep*, *equil*, and *peak*. The *zero* treatment does what the name implies: it sets the isotope number densities to 0 regardless of any depletion that might have been previously calculated. The *dep* option means that no special treatment is applied, allowing the depletion solver to populate the isotope number densities with whatever method it uses for the calculation. The other two treatments are discussed in more detail in the following.

10.3.1.1 Equilibrium

The *equil* option is used to calculate equilibrium isotope concentrations. This is important because ^{135}Xe is tightly coupled to the power shape and has a half-life much shorter than a typical depletion step, so the shape of the ^{135}Xe concentration must be iterated alongside the neutron transport and thermal hydraulics calculations. Thus, the depletion solve is usually insufficient for treating the ^{135}Xe calculation. This discussion speaks exclusively in terms of ^{135}I and ^{135}Xe, but the same procedure can be applied to ^{149}Pm and ^{149}Sm.

To begin, assume that the depletion time step is much greater than the isotope half-lives, allowing equilibrium to be assumed. This gives steady-state equations:

$$\frac{dN_I}{dt} = 0 = \gamma_I \Sigma_f(t)\phi(t) - \lambda_I N_I(t) \ , \tag{10.18a}$$

$$\frac{dN_{Xe}}{dt} = 0 = \gamma_{Xe}\Sigma_f(t)\phi(t) + \lambda_I N_I(t) - \lambda_{Xe}N_{Xe}(t) - \sigma_{a,Xe}(t)\phi(t)N_{Xe}(t) \ . \tag{10.18b}$$

The first equation can be rearranged to directly solve for the ^{135}I number density:

$$N_I(t) = \frac{\gamma_I \Sigma_f(t)\phi(t)}{\lambda_I} \ . \tag{10.19}$$

Substituting that result into the second equation gives a solution for the ^{135}Xe number densities:

$$\begin{aligned} &\gamma_{Xe}\Sigma_f\phi(t) + \lambda_I\left(\frac{\gamma_I\Sigma_f(t)\phi(t)}{\lambda_I}\right) - \lambda_{Xe}N_{Xe}(t) - \sigma_{a,Xe}(t)\phi(t)N_{Xe}(t) = 0 \\ \rightarrow &N_{Xe}(t)\left(\lambda_{Xe} + \sigma_{a,Xe}(t)\phi(t)\right) = \gamma_{Xe}\Sigma_f\phi(t) + \gamma_I\Sigma_f(t)\phi(t) \\ \rightarrow &N_{Xe}(t) = \frac{\gamma_{Xe}\Sigma_f\phi(t) + \gamma_I\Sigma_f(t)\phi(t)}{\lambda_{Xe} + \sigma_{a,Xe}(t)\phi(t)} \ . \end{aligned} \tag{10.20}$$

This calculation is performed independently for every region using the flux and cross sections for that region. The update is then performed every coupled iteration so that the effects of TH feedback are properly incorporated into the equilibrium calculations.

10.3.1.2 Peak

The other special treatment is *peak*. This treatment calculates the maximum possible amount of ^{149}Sm that could build up based on the current reactor conditions. This treatment makes sense only because ^{149}Sm is stable; this treatment is not allowed for ^{135}Xe because it will completely decay away as it is produced by ^{135}I. This treatment is quite simple: if $\tilde{N}_{Pm}$ and $\tilde{N}_{Sm}$ are the number densities calculated by the depletion solve for ^{149}Pm and ^{149}Sm, respectively, then the number densities calculated by this treatment for the neutron transport calculations is as follows:

$$N_{Pm}(t) = 0\,, \tag{10.21a}$$

$$N_{Sm}(t) = \tilde{N}_{Pm}(t) + \tilde{N}_{Sm}(t)\ . \tag{10.21b}$$

This represents the condition at $t + \infty$ with no scalar flux, at which point all the ^{149}Pm will have decayed to ^{149}Sm without any loss of ^{149}Sm along the way.

10.3.2 Transient

First, the iodine equation can be solved using an integrating factor:

$$\begin{aligned}
\frac{dN_I}{dt} + \lambda_I N_I(t) &= \gamma_I \Sigma_F \phi(t) \\
\frac{d}{dt}\left[e^{\lambda_I t} N_I(t)\right] &= e^{\lambda_I t}\gamma_I \Sigma_F \phi(t) \\
\int_{t_1}^{t_2}(\cdot) \Rightarrow e^{\lambda_I t_2} N_I(t_2) - e^{\lambda_I t_1} N_I(t_1) &= \gamma_I \Sigma_F \int_{t_1}^{t_2} e^{\lambda_I t'} \phi(t')\,dt' \\
e^{\lambda_I t_2} N_I(t_2) &= e^{\lambda_I t_1} N_I(t_1) + \gamma_I \Sigma_f \int_{t_1}^{t_2} e^{\lambda_I t'} \phi(t')\,dt' \\
N_I(t_2) &= e^{\lambda_I (t_1 - t_2)} N_I(t_1) + \gamma_I \Sigma_f e^{-\lambda_I t_2} \int_{t_1}^{t_2} e^{\lambda_I t'} \phi(t')\,dt'\ .
\end{aligned}$$

It can be assumed that the flux is separable in space and time over short time steps:

$$\phi(t) \approx \overline{\phi} p(t)\ . \tag{10.22}$$

Substituting this expression into the equation for $I(t_2)$ gives the final solution for $N_I(t_2)$:

$$N_I(t_2) = e^{\lambda_I (t_1 - t_2)} N_I(t_1) + \gamma_I \Sigma_f \overline{\phi} e^{-\lambda_I t_2} \int_{t_1}^{t_2} e^{\lambda_I t'} p(t')\,dt'\ . \tag{10.23}$$

Everything in this expression is known except for the integral of the scalar flux shape function $p(t)$. This shape function is effectively the shape of the core power change as a function of time. In the MPACT iteration scheme, the next time step's power is calculated before feedback operations; therefore, the integral

of $p(t)$ can be numerically evaluated at the start of the calculation. Thus, $N_I(t_2)$ is known if beginning-of-time step values are used for everything else in Eq. (10.23). Note that the approximation in Eq. (10.22) is not necessarily a good one; this is discussed later in greater detail.

Now a similar procedure can be applied to the xenon equation:

$$\frac{dN_{Xe}}{dt} + (\sigma_a \phi(t) + \lambda_{Xe}) N_{Xe}(t) = \gamma_{Xe} \Sigma_F \phi(t) + \lambda_I N_I(t) \ . \tag{10.24}$$

Since the iodine concentration does not depend on the xenon concentration, the RHS of the equation is already known. An integrating factor can be applied to the left-hand side as was done with iodine, except that the integrating factor will be more complex in this case.

$$\begin{aligned} v(t) &= \int (\sigma_a \phi(t') + \lambda_{Xe})\, dt' \\ &= \sigma_a \overline{\phi} \int p(t')\, dt' + \lambda_{Xe} t \\ &= \sigma_a \overline{\phi} \left(b_p t + \frac{m_p}{2} t^2 \right) + \lambda_{Xe} t \ , \end{aligned} \tag{10.25}$$

where we have assumed linear behavior for the power $p(t) = p_0 + p_1 t$, $b_p = p(0)$, and $m_p = \frac{p(t_2)-p(t_1)}{t_2-t_1}$.

$$\frac{d}{dt}\left[e^{v(t)} N_{Xe}(t) \right] = e^{v(t)} \gamma_{Xe} \Sigma_F \phi(t) + e^{v(t)} \lambda_I N_I(t)$$

$$\int_{t_1}^{t_2} (\cdot) \Rightarrow e^{v(t_2)} N_{Xe}(t_2) - e^{v(t_1)} N_{Xe}(t_1) = \gamma_{Xe} \Sigma_f \overline{\phi} \int_{t_1}^{t_2} e^{v(t')} p(t')\, dt' + \lambda_I \int_{t_1}^{t_2} e^{v(t')} N_I(t')\, dt'$$

$$e^{v(t_2)} N_{Xe}(t_2) = e^{v(t_1)} N_{Xe}(t_1) + \gamma_{Xe} \Sigma_f \overline{\phi} \int_{t_1}^{t_2} e^{v(t')} p(t')\, dt' + \lambda_I \int_{t_1}^{t_2} e^{v(t')} N_I(t')\, dt'$$

$$N_{Xe}(t_2) = e^{v(t_1)-v(t_2)} N_{Xe}(t_1) + \gamma_{Xe} \Sigma_f \overline{\phi} e^{-v(t_2)} \int_{t_1}^{t_2} e^{v(t')} p(t')\, dt' + \lambda_I e^{-v(t_2)} \int_{t_1}^{t_2} e^{v(t')} N_I(t')\, dt'$$

$$t_1 = 0 \Rightarrow v(t_1) = 0 \Rightarrow N_{Xe}(t_2) = e^{-v(t_2)} \left(N_{Xe}(0) + \gamma_{Xe} \Sigma_f \overline{\phi} \int_0^{t_2} e^{v(t')} p(t')\, dt' + \lambda_I \int_0^{t_2} e^{v(t')} N_I(t')\, dt' \right)$$

$$N_{Xe}(t_2) = e^{-\left(\sigma_a \overline{\phi}\left(\frac{p(t_2)+p(0)}{2}\right)+\lambda_{Xe}\right)t} \left(N_{Xe}(0) + \gamma_{Xe} \Sigma_f \overline{\phi} \int_0^{t_2} e^{v(t')} p(t')\, dt' + \lambda_I \int_0^{t_2} e^{v(t')} N_I(t')\, dt' \right) \tag{10.26}$$

Equation (10.23) can be solved directly if the power $p(t')$ is assumed to be linear. Applying integration by parts to the integral yields a solution that can be easily implemented in the code. Equation (10.26) is less straightforward. Everything on the right-hand side of the equation is known, but the integrals have highly complicated solutions, even assuming that $p(t')$ and $N_I(t')$ are linear. To resolve this issue, the integrals can be numerically integrated instead. This is accomplished by using the trapezoidal rule with small intervals to resolve the exponential shape with sufficient resolution to maintain accuracy.

CHAPTER 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), which involves the time discretization and the precursor integration technique. The chapter then gives the 2D-1D discretization of the TFSP for 2D MOC and then for the 1D axial solver. This follows closely with chapters 2, 4, 5, and 6, in which 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. [42], [43]. Because thermal-hydraulic feedback is an essential part of the transient analysis, a discussion 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).

$$\begin{aligned}\frac{1}{v_g}\frac{\partial\psi_g(\mathbf{r},\mathbf{\Omega},t)}{\partial t} &= -\mathbf{\Omega}\cdot\nabla\psi_g(\mathbf{r},\mathbf{\Omega},t) - \Sigma_{t,g}(\mathbf{r},t)\psi_g(\mathbf{r},\mathbf{\Omega},t) \\ &\quad + \sum_{g'=1}^{G}\int_0^{4\pi}\Sigma_{s,g'\rightarrow g}(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}',t)\psi_{g'}(\mathbf{r},\mathbf{\Omega}',t)d\Omega' \\ &\quad + \frac{1}{4\pi}\left(\chi_{p,g}(\mathbf{r},t)(1-\beta(\mathbf{r},t))S_F(\mathbf{r},t) + \chi_{d,g}(\mathbf{r},t)S_d(\mathbf{r},t)\right),\end{aligned} \tag{11.1}$$

$$\frac{\partial C_\tau(\mathbf{r},t)}{\partial t} = \beta_\tau(\mathbf{r},t)S_F(\mathbf{r},t) - \lambda_\tau C_\tau(\mathbf{r},t) \quad ,\tau = 1,2,...,6, \tag{11.2}$$

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 multigroup 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 spectra, 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 equations shown above, 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 [45], and so on. A detailed discussion of these methods can be found in [74]. The most commonly used second order θ-method requires the explicit calculation and storage of the right-hand side (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 left-hand side (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 RHS 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 [146], [74] and is widely used in current state-of-the-art time-dependent transport solver codes [47], [132], [41].

$$\frac{\psi_g^n(\mathbf{r},\mathbf{\Omega}) - \psi_g^{n-1}(\mathbf{r},\mathbf{\Omega})}{v_g \Delta t_n} \approx \frac{\phi_g^n(\mathbf{r},\mathbf{\Omega}) - \phi_g^{n-1}(\mathbf{r},\mathbf{\Omega})}{4\pi v_g \Delta t_n}. \tag{11.7}$$

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 are then substituted into Eq. (11.1). The remainder 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 step's 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 by substituting 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:

$$C_\tau^n(\mathbf{r}) = \tilde{\Omega}_\tau^0(\tilde{\lambda}_\tau^n) C_\tau^{n-1}(\mathbf{r}) + \frac{1}{\lambda_\tau^n} \left(\beta^n(\mathbf{r}) S_F^n(\mathbf{r}) \tilde{\Omega}_\tau^n(\tilde{\lambda}_\tau^n) + \beta^{n-1}(\mathbf{r}) S_F^{n-1}(\mathbf{r}) \tilde{\Omega}_\tau^{n-1}(\tilde{\lambda}_\tau^n) + \beta^{n-2}(\mathbf{r}) S_F^{n-2}(\mathbf{r}) \tilde{\Omega}_\tau^{n-2}(\tilde{\lambda}_\tau^n)\right), \tag{11.10}$$

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), utilizing 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

$$\mathbf{\Omega} \nabla \psi_g^n(\mathbf{r}, \mathbf{\Omega}) + \Sigma_t^n \psi_g^n(\mathbf{r}, \mathbf{\Omega}) = \sum_{g'=1}^{G} \int_0^{4\pi} \Sigma_{s,g' \to g}^n(\mathbf{r}, \mathbf{\Omega} \cdot \mathbf{\Omega}') \psi_{g'}^n(\mathbf{r}, \mathbf{\Omega}') d\mathbf{\Omega}' + \frac{1}{4\pi} \left(\chi_g^n(\mathbf{r}) S_F^n(\mathbf{r}) + S_{tr,g}^n(\mathbf{r}) \right), \tag{11.15}$$

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, because it is written in such a way that it may readily be solved by any standard steady-state neutron transport solver after discretizing over space and angle. The 2D MOC solution of Eq. (11.15) is discussed below in Section 11.1.3.

11.1.3 2D MOC Solution of the Transient Fixed Source Problem

The discretized steady-state transport 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 $q_{m,g}^n$ as:

$$q_{m,g}^n(x,y) = \sum_{g'=1}^{G} \sum_{m'=1}^{M} \Sigma_{s,g' \to g}^n(x,y, \mathbf{\Omega}_{m'} \cdot \mathbf{\Omega}_m) \psi_{m',g'}^n(x,y) w_{m'} + \frac{1}{4\pi} [\chi_g^n(x,y) S_F^n(x,y) + A_g^n(x,y) \phi_g^n + B_g^n(x,y) S_F^n + C_g^n(x,y)] - \frac{1}{4\pi \Delta z} [J_{z,T,g}^n(x,y) - J_{z,B,g}^n(x,y)]. \tag{11.20}$$

In this formulation of the 2D MOC problem, all the coefficients in Eq. (11.20) are averaged over the axial direction z.

$$A_g^n(x,y) = \frac{\int_{z_B}^{z_T} A_g^n(x,y,z)\phi_{g,m}^n(x,y,z)dz}{\int_{z_B}^{z_T} \phi_{g,m}^n(x,y,z)dz}, \tag{11.21a}$$

$$B_g^n(x,y) = \frac{\int_{z_B}^{z_T} B_g^n(x,y,z)S_F^n(x,y,z)dz}{\int_{z_B}^{z_T} S_F^n(x,y,z)dz}, \tag{11.21b}$$

$$C_g^n(x,y) = \frac{\int_{z_B}^{z_T} C_g^n(x,y,z)dz}{\int_{z_B}^{z_T} dz}. \tag{11.21c}$$

Because the MOC solution of Eq (5.3) in Chapter 5 is performed 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 using essentially the same procedure as that given in Section 11.1.3. The steady-state 1D equation Eq. (4.10) from Chapter 4 is rewritten below:

$$\mu\frac{\partial\hat{\psi}}{\partial z}(\mathbf{r},\mu) + \Sigma_t\hat{\psi}(\mathbf{r},\mu) = \frac{\Sigma_s}{2}\int_{-1}^{1}\hat{\psi}(\mathbf{r},\mu')d\mu' + \frac{\nu\Sigma_f}{2k_{\text{eff}}}\int_{-1}^{1}\hat{\psi}(\mathbf{r},\mu')d\mu' - \frac{1}{2}\left[\frac{\partial J_x}{\partial x}(\mathbf{r}) + \frac{\partial J_y}{\partial y}(\mathbf{r})\right]. \tag{11.22}$$

The discretizations yielding Eq. (11.15) are still amenable to the 2D-1D procedure, so the transient form of Eq. (11.22) is simply

$$\mu\frac{\partial\hat{\psi}^n}{\partial z}(\mathbf{r},\mu) + \Sigma_t^n\hat{\psi}^n(\mathbf{r},\mu) = \frac{\Sigma_s^n}{2}\int_{-1}^{1}\hat{\psi}^n(\mathbf{r},\mu')d\mu' + \frac{\nu\Sigma_f^n}{2k_{\text{eff}}}\int_{-1}^{1}\hat{\psi}^n(\mathbf{r},\mu')d\mu' - \frac{1}{2}\left[\frac{\partial J_x^n}{\partial x}(\mathbf{r}) + \frac{\partial J_y^n}{\partial y}(\mathbf{r})\right] + S_t^n r(\mathbf{r}). \tag{11.23}$$

As noted in Chapter 6, the solution of equations in 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:

$$S_{tr,g,m}^n(\xi) = \sum_{k=0}^{2} S_{tr,k}P_k(\xi), \tag{11.24}$$

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 multigroup sweeping (MGS) CMFD and a multigroup matrix (MGM) CMFD. The MGS CMFD method solves the CMFD equations group by group, and it requires multiple iterations over energy to converge the energy-dependent source, especially for transient problems with large power changes. The MGM CMFD formulates one single matrix for the whole multigroup 3D CMFD problem and uses PETSc [9] 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 because it performs much more efficiently than MGS CMFD for all problems analyzed thus far.

The derivation of the CMFD operator follows essentially the same procedure for transient as for steady-state. The discretized *steady-state* CMFD operator given in Eq. (7.2) is rewritten below:

$$\sum_s -\tilde{D}_{g,j,s}\left(\phi_{g,j}-\phi_{g,j,s}\right)+\hat{D}_{g,j,s}\left(\phi_{g,j}+\phi_{g,j,s}\right)\delta A_{j,s} + \Sigma_{t,g,j}\phi_{g,j}V_j = \left[\sum_{g'=1}^{G}\left(\Sigma_{s0,g'\to g,j}+\frac{\chi_g}{k_{\text{eff}}}\nu\Sigma_{f,g',j}\right)\phi_{g',j}\right]V_j, \quad (11.25)$$

where $\tilde{D}$ and $\hat{D}$ have the usual definitions from Eq. (7.6) and Eq. (7.8). $\delta A_{j,s}$ 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

$$\sum_s -\tilde{D}^n_{g,j,s}\left(\phi^n_{g,j}-\phi^n_{g,j,s}\right)+\hat{D}^n_{g,j,s}\left(\phi^n_{g,j}+\phi^n_{g,j,s}\right)\delta A_{j,s} + \Sigma^n_{t,g,j}\phi^n_{g,j}V_j = \left[\sum_{g'=1}^{G}\left(\Sigma^n_{s0,g'\to g,j}+\frac{\chi^n_g}{k_{\text{eff}}}\nu\Sigma^n_{f,g',j}\right)\phi^n_{g',j}\right]V_j + S^n_{tr,g,j}V_j. \quad (11.26)$$

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:

$$A^n_{g,j}=\frac{\sum_{i\in j}A^n_{g,i}\phi^n_{g,i}V_i}{\sum_{i\in j}\phi^n_{g,i}V_i}, \quad (11.27a)$$

$$B^n_{g,j}=\frac{\sum_{i\in j}B^n_{g,i}\chi^n_g S_{F,g,i}V_i}{\sum_{i\in j}\chi^n_g S_{F,g,i}V_i}, \quad (11.27b)$$

$$C^n_{g,j}=\frac{\sum_{i\in j}C^n_{g,i}V_i}{\sum_{i\in j}V_i}. \quad (11.27c)$$

The solution of the aforementioned equations is described in subsections 11.1.5.1 and 11.1.5.2. To facilitate the notation in these sections, the following *operator* notation is presented:

$$\text{Removal term:} \quad \mathbf{M}=\mathbf{D}+\mathbf{\Sigma}_t \quad (11.28a)$$

$$\text{Scattering:} \quad \mathbf{S} = \mathbf{\Sigma}_s \tag{11.28b}$$

$$\text{Fission source:} \quad \mathbf{F} = \frac{\chi \nu \mathbf{\Sigma}_f}{k_{\text{eff}}} \tag{11.28c}$$

$$\text{Transient source:} \quad \mathbf{S_{tr}} = \mathbf{A}\phi + \mathbf{BF}\phi + \mathbf{C} \tag{11.28d}$$

Here, $\mathbf{\Sigma}_t$ 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 multigroup 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 using this type of iteration is to minimize storage requirements for the linear system solved in Eq. (11.29); this type of iteration also prevents 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, because the RHS (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 above, 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 LHS. 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.

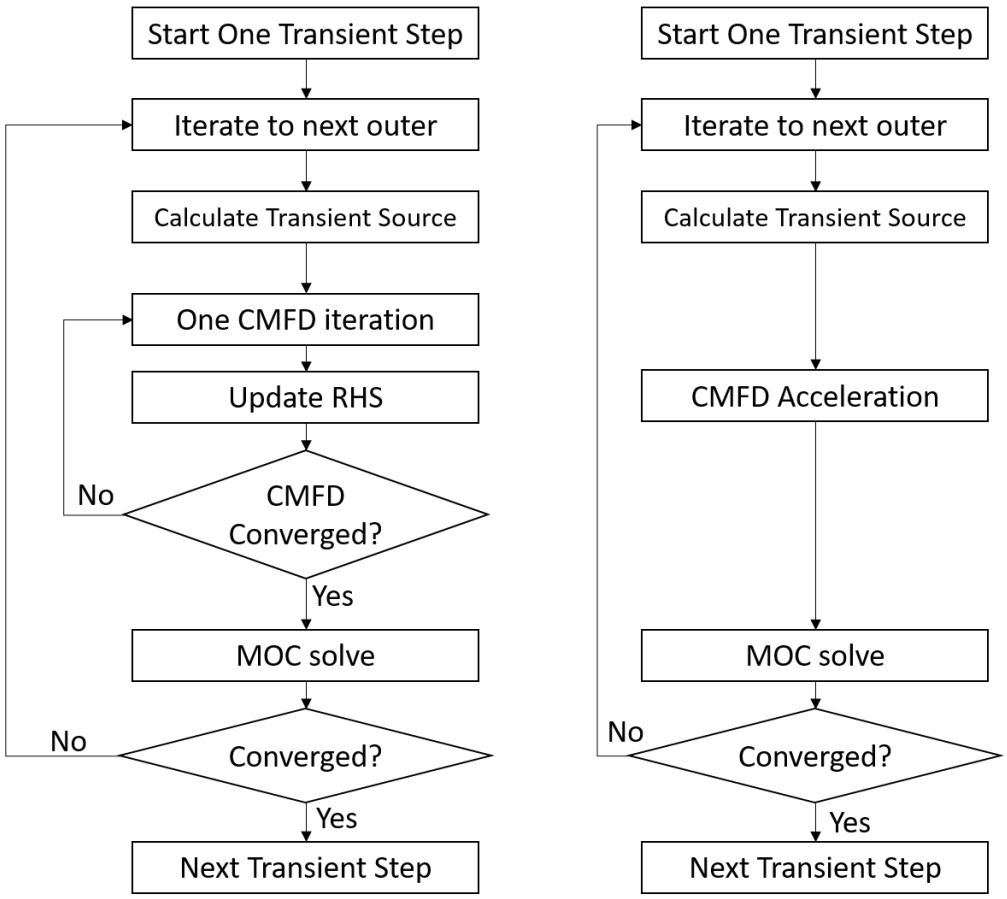


Figure 11.1. Flow charts for MGS (left) and MGM (right) CMFD.

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 is iteratively solved with 2D or 3D CMFD acceleration until the convergence criteria are satisfied.

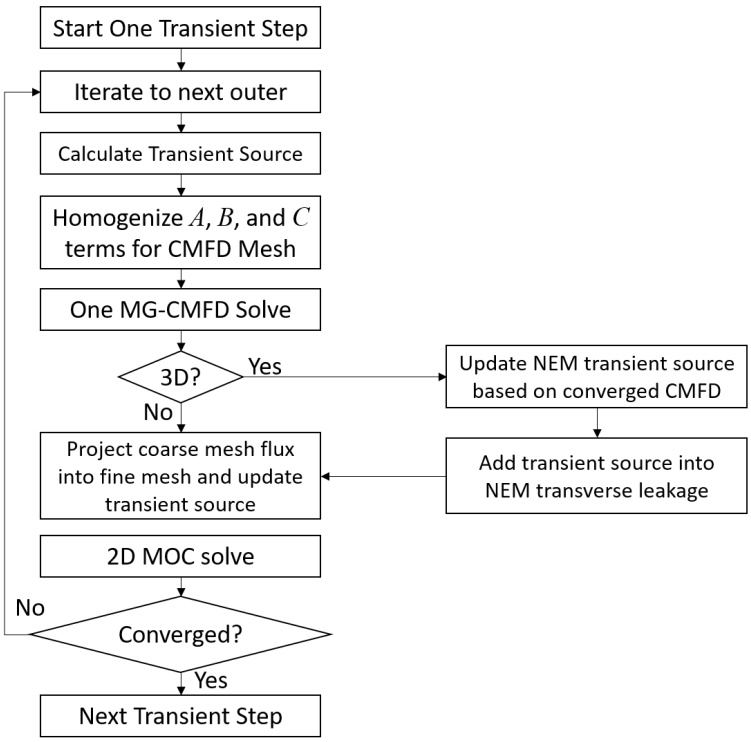


Figure 11.2. 2D-1D transient iteration scheme.

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, which is 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 Multigroup CMFD (MG) TFSP. Finally the coarsest, third level makes use of the exact point kinetic equations (EPKEs). 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.

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 [116]. One of the important perturbations for practical neutron transport applications is the insertion of reactivity, where the change of material composition, temperature, and other factors 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 [149]. 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 [40] 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 [40]. As described in Ott and Neuhold [116], 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:

$$< \mathbf{\Psi}, \boldsymbol{H}\mathbf{\Phi} >=< \boldsymbol{H}^*\mathbf{\Psi}, \mathbf{\Phi} >=< \mathbf{\Phi}, \boldsymbol{H}^*\mathbf{\Psi} > . \tag{11.31}$$

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's work [116], 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 [9]. The fission source operator $\mathbf{F}$ is not constructed explicitly, but rather, it 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):

$$\mathbf{F} = \boldsymbol{\chi}\boldsymbol{\nu}\boldsymbol{\Sigma}_f, \tag{11.34}$$

$$\mathbf{F}^* = \boldsymbol{\nu}\boldsymbol{\Sigma}^*_f\boldsymbol{\chi}^*. \tag{11.35}$$

Equation (11.33) can also be solved using the power iteration, which is the same procedure that is used to solve the forward CMFD equation. Because 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, $\mathbf{\Omega}$, and then it uses the integration defined in Eq. (11.31) to integrate over space and energy. The detailed derivation is provided in Dulla et al. [38, 51]. The result of this procedure yields the EPKEs as

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)}p(t) + \frac{1}{\Lambda(0)}\sum_{\tau}\lambda_\tau(t)\zeta_\tau(t), \tag{11.36}$$

$$\frac{d\zeta_\tau(t)}{dt} = \frac{\Lambda(0)}{\Lambda(t)}\beta^{eff}_\tau(t)p(t) - \lambda^{PK}_\tau(t)\zeta_\tau(t) \quad , \quad \tau = 1, 2, ..., 6, \tag{11.37}$$

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, because 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 four-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 (1) 3D transport to the 3D CMFD and (2) the 3D CMFD to the EPKE.

This chapter introduces the adjoint flux necessary to couple the spatial (CMFD) and amplitude (EPKE) flux levels, and then it provides the details of the derivation and implementation of the TML method.

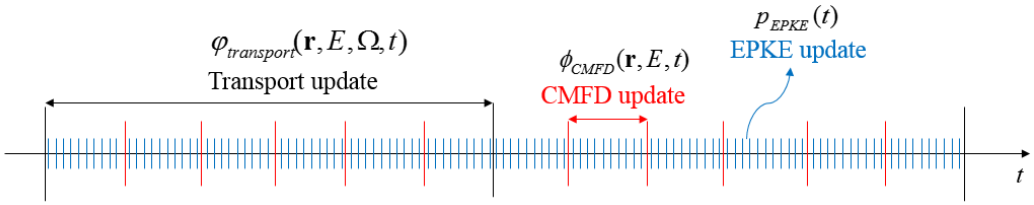


Figure 11.3. Illustration of TML scheme.

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, then 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, so 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, then the amplitude function in each CMFD cell is identical to the corresponding CMFD scalar flux:

$$\frac{1}{V_j}\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \psi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV = \frac{1}{V_j}P_{j,g}(t)\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \Phi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV = P_{g,j}(t). \tag{11.47}$$

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 the constraint in Eq. (11.46) is used again, then the LHS reduces to Eq. (11.49).

$$\begin{aligned}
&\frac{1}{V_j}\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \frac{\partial\psi_g(\mathbf{r},\mathbf{\Omega},t)}{v_g\partial t}d\mathbf{\Omega}dV \\
&= \frac{1}{V_j v_g}\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \left(P_{g,j}(t)\frac{\partial\Phi_g(\mathbf{r},\mathbf{\Omega},t)}{\partial t} + \Phi_g(\mathbf{r},\mathbf{\Omega},t)\frac{\partial P_{g,j}(t)}{\partial t}\right)d\mathbf{\Omega}dV \\
&= \frac{1}{V_j v_g}P_{g,j}(t)\frac{\partial}{\partial t}\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \Phi_g(\mathbf{r},\mathbf{\Omega},t) + \frac{1}{V_j v_g)}\frac{\partial P_i(E,t)}{\partial t}\int\limits_{\mathbf{r}\in j}\int\limits_{\mathbf{\Omega}} \Phi_g(\mathbf{r},\mathbf{\Omega},t)d\mathbf{\Omega}dV \\
&= \frac{1}{v_g}\frac{\partial P_{g,j}(t)}{\partial t}.
\end{aligned} \tag{11.49}$$

The RHS would be the same as Eq. (7.2).

The derivation presented above 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 *transport CMFD* (TCMFD) method [42].

The PCQM algorithm for the transport/CMFD level is as follows:

- 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 that 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 the paper by Cho et al. (2005) [47].

The PCQM algorithm for CMFD/EPKE level is then as follows:

- Step 1: Evaluate the steady-state CMFD-based adjoint flux $\phi^*_{g,j}$ [40] 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 $\phi^P_{g,j}(t_n)$ 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 steps 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 [44] 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. However, several physical phenomena related to the conduction—such as creep, swelling, and thermal expansion, for example—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. It should be noted that the explicit TH/neutronics coupling and sub-pin Doppler feedback require that special attention be given to the TML couplings. Furthermore, 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 [127], which includes 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. Because of the subpin-dependent Doppler temperature feedback, the TH calculation is performed in each transport time step. Therefore, 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.

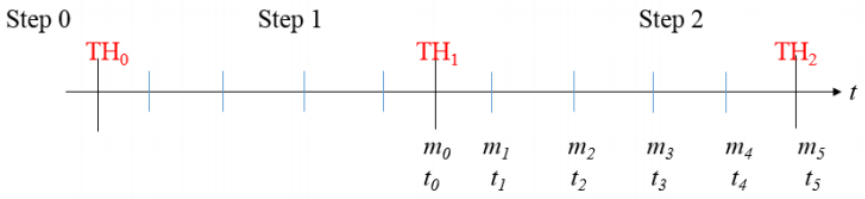


Figure 11.4. Illustration of first-level TH coupling for TML.

With explicit coupling, the TH condition is updated at the end of each transport step and is used for the next transport time step. For example, TH_1 calculated at the end of step 1 and 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. Because 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 as shown in 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.

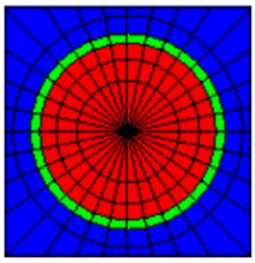


Figure 11.5. Single pin case geometry to demonstrate TH feedback.

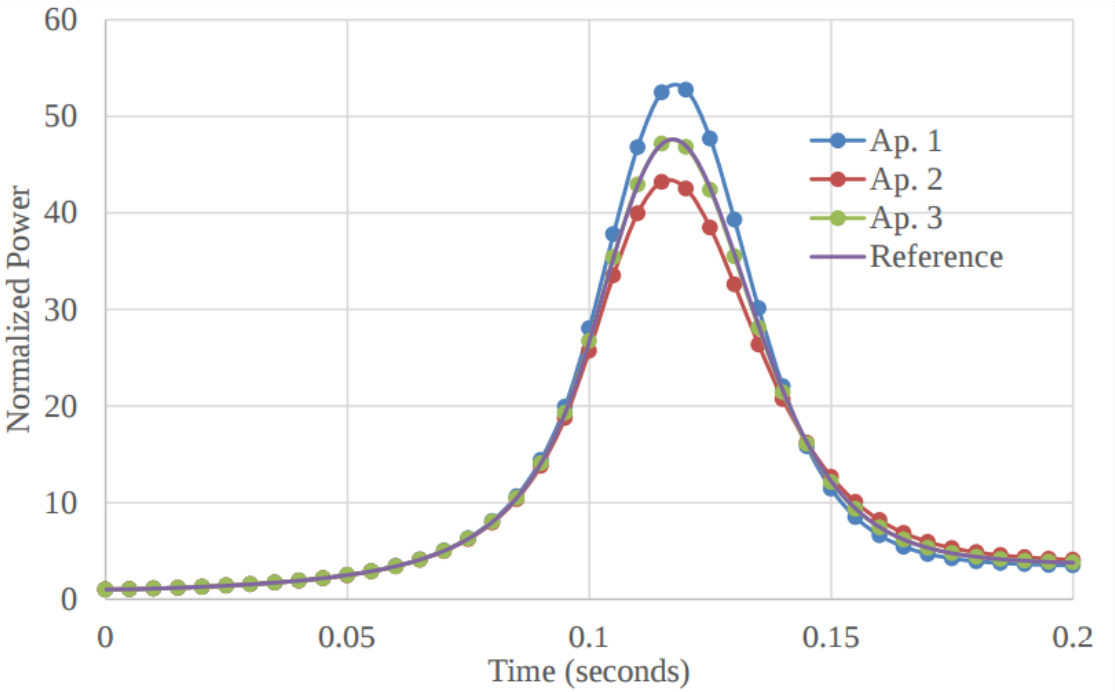


Figure 11.6. Power pulse with three approximations.

As shown in Figure 11.6, approximation 3 predicts the solution very well, whereas approximations 1 and 2 overestimate and underestimate the pulse height, respectively. In approximation 1, the TH condition used is from 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 as a result of the Doppler feedback effect, and the flux in the fuel region is reduced. If the weighting flux is from the previous time step, then the flux decrement from the increase of absorption is ignored, resulting in overestimation of the Doppler feedback and underestimating the pulse height.

Approximation 3 is a reasonable assumption, because 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, and so on. Therefore, a reasonable approximation to the flux is acceptable, because it is only a weighting function. Secondly, the TH sub-pin temperature distribution is global, whereas 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. Therefore, 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 must 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, in which 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, as 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.

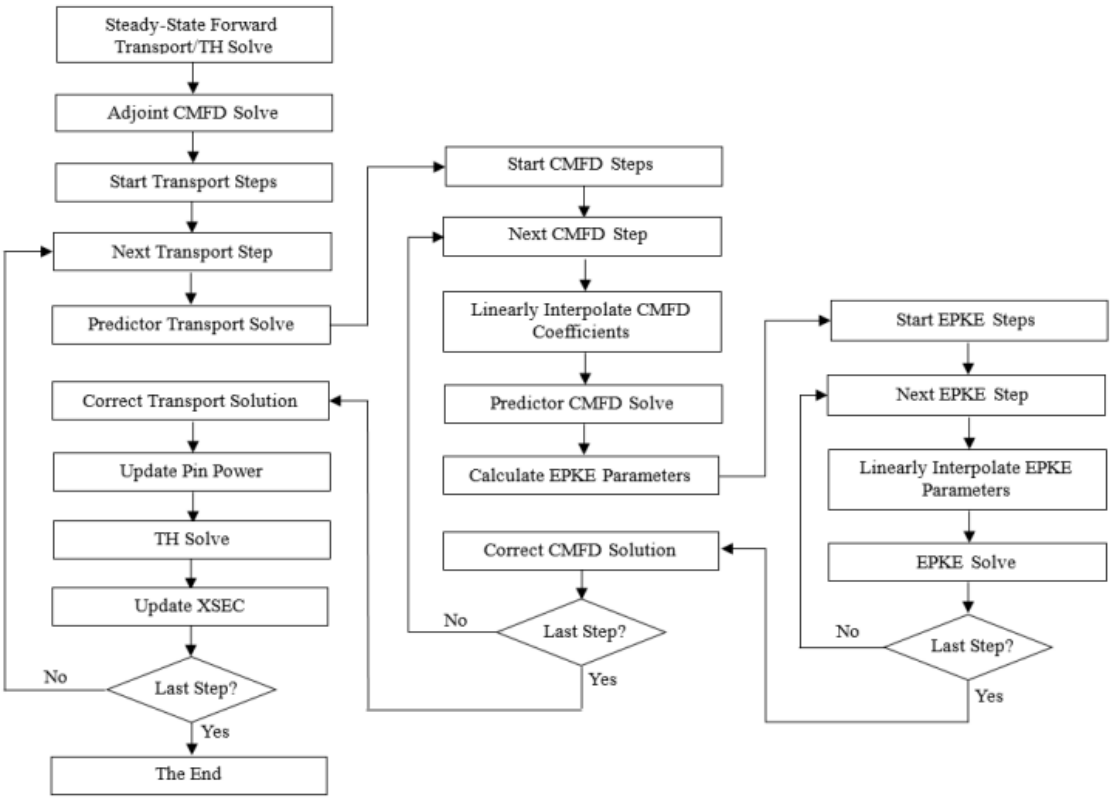


Figure 11.7. Flow chart for TML with TH feedback.

11.2.6 Transient Multilevel Method with One Group CMFD Acceleration

A one-group CMFD (1GCMFD) solver is incorporated into TML to improve the overall performance. In the new TML method, the 1GCMFD equation is introduced into the original multigroup CMFD equation (MGCMFD) to converge the transient fixed source. At the same time, a new 1GCMFD level is introduced between the 3D CMFD level and the EPKE level to correct MGCMFD solutions and reduce the number of MGCMFD calculations. The new TML (TML4) method was shown to be more accurate and faster than the

original TML method, because 1GCMFD equations help reduce the complexity and computational intensity of MGCMFD calculations.

This chapter introduces the 1GCMFD equation to accelerate MGCMFD equation, and then it provides the details of the derivation and implementation of the 1GCMFD.

11.2.6.1 One-group CMFD acceleration in MGCMFD calculation

The one-group and two-group diffusion equations have been well studied and used to accelerate the higher order calculations in steady-state simulations [1, 161, 164]. The 1GCMFD method is one acceleration scheme. It may be viewed as an extension of the CMFD method in which a lower order method is used to accelerate the higher order method. Recently, a multilevel in space and energy diffusion (MSED) method which used a 1GCMFD has been implemented in MPACT to accelerate the steady-state eigenvalue problem [161]. Following the same idea, the 1GCMFD is used to accelerate the MGCMFD calculations in the TFSP. It is assumed that 1GCMFD is a sufficiently accurate approximation to converge the fission source, and this has been demonstrated for steady calculations [161]. With the 1GCMFD, the fission source term in MGCMFD is moved to RHS, and the MGCMFD system becomes much easier to solve. Because the number of unknowns in 1GCMFD is much smaller than in the MGCMFD system, the overall computation time to solve MGCMFD is reduced, although some computation effort is needed to solve the 1GCMFD system.

The 1GCMFD coefficients are collapsed from MGCMFD coefficients. Then the 1GCMFD flux is obtained by solving the 1GCMFD equation, and the fission source can be calculated. Eventually, with the fission source from the 1GCMFD, the MGCMFD scalar flux is calculated. The original MGCMFD calculations become the inner iterations between the simpler 1G/MGCMFD calculations.
The details of the collapsing process are illustrated here. For cross sections,

$$< \Sigma_m^n >= \frac{\sum_{g=1}^{G} \phi_{g,m}^n \Sigma_{g,m}}{\Phi_m^n}, \quad \Sigma = A,\ \Sigma_a,\ \nu\Sigma_f, \tag{11.61}$$

where m denotes the index of coarse mesh, and

$$\Phi_m^n = \sum_{g=1}^{G} \phi_{g,m}^n. \tag{11.62}$$

For the spectrum and source,

$$< S_m^n >= \sum_{g}^{G} S_{m,g}^n, \quad S = C,\ B. \tag{11.63}$$

The diffusion coefficients can be collapsed just as the cross sections. The details about how the coefficients are collapsed can be found in [161]. Essentially, the MGCMFD equations are simply summed over energy. Once the 1GCMFD coefficients are calculated, the 1GCMFD scalar flux is calculated by

$$(\boldsymbol{M_1 - S_1 - F_1 - A_1 - B_1 F_1})\boldsymbol{\Phi_1 = C_1}, \tag{11.64}$$

where the subscript 1s denote the 1GCMFD coefficient. Then the MGCMFD flux is calculated by

$$(\boldsymbol{M - S - A - BF})\boldsymbol{\Phi = C + \chi F_1 \Phi_1}. \tag{11.65}$$

11.2.6.2 1GCMFD level in TML

Beyond a simple 1GCMFD accelerator of the MGCMFD system, the TML scheme is modified to include a 1GCMFD level. This is illustrated in Figure 11.8. At this level, the shape function of coarse mesh scalar flux in energy is assumed to vary slowly in time, and the amplitude function is needed to be corrected by 1GMCFD calculation. 1GCMFD solutions are only used to update MGCMFD results. As a result, there is no direct coupling between 1GCMFD and EPKE. The EPKE is coupled with MGCMFD solutions corrected with 1GCMFD.

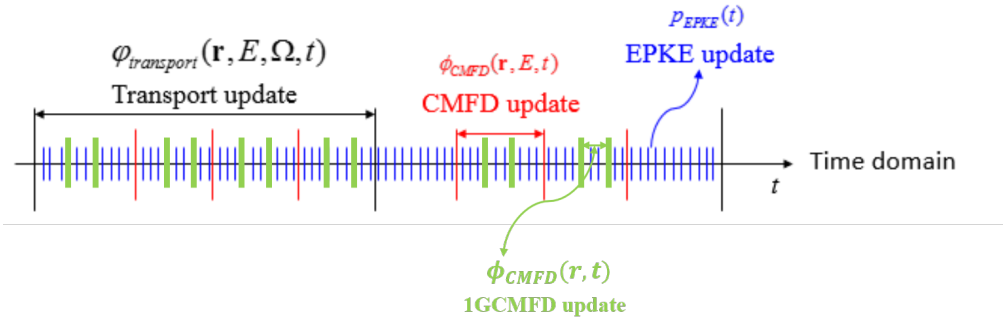


Figure 11.8. TML iteration scheme with 1GCMFD (TML-4).

11.2.6.3 1GCMFD/MGCMFD coupling in TML

The 1GCMFD can be factorized as

$$\phi_{g,j}(t) = \Phi_j(t)\varphi_{g,j}(t), \tag{11.66}$$

where $\phi_{g,j}$ is the MGCMFD flux in cell j for group g, and $\Phi_j(t)$ and $\varphi_{g,j}(t)$ represent the coarse mesh–wise single group amplitude and coarse mesh group dependent shape functions, respectively. Once 1GCMFD flux is solved from Equation 11.61, the MGCMFD flux is updated by

$$\phi^C_{g,i} = \frac{\Phi^C_i}{\sum_{g=1}^{G}\phi^P_{g,i}}\phi^P_{g,i}. \tag{11.67}$$

The PCQM algorithm for the MGCMFD/1GCMFD level is as follows:

- Step 1:Solve the MGCMFD TFSP for MGCMFD step $dt_{n,CMFD}$ using Equation 11.26.
- Step 2: Linearly interpolate the MGCMFD coefficients between and $t_{n-1,CMFD}$ and $t_{n,CMFD}$, and the 1GCMFD coefficients are generated using the equations from Equation 11.61 with the predicted MGCMFD flux at $t_{n,CMFD}$.
- Step 3: The 1G transient fixed source is summed from the MG transient fixed source. Solve the 1GCMFD using the $dt_{n,1G} = dt_{n,CMFD}/N$. Update the MGCMFD flux using Equation 11.67.
- Step 4: Update the precursor concentrations and transient fixed sources.

11.2.6.4 Dynamic iteration strategy

The 1GCMFD in TFSP is slightly different from the 1GCMFD in MSED for steady-state eigenvalue problems. For the TFSP, the CMFD coefficients vary for each time step because the cross sections vary as a result of perturbation and feedback. However, in MSED, the CMFD coefficients gradually converge to constants.

Therefore, if the number of iterations is insufficient at the transport level, then the fission source calculated in the 1GCMFD would not be accurate, and the total calculation time would increase as a result of the additional transport iterations required to converge the pin-wise transient source. Moreover, in the MGCMFD level, the MG scalar flux becomes less accurate and cannot capture the the amplitude of the flux.

To remedy this issue, a dynamic iteration strategy is proposed as illustrated in Figure 11.9. *nIter* is the number of 1G/MGCMFD iterations. In the MGCMFD level, *nIter* is set to be 3 based on the results of numerical test investigations. In the 3D transport level, if the reactivity is increasing or decreasing by more than 1%, then the number of sweeping is set to 3. If the outer iteration number of MOC calculation in the 3D

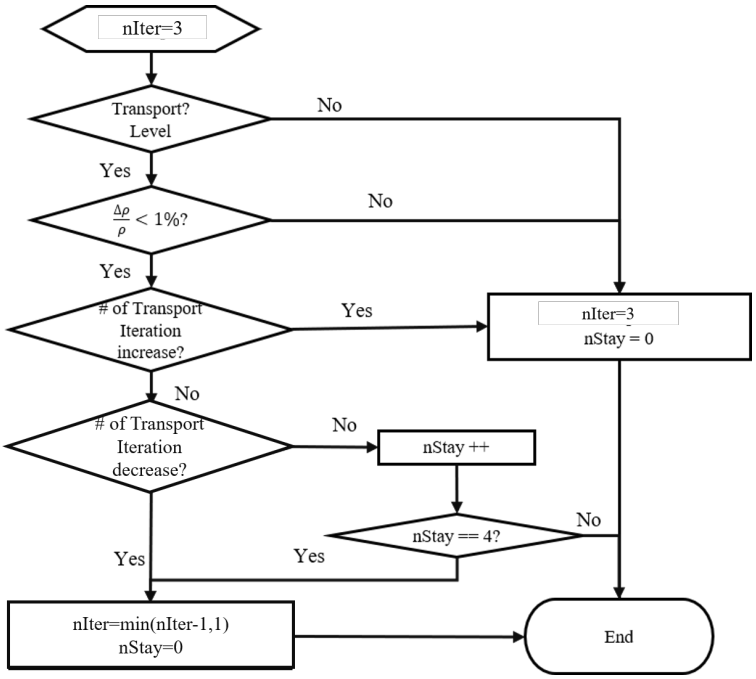


Figure 11.9. Flowchart of dynamic 1G/MGCMFD iterations strategy.

transport calculation stays unchanged for 4 steps or iteration number decreases, then the iteration number is deducted by 1.

In general, the time of one 1G/MGCMFD iteration is about one-fourth of that of a pure MGCMFD calculation, so 3 would be the maximum number of iterations to accelerate the calculation.

11.2.7 Overall Flowchart for TML-4 with TH feedback

To summarize the TML-4 scheme, a chart is provided in Figure 11.10. The second and third vertical blocks represent the application of the 1GCMFD method to accelerate the transient calculation. In the MGCMFD level, the 1GCMFD equation is used to converge the source. In the 1GCMFD level, the 1GCMFD solver is used to reduce the number of MGCMFD steps, assuming that the flux shape varies slowly in energy.

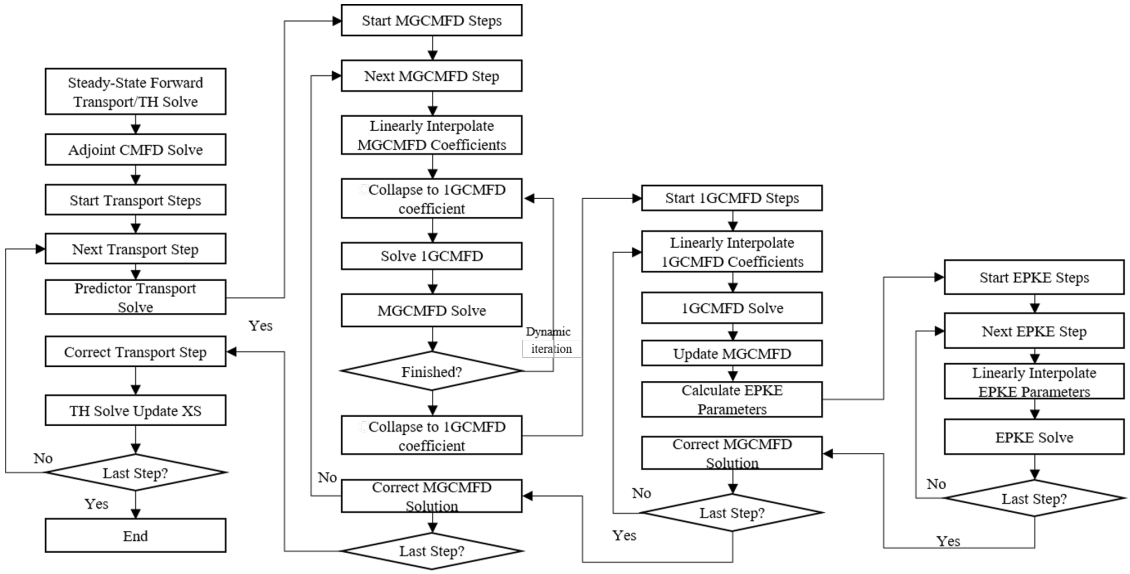


Figure 11.10. Flowchart of TML iteration scheme with 1GCMFD.

CHAPTER 12. MULTIPHYSICS COUPLING

This chapter describes some of the multiphysics coupling capabilities included in MPACT. These include internal TH solvers, coupling the CTF TH code and its ANTS solver [127,128], species transport capabilities for advanced reactors, chemistry coupling for CRUD calculations, and coupling to the thermochemistry code *Thermochimica* [120, 121].

12.1. SIMPLIFIED TH

There are two TH solvers in MPACT. The first is a simplified TH solver which 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.

12.1.1 Fluid Flow Model

12.1.1.1 Single-Phase Flow

The first flow model in MPACT provides the basic mass-energy balance of the fluid flow through the reactor. This is applicable for single phase flow for PWRs operating at 2250 psia. 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:

$$\bar{h} = \frac{h_{out} + h_{in}}{2} \, . \tag{12.3}$$

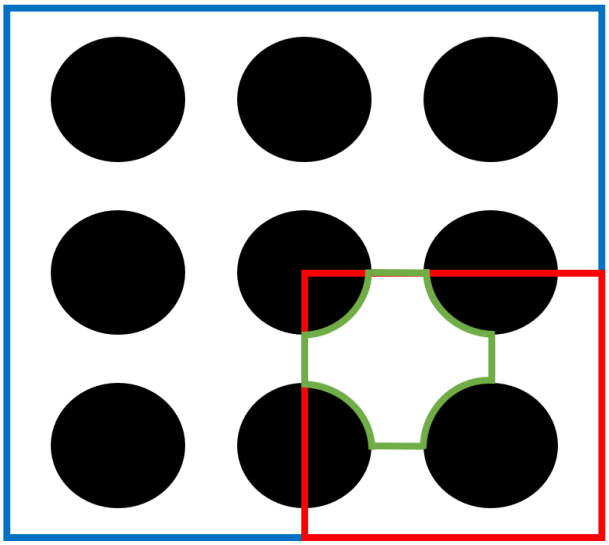


Figure 12.1. Representative flow channels for a 3×3 assembly: full assembly (blue), quarter assembly (red), or closed channel (green).

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 subchannel TH code CTF [127], where the property of interest can be looked up as a function of enthalpy or temperature. The pressure is assumed to be 2,250 psia, the nominal operating pressure for most PWRs. The temperature ranges from 279 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 is applied to each pin in the core.

12.1.1.2 Two-Phase Flow

For BWRs, significantly more physics are included in order to ensure a sufficient approximation of the boiling and two-phase flow. Start with the steady-state mixture mass, momentum, and energy balance for a closed channel:

$$\nabla\left(\rho_m v_m\right) = 0 , \tag{12.5a}$$

$$\nabla\left(\rho_m v_m^2\right) = -\nabla P + \nabla\left(\tau + \tau_{turb}\right) - \nabla\left(\frac{\alpha}{1-\alpha}\frac{\rho_f \rho_g}{\rho_m} V_{gj}^2\right) + \rho_m g + M , \tag{12.5b}$$

$$\nabla\left(\rho_m v_m h_m\right) = \nabla q - \nabla\left(\alpha \frac{\rho_f \rho_g}{\rho_m} V_{gj} h_m\right) + \frac{DP}{Dt} + \Lambda . \tag{12.5c}$$

Several terms are neglected for this simplified solver:

1. Ignore the momentum carried by the drift velocity: $\nabla\left(\frac{\alpha}{1-\alpha}\frac{\rho_f\rho_g}{\rho_m}V_{gj}^2\right)\approx 0$,
2. Ignore the extra momentum body force: $M\approx 0$,
3. Ignore the energy involved in void drift: $\nabla\left(\alpha\frac{\rho_f\rho_g}{\rho_m}V_{gj}h_m\right)\approx 0$,
4. Ignore the energy transport through the material derivative of pressure: $\frac{DP}{Dt}\approx 0$,
5. The energy body forace is fission energy directly into the coolant: $\Lambda = q'''$,
6. Ignore all conduction in the liquid: $\nabla q = 0$.

With these assumptions, the final set of equations becomes:

$$\nabla\left(\rho_m v_m\right) = 0\ , \tag{12.6a}$$

$$\nabla\left(\rho_m v_m^2\right) = -\nabla P + \nabla\left(\tau + \tau_{turb}\right) + \rho_m g\ , \tag{12.6b}$$

$$\nabla\left(\rho_m v_m h_m\right) = q'''\ . \tag{12.6c}$$

The equations are integrated over an axial node and the entire area of the channel. In order to discretize this problem, we'll use a face-centered grid scheme with node-averaged quantities being the linear interpolation of the inlet and outlet conditions of the node:

$$\overline{y} = \frac{y_n + y_s}{2}\ ,$$

where y is one of the quantities of interest for the node such as enthalpy, quality, temperature, or density.

The volume integrals become surface derivatives at the north and south faces; body forces will be defined through node averages. This gives the final form of the conservation equations as they are implemented in the node:

$$\left(\rho_m v_m A\right)_n = \left(\rho_m v_m A\right)_s\ , \tag{12.7a}$$

$$\left(\rho_m v_m^2 A\right)_n - \left(\rho_m v_m^2 A\right)_s = -P_n A_n + P_s A_s + \int \nabla\left(\tau + \tau_{turb}\right) dV - \overline{\rho_m} g V\ , \tag{12.7b}$$

$$\left(\rho_m v_m h_m A\right)_n - \left(\rho_m v_m h_m A\right)_s = q''' V\ , \tag{12.7c}$$

where ρ is density, v is velocity, A is the flow area, P is pressure, τ is shear stress, τ_{turb} is the turbulent shear stress, g is the gravitational force constant, V is the node volume, h is the enthalpy, and q''' is the volumetric heat source in the node.

There are three components of the shear stress: wall friction, form losses, and axial shear. For these purposes, only wall shear and form losses are considered, giving the final form of the momentum equation:

$$\left(\rho_m v_m^2 A\right)_n - \left(\rho_m v_m^2 A\right)_s = -P_n A_n + P_s A_s - \frac{\overline{\rho_m v_m^2}}{2}\left(\frac{\Phi_{2\Phi} f}{D_h} + \frac{K_{loss}}{\Delta z}\right) V - \overline{\rho_m} g V\ , \tag{12.8}$$

where $\Phi_{2\Phi}$ is the two-phase friction multiplier, f is the wall friction coefficient, D_h is the hydraulic diameter, K_{loss} is the geometry-dependent forms loss coefficient input by the user, and Δz is the height of the node.

Several closures must also be defined. In each of these closures, the subscripts m, f, and g refer to mixture, saturated liquid, and saturated vapor, respectively.

1. Mixture density:
$$\rho_m = \alpha\rho_g + (1-\alpha)\rho_f \,, \tag{12.9}$$
where *alpha* is the void fraction.

2. Mixture quality, ignoring the effect of subcooled boiling:
$$x = \frac{h_m - h_f}{h_g - h_f} \,; \tag{12.10}$$

3. Void fraction:
$$\alpha = \frac{x}{C_0\left(x + \frac{\rho_g}{\rho_f}(1-x)\right) + \frac{\rho_g V_{gj}}{\rho_m v_m}} \,, \tag{12.11}$$
where C_0 is the distribution parameter and V_{gj} is the drift velocity. In MPACT's STH implementation, these parameter are obtained from Chexal and Lellouche's correlation [19], but other correlations could be used as well;

4. Two-phase friction modifier: the Chisholm correlation is used [20];

5. Wall friction, which uses a modified form of the Blasius relationship [52]:
$$f = 0.1892 Re^{-0.2} \,, \tag{12.12}$$
where *Re* is the Reynold's number.

The iteration scheme for a channel presumes an outlet pressure, an inlet mass flow rate, and an inlet enthalpy. Starting at the inlet, the energy equation will be solved for the north face enthalpy, quality, void fraction, and mixture density. This solution is marched upward until the outlet is reached. Once at the outlet, a sweep downward can be solved to obtain the pressure distribution.

Figure 12.2 illustrates the single channel solve. For each node starting at the bottom:
$$h_n = h_s + \frac{q_i''' V}{\rho_s v_s A_s} \,. \tag{12.13}$$

Compute quality, void fraction, and mixture density, then compute cell-averaged quantities for density, enthalpy, and velocity. The north face then becomes the new south face and the process is repeated. Upon reaching the top, the friction loss terms are computed for the south face of the top node:
$$P_s = \frac{1}{A_s}\left(P_n A_n + \left(\rho_m v_m^2 A\right)_n - \left(\rho_m v_m^2 A\right)_s + \frac{\overline{\rho_m v_m^2}}{2}\left(\frac{\Phi_{2\Phi} f}{D_h} + \frac{K_{loss}}{\Delta z}\right) V + \overline{\rho_m} g V\right) \,. \tag{12.14}$$

The south face then becomes the north face, and the process is repeated until the bottom is reached. This completes the channel solve by determining the total pressure drop in the channel.

The channel solve is performed for every fuel bundle in the core, then a core-wide mass and momentum balance is performed. For the first iteration, the inlet flow distribution is assumed to be uniform. After each iteration, an updated distribution is calculated and normalized:
$$\dot{m}_i^{k+\frac{1}{2}} = \dot{m}_i^k + \frac{d\dot{m}_i}{dP_{i,inlet}}\left(P^k_{core,inlet} - P^k_{i,inlet}\right) \,, \tag{12.15a}$$
$$\dot{m}_i^{k+1} = \frac{\dot{m}_{core}}{\sum_i \dot{m}_i^{k+\frac{1}{2}}}\dot{m}_i^{k+\frac{1}{2}} \,, \tag{12.15b}$$

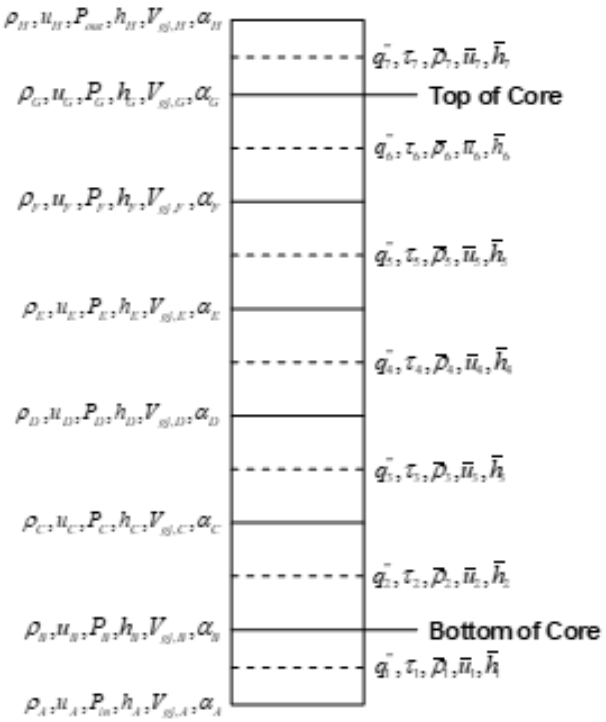


Figure 12.2. Illustration of single channel calculation quantities for BWR STH.

where k is the iteration index and i is the channel index. The average pressure drop across the core is calculated as

$$P^k_{core,inlet} = \frac{\sum_i P^k_{i,inlet}}{N_{chan}} , \tag{12.16}$$

where N_{chan} is the number of channels in the core. If at any point the relative difference between the core-averaged pressure drop and any single channel pressure drop is greater than 10^{-5}, then the derivative of inlet flow with respect to pressure is updated based on the previous two iterations and the fuel bundle calculations are repeated.

12.1.2 1D Heat Conduction

To obtain fuel and clad temperatures, two different methods can be used. The first is a 1D heat conduction solver that can be applied to each pin in the model. The second method uses fuel temperature tables to obtain average fuel and clad temperatures.

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. This is achieved through the use of the Dittus-Boelter heat transfer coefficient, which is modified to account for a regular square array of pins [148]:

$$T_{clad,surf} = T_{mod} + \frac{P_l}{A_{surf} h_{mDB}} , \tag{12.17a}$$

$$h_{mDB} = C_0 \frac{k}{D_h} \mathrm{Re}^{0.4} \mathrm{Pr}^{0.2} , \tag{12.17b}$$

$$C_0 = 0.042 \frac{p_{pin}}{2 r_{clad}} - 0.024 , \tag{12.17c}$$

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 is 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.

The second step is to treat the fuel-clad gap, a small region filled with an inert gas (typically He), providing a relatively large thermal resistance. Many complex physical phenomena 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 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-empirical correlation or tabulated evaluation. The equation relating the fuel pellet's surface temperature to the clad's inner surface temperature is given by

$$T_{fuel,surf} = T_{clad,in} + \frac{P_l}{A_{surf} h_{gap}} , \tag{12.18}$$

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.

The final step is to solve the heat conduction equation,

$$-\boldsymbol{\nabla} \cdot k\left(T\right) \boldsymbol{\nabla} T\left(\boldsymbol{x}\right) = \dot{q}\left(\boldsymbol{x}\right) , \tag{12.19}$$

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 for the solution here,

$$-\frac{1}{r}\frac{d}{dr} r k\left(T\right) \frac{d}{dr} T\left(r\right) = \dot{q}\left(r\right) . \tag{12.20}$$

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:

$$T\left(r\right) = -\frac{\dot{q}}{4k} r^2 + C_0 \ln\left(r\right) + C_1 , \tag{12.21a}$$

$$T\left(r_{i+1}\right) = T_{out} , \tag{12.21b}$$

$$q_i'\left(r_i\right) = -k_i \frac{dT}{dr} . \tag{12.21c}$$

These conditions result in the following equation for the temperature inside a volume:

$$T\left(r\right) = \frac{\dot{q}_i}{4k_i}\left(r_{i+1}^2 - r^2\right) + \left(\frac{\dot{q}_i r_i^2}{2k_i} - \frac{q_i' r_i}{k_i}\right) \ln\left(\frac{r}{r_{i+1}}\right) + T\left(r_{i+1}\right) , \quad r_i < r < r_{i+1} , \tag{12.22}$$

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. (12.22), the temperature on the inside of the volume, r_i, can be determined. The volume-averaged temperature of the volume is

$$\overline{T_i} = \frac{1}{\pi\left(r_{i+1}^2 - r_i^2\right)} \int_{r_i}^{r_{i+1}} 2\pi r T\left(r\right) dr . \tag{12.23}$$

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:

$$T(r) = \frac{q_i' r_i}{k_i} \ln\left(\frac{r_{i+1}}{r}\right) + T(r_i) \ , \quad r_i < r < r_{i+1} \ . \tag{12.24}$$

Because 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's 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 $q'(0) = 0$ because of the symmetry of the 1D equations.

12.1.2.1 Thermal properties

The thermal conductivity of the clad and fuel are modeled in units of $\frac{W}{m \cdot K}$ using semi-empirical correlations based on low order polynomials. The thermal conductivity of the clad is modeled using

$$k_{clad}(T) = 7.51 + 2.09 \cdot 10^{-2} T - 1.45 \cdot 10^{-5} T^2 + 7.67 \cdot 10^{-9} T^3 \ . \tag{12.25}$$

This correlation is recommended in NUREG/CR-6150 [46], which provides a detailed review of the experimental data justifying this correlation.

The thermal conductivity of the fuel does not account for burnup, porosity, initial plutonium loading, or gadolinium loading; instead, it is only a function of temperature:

$$k_{fuel}(T) = 1.05 + \frac{2150}{T - 73.15} \ . \tag{12.26}$$

12.1.3 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:

$$\overline{T_{fuel}} = T_{bulk} + aP_l + bP_l^2 \ , \tag{12.27a}$$

$$\overline{T_{clad}} = f_{clad} \overline{T_{fuel}} + (1 - f_{clad}) \overline{T_{fuel}} \ , \tag{12.27b}$$

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.

12.1.4 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 only 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 [127] and the fuel performance code BISON [59, 67, 156] be used for production calculations.

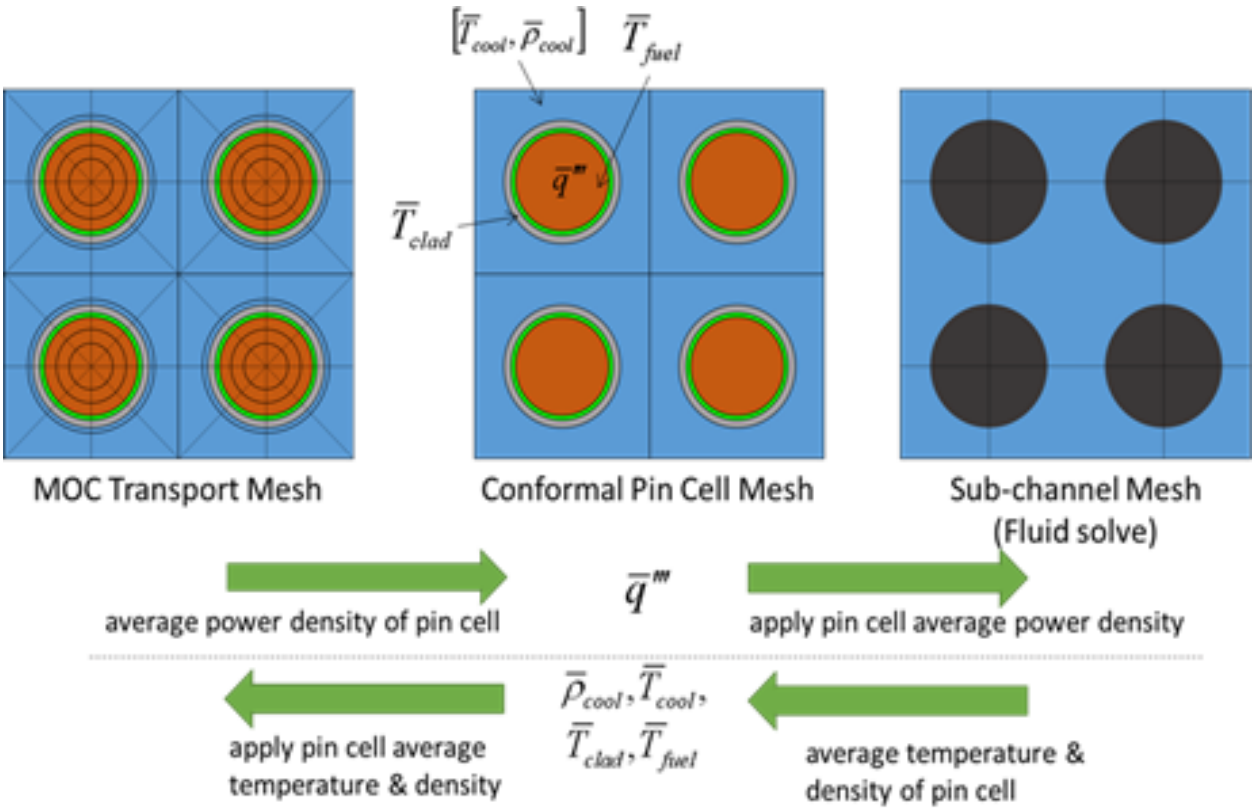


Figure 12.3. The MPACT mesh (left), CTF mesh (right), and conformal coupling mesh (center) [95].

12.2. CTF COUPLING

The other TH solver is direct coupling to the CTF thermal hydraulics code [127]. In this case, CTF is called as a subroutine through exposed coupling interfaces. This solve is more expensive that the simplified TH solver, but it contains more physics and provides more accurate solutions.

The CTF coupling consists of three basic phases: a setup phase, the solve, and a results phase. During the setup phase, solution data required for the TH solve are sent from MPACT to CTF. During the calculation phase, CTF performs the fluid calculations, and if temperature tables are not in use, then the fuel heat conduction calculations are performed. During the last phase, results of the calculation are passed from CTF back to MPACT.

Two different coupling meshes can be used. The first is a pin-based coupling mesh that is ideal for LWRs or other similar models with regular rod-based lattices. The second is a more general channel-based mesh. The channel-based approach can be used for molten salt reactors that do not have fuel rods or for BWRs with important flow regions that are not part of a fuel rod cell in the pin-based coupling.

12.2.1 Pin-Based Coupling

Figure 12.3 shows the coupling mesh in the center. MPACT calculates the linear power rate and burnup for each rod on the coupling mesh and passes that to CTF, which solves the TH problem on the subchannel mesh. Once the calculation is complete, CTF averages the solution back onto the rod-centered coupling mesh by averaging all the subchannels that surround each rod. The rod powers are calculated as follows:

$$P'_{r,z} = \frac{\sum_{r' \in (r,z)} \sum_{g=1}^{G} \kappa \Sigma_{f,r',g} \phi_{r',g}}{\overline{P}'}, \tag{12.28a}$$

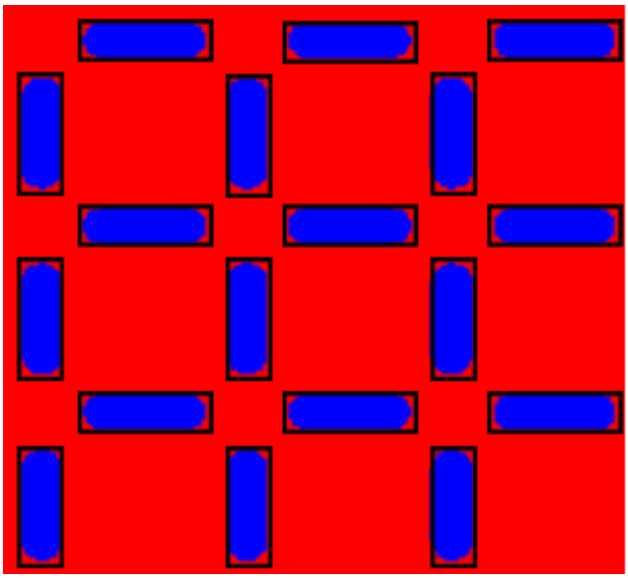


Figure 12.4. Salt channels (blue) in a graphite block (red) with overlaid CTF channels (black).

$$\overline{P}' = \frac{P_{total}}{\sum_{r \in R} \sum_{zinZ} l_{z,r}}, \tag{12.28b}$$

where subscript r, z is rod index r at elevation z, subscript r' is an MPACT mesh region index, subscript g is the energy group index, G is the total number of energy groups, R is the total number of rods, Z is the total number of axial levels, $\kappa\Sigma_f$ is the heat production from fission, ϕ is the scalar flux, $\overline{P}'$ is the nominal linear heat rate (LHR), l is the heated length at axial level z, and P_{total} is the total power for the reactor.

In addition to simply setting the rod power, MPACT also contains direct energy deposition models to account for the energy deposition of neutrons in various reactor components as they slow down. Power sources can be set for the entire rod, as described previously, or they can be set individually for the fuel, clad, and moderator materials to calculate a more precise temperature distribution.

CTF also exposes interfaces for providing a radial shape function for the power and burnup distributions in each rod. This accounts for the fact that the power generation is not uniform throughout each pellet and provides a more accurate radial power shape for the heat conduction calculations. MPACT uses multiple radial rings to mesh each fuel rod, so the average power generation in each ring can be used to calculate coefficients of a Zernike polynomial that accurately preserves the total power in each mesh while providing a continuous shape function for the CTF heat conduction calculations. The same procedure can be followed to generate a radial burnup function for each rod.

Once the calculation is complete, CTF returns temperatures for the fuel, clad, gap, moderator, clad surface, pellet surface, and guide tube. It can also return densities for the moderator surrounding the rod or filling a guide tube. These temperatures and densities are then used to modify the isotopics, and thus the cross sections, for the next MPACT neutronics calculations.

12.2.2 Channel-Based Coupling

Channel-based coupling follows a similar procedure, except that instead of mapping from MPACT rods to CTF rods, MPACT must now map all of its cross section regions to the appropriate CTF channel. To do this, CTF provides interfaces for the number of channels, the bounding box of each channel, and the axial

mesh within each channel. The bounding box is defined in terms of minimum and maximum positions in x, y, and z. The axial mesh for the channel is assumed to align with the MPACT axial mesh for parts of the channel that are modeled by both codes. If the centroid of a cross section region falls inside the bounding box, then it is mapped to a channel. Otherwise it is ignored. Regions that are not mapped to any channels will not contribute to the power calculations and will not be modified by the TH solutions returned by CTF. Figure 12.4 shows a 2D example of the bounding boxes for a portion of a model of the Molten Salt Reactor Experiment.

Once the mapping is complete, a similar but simpler procedure to the pin-based coupling is followed:

1. the volumetric power of each level of each channel is calculated by MPACT and sent to CTF,
2. the TH calculation is conducted,
3. the temperature, density, and pressure of the fluid at each level of each channel are then retrieved from CTF by MPACT to update the cross sections for the next neutronics calculation.

12.3. USER TH

MPACT also has a user TH capability. This allows the user to set up an HDF5 file with constant TH distributions specified at pin and assembly levels and allows the user to include fuel temperatures and coolant temperatures and densities. The format of the HDF5 datasets is the same as the output from coupled calculations, so the user can apply constant but realistic TH conditions to MPACT calculations without coupling to another code.

12.4. MULTISCALE COUPLING

For BWR calculations, the CTF solution can be prohibitively time consuming. This is due to two factors. First, the two-phase flow is a much more challenging TH problem that results in a stiffer numerical solution that takes many more iterations to solve than the single-phase PWR calculation. Second, during the early coupled iterations, the MPACT power distribution sent to CTF is far from reality and changing rapidly, thus causing CTF to spend even more time converging to a TH solution driven by evolving power distributions.

To mitigate these runtime issues, a multiscale TH coupline approach was developed. This approach uses the BWR STH capability during the first few iterations to provide an approximate TH solution for MPACT. Once the power distribution is partially converged from the approximate STH solution, the flow distribution calculated by STH is sent to CTF and CTF is used for the TH calculation for the remainder of the solve. This has no impact on accuracy because the calculation still ends with CTF, but dramatically improves the runtime. This procedure is illustrated in Figure 12.5 and described in greater detail in [64].

12.5. BYPASS FLOW CALCULATION

For BWR models, a significant amount of heating occurs in the bypass region due to several factors: control blade heating caused by absorption, bypass flow heating caused by down-scatter, and conduction through the channel box of heat deposited in both the coolant and channel box. The simplified TH (STH) solver described in Section 12.1.1.2 can be applied to the bypass flows as it already is for the channel to address the heating in the bypass region. A single channel is set up for each assembly's bypass flow. The geometry of the channel accounts for the detailed outer edge of the channel box—including thick-thin channel boxes—and the presence of control blades. The flow areas increase and the perimeters decrease above the tips of the control blades. The heated perimeters are set equal to the wetted perimeters.

Given these channel descriptions, the STH solver calculates axial temperature and density distributions for the bypass flow around each assembly; it also calculates axial temperature distributions for the control blades

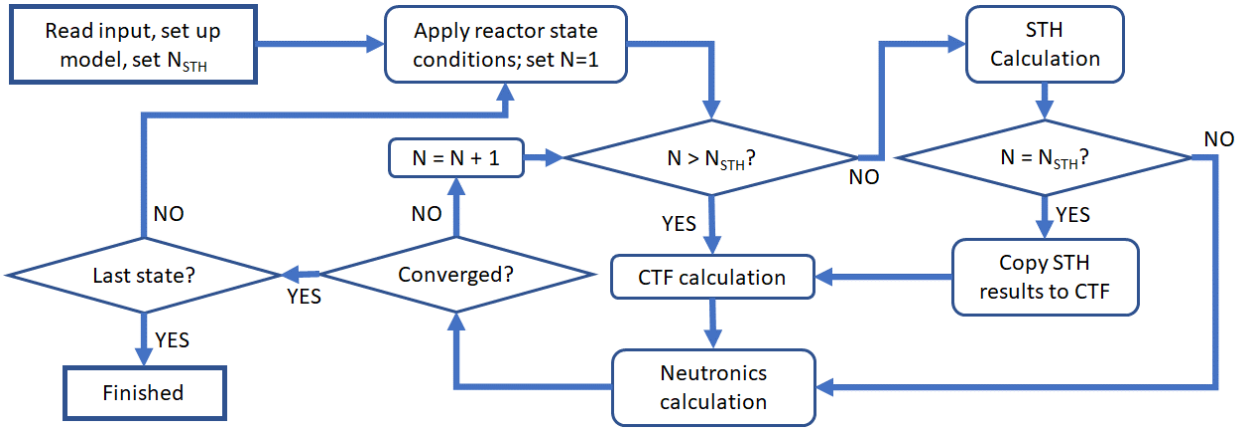


Figure 12.5. Illustration of the multiscale coupling procedure.

and channel boxes and adjusts the amount of heat in the channel from conduction through the channel box. As with the in-bundle coolant solve, the bypass flow around each assembly is rebalanced to obtain the same pressure drop in all bypass flow channels, which results in a radial flow distribution for the bypass as well. This information affects the neutronics calculation for coupled simulations.

MPACT calculates the direct heating of the channel box, bypass flow, and control blades caused by down-scatter and absorption. These calculations are performed as follows:

$$P_{box,i,z} = \int_{V\in(box,i,z)} \int_0^\infty \kappa_c \Sigma_c(\boldsymbol{x}, E)\,\phi(\boldsymbol{x}, E) + \int_0^\infty (E - E')\,\Sigma_s(\boldsymbol{x}, E \to E')\,\phi(\boldsymbol{x}, e)\,dE'dEdV\,, \tag{12.29a}$$

$$P_{bypass,i,z} = \int_{V\in(bypass,i,z)} \int_0^\infty \kappa_c \Sigma_c(\boldsymbol{x}, E)\,\phi(\boldsymbol{x}, E) + \int_0^\infty (E - E')\,\Sigma_s(\boldsymbol{x}, E \to E')\,\phi(\boldsymbol{x}, e)\,dE'dEdV\,, \tag{12.29b}$$

$$P_{blade,i,z} = \int_{V\in(blade,i,z)} \int_0^\infty \kappa_c \Sigma_c(\boldsymbol{x}, E)\,\phi(\boldsymbol{x}, E) + \int_0^\infty (E - E')\,\Sigma_s(\boldsymbol{x}, E \to E')\,\phi(\boldsymbol{x}, e)\,dE'dEdV\,, \tag{12.29c}$$

where subscripts *box*, *bypass*, and *blade* refer to quantities in the channel box, bypass flow, and control blade, respectively; subscripts i and z refer to quantities in assembly i and axial level z, respectively; κ_c is the heat release per capture; Σ_c is the capture cross section; Σ_s is the scattering cross section; and *phi* is the scalar flux. These values are then used in the STH solve for the bypass region to calculate temperatures and densities.

12.5.1 Bypass Heating Parameters

Three major parameters control the behavior of the bypass STH model. These parameters are presently user inputs to VERA, and therefore reasonable values must be chosen. Each of these values and the value selection process are discussed in the following sections.

12.5.1.1 Bypass Flow Fraction

The bypass flow fraction is the most well understood of the three parameters at this time. It is a state-dependent value that must be input by the user. For full-core models, the bypass fraction is typically input as a power- and flow-dependent table based on plant data. This allows the bypass flow to properly match the state-dependent power and flow without the user entering the bypass every time.

12.5.1.2 Bypass Heating Fraction

A significant amount of the reactor power is due to gamma rays, which are not typically modeled during multiphysics calculations with MPACT. To account for this gamma heating and other secondary effects that may not be resolved by MPACT, a second user-specified parameter is the bypass heating fraction, which is a single value for the entire calculation. This controls the fraction of additional power that is deposited into the solid structures that affect the bypass flow: the control blades and channel boxes. A reasonable value for this may be several percent.

The bypass heating fraction is a fraction of the total core power that will be added to the MPACT-calculated power distributions for the bypass. The bypass heating fraction will then be divided up as follows:

$$\tilde{P}_{box,i,z} = f_{bh} P_{total} \frac{P_{box,i,z}}{\sum_i \sum_z P_{box,i,z} + P_{blade,i,z}} , \tag{12.30a}$$

$$\tilde{P}_{blade,i,z} = f_{bh} P_{total} \frac{P_{blade,i,z}}{\sum_i \sum_z P_{box,i,z} + P_{blade,i,z}} , \tag{12.30b}$$

where f_{bh} is the user-specified bypass heating fraction and P_{total} is the total reactor power. In other words, additional user-specified heating is placed in all the channel boxes and control blades and uses the MPACT-calculated heating as a shape function to distribute the user-specified heating. The reason this heating is restricted only to the solid structures is that gamma rays primarily interact with high-Z material, so it is expected that the vast majority of gamma heating would occur in the structural materials, not the fluid. The $\tilde{P}_{box}$ and $\tilde{P}_{blade}$ values are what are actually used in the STH calculation.

12.5.1.3 Bypass Inlet Orifice Loss

The final parameter is the inlet orifice loss for the bypass flow, which is specified as an assembly-wise map. For the channel TH calculation, the inlet orifice loss coefficients may be on the order of 1.0 or less; typically, the map has one value for the peripheral assemblies and a different value for the interior assemblies. However, for the bypass flow, this parameter is about two orders of magnitude higher, and typically no data are available for the actual loss coefficient for the bypass flow. To avoid forcing the user to manually tune the distribution for the bypass flow, a search capability was implemented. If the distribution is not input, then the code will assume (1) that all assemblies have the same inlet orifice loss for the bypass flow and (2) that the pressure drop in the bypass regions must be the same as that inside the bundle. Using these two assumptions, the bypass flow solver can iterate on the inlet orifice loss until the bypass pressure drop matches the in-bundle pressure drop. This iteration is implemented as a Newton iteration, which in practice takes only about three Newton iterations to sufficiently converge. The bypass flow solver is efficient enough that the increased runtime from the additional STH solves to obtain the loss coefficient is negligible.

12.5.2 Channel Box Conduction

To account for the channel box conduction, a simple 1D slab geometry conduction calculation is performed. The conduction equations for this scenario are as follows:

$$-k \frac{d^2 T}{dx^2} = \dot{q}''' , \tag{12.31a}$$

$$h_{bypass}\left(T\left(0\right)-T_{bypass}\right)=k\left.\frac{dT}{dx}\right|_{x=0}, \tag{12.31b}$$

$$h_{coolant}\left(T\left(L\right)-T_{coolant}\right)=k\left.\frac{dT}{dx}\right|_{x=L}, \tag{12.31c}$$

where $x = 0$ is interface between the bypass flow and outer edge of the channel box, $x = L$ is the interface between the coolant flow and the inner edge of the channel box, T_{bypass} and $T_{coolant}$ are the bypass and coolant fluid temperatures from the previous TH solve, h_{bypass} and $h_{coolant}$ are the convective heat transfer coefficients for the bypass and coolant flows, k is the channel box heat conductivity, and $\dot{q}'''$ is the volumetric heat rate in the channel box. These equations do not have an analytic solution, but they can be integrated over finite volumes to set up a simple linear system to solve for the channel box temperature. Then, the heat flux from the channel box to the bypass and coolant flows is calculated and used to update the power deposition in the fluid.

12.5.3 Coupled Calculation Flow

The calculation flow for a coupled neutronics–TH solve with the bypass flow heating and inlet orifice loss coefficient search enabled is shown in Figure 12.6. The coolant TH is solved first, iterating until the pressure drop in all fuel bundles is the same. The channel box conduction occurs next, using the most recently available coolant and bypass conditions; for the first iteration, the bypass is assumed to be at inlet conditions. The heat fluxes out of either side of the channel box are used to modify the total heat deposition in the bypass flow for the upcoming solve, as well as in the coolant flow for the subsequent iteration's coolant solve. The bypass flow TH solve occurs next. As with the coolant TH solve, the solver iterates on the flow until the pressure drop in each bypass region is the same. Next, the bypass pressure drop is compared to the coolant pressure drop. If they are not equal, then the inlet orifice loss coefficients are updated, and the bypass flow TH calculation is repeated. This results in bypass flow temperature and density distributions that have a pressure drop consistent with the rest of the coolant.

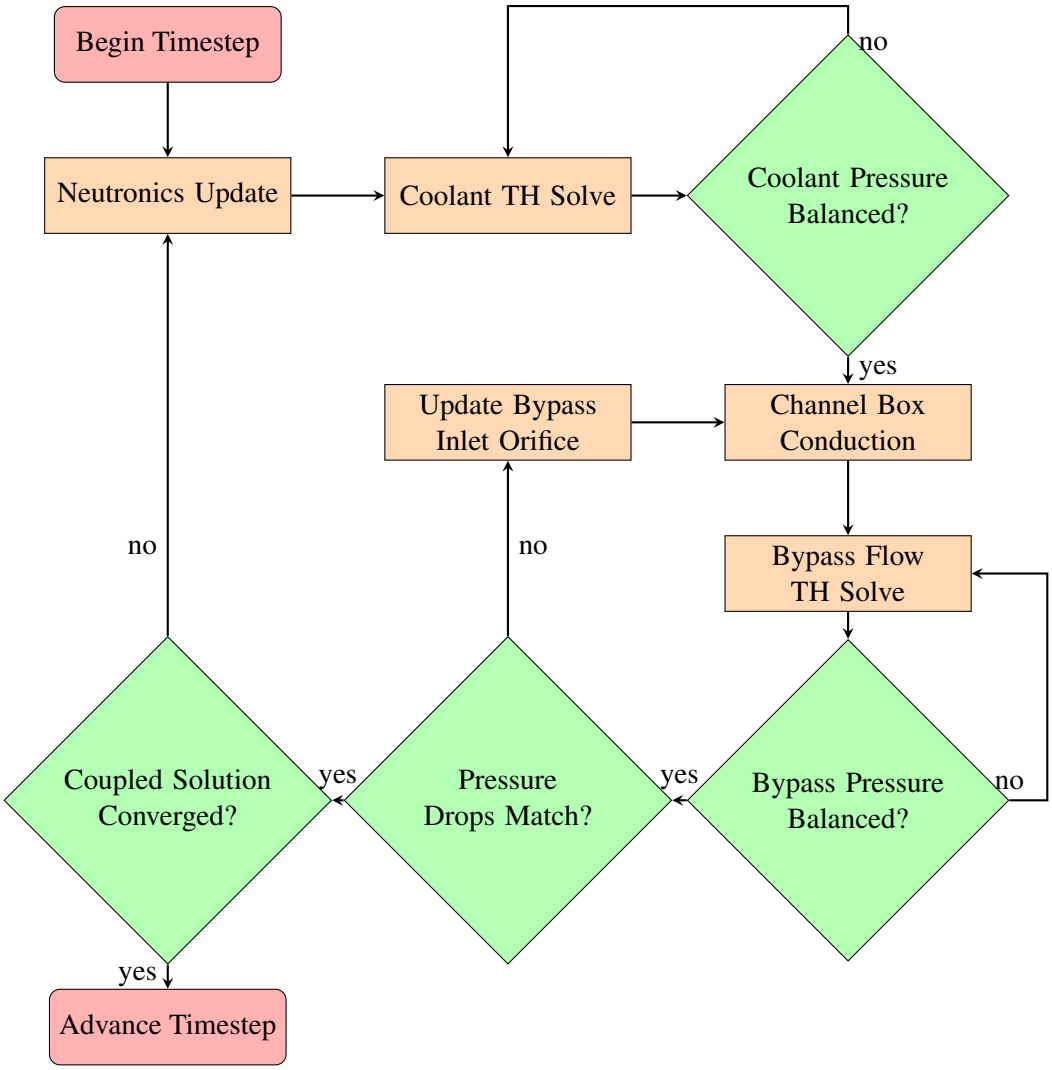


Figure 12.6. Calculation flow when using bypass heating with inlet orifice loss coefficient iteration.

12.6. NATURAL CIRCULATION

A simple natural circulation capability is available in MPACT as part of the STH module. The fundamental equation for PWR flow rate given a computed pressure drop from STH or CTF is given by

$$\Delta P_{Rx} = \frac{f_{Ch}(H_3 - H_2)\dot{m}^2}{2A_{x,Ch}^2\rho_2 D_{e,Ch}} + \frac{f_{SG}L_{SG}\dot{m}^2}{2A_{x,SG}^2\bar{\rho}_{SG}D_{e,SG}} + \frac{f_{DC}(H_1 - H_4)\dot{m}^2}{2A_{x,DC}^2\rho_1 D_{e,DC}} + \frac{K_{Ch}\dot{m}^2}{2A_{x,Ch}^2\rho_2} + \frac{K_{SG}\dot{m}^2}{2A_{x,SG}^2\bar{\rho}_{SG}}$$
$$+ \frac{K_{DC}\dot{m}^2}{2A_{x,DC}^2\rho_1} + \rho_2 g(H_3 - H_2) + \rho_1 g(H_1 - H_4) - \bar{\rho}_{SG}g(H_3 - H_4) + \frac{\dot{m}^2}{A_{x,SG}^2}\left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right), \quad (12.32)$$

where the subscripts and variables have the following meanings:

- Rx: reactor core,
- Ch: chimney,
- SG: steam generator,
- DC: downcomer,

- Integer subscripts 1, 2, 3, 4 refer to the labels in Figure 12.7a,
- H: elevation at the location indicated by the subscript,
- A_x: cross-sectional area,
- f: friction factor,
- $\dot{m}$: mass flow rate,
- K: local loss coefficients due to geometry changes (e.g., pipe branching or recombining, pipe narrowing or widening, spacer grids)
- L: length of a component of the loop,
- ρ: fluid density,
- $\overline{\rho}$: component-averged density,
- D_e: hydraulic diameter,
- g: gravitational acceleration.

The equivalent for BWRs is given by

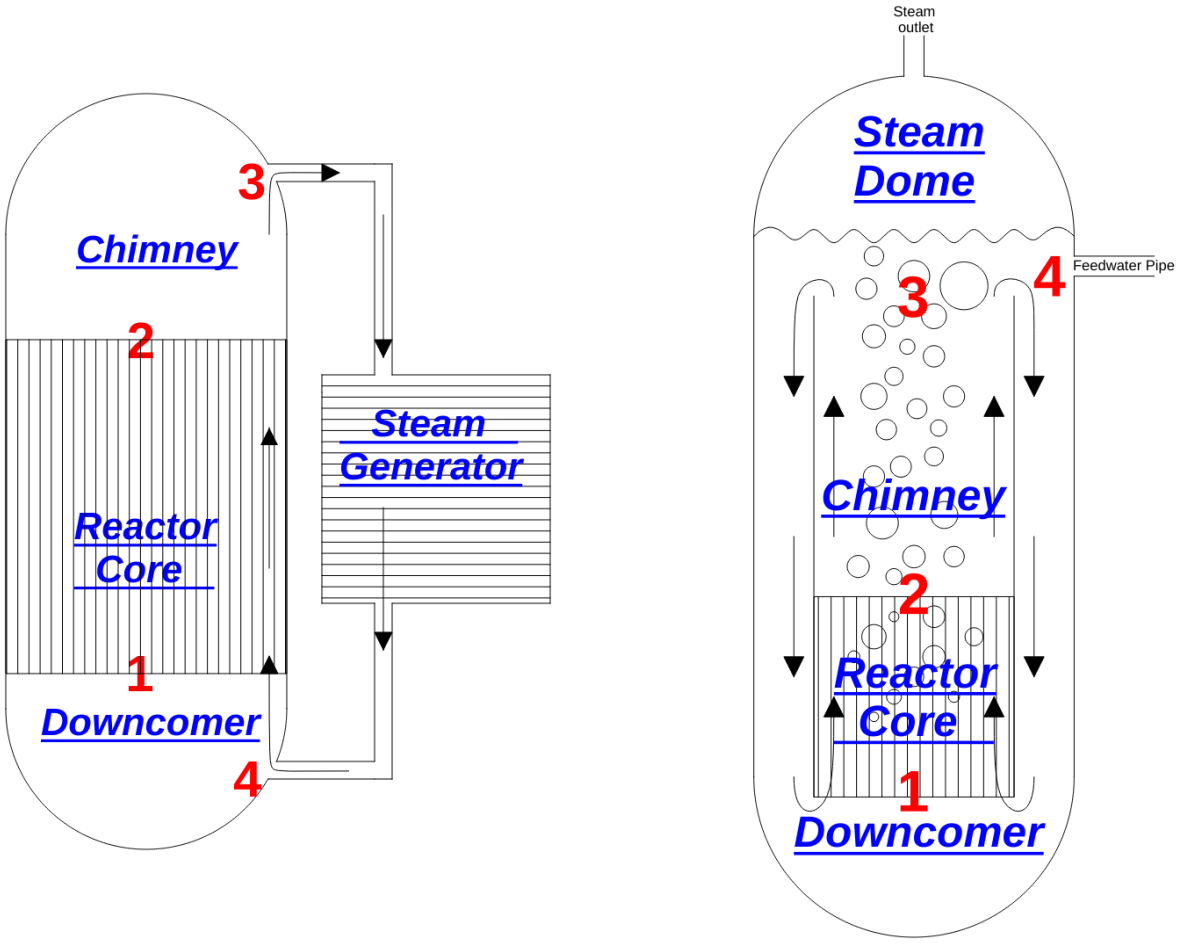
$$\Delta P_{Rx} = \frac{f_{Ch}(H_3 - H_2)\dot{m}^2}{2A_{x,Ch}^2 \rho_f D_{e,Ch}} \Phi_{lo,Ch}^2 + \frac{f_{DC}(H_3 - H_1)\dot{m}^2}{2A_{x,DC}^2 \rho_1 D_{e,DC}} + \frac{K_{Ch}\dot{m}^2}{2A_{x,Ch}^2 \rho_f} \Psi_{Ch} + \frac{K_{DC}\dot{m}^2}{2A_{x,DC}^2 \rho_1} + \rho_2 g(H_3 - H_2) - \rho_1 g(H_3 - H_1) \,, \tag{12.33}$$

where variables retain the same meanings and new variables have the following meanings:

- ρ_f: density of saturated liquid,
- ρ_g: desnity of saturated vapor,
- Φ_{lo}^2: two-phase friction multiplier,
- Ψ: two-phase local loss multiplier.

Additionally, the number and components subscripts for the BWR expression refer to Figure 12.7b instead of Figure 12.7a.

The natural circulation solver calls the in-core TH solver first, then uses one of the preceding equations to update the mass flow rate $\dot{m}_n$ for the current iteration n. The solver continues to iterate until $|\dot{m}_n - \dot{m}_{n-1}| > \dot{m}_n \epsilon$, where ϵ is convergence criterion. Additional details on the derivation, implementation, and results can be found in the paper by Herring [73].



(a) PWR **(b) BWR**

Figure 12.7. Illustration of primary loop model used for natural circulation calculations.

12.7. PARTIALLY CONVERGED CMFD FOR TH COUPLING

When using TH feedback or other feedback mechanisms, the convergence behavior of the 2D/1D calculations accelerated by CMFD can be degraded. The traditional approach to solving this problem is to introduce a relaxation factor in the CMFD calculation. While this helps to stabilize the iteration scheme, it can significantly increase the total number of coupled iterations required to solve the problem.

To address this, Fourier analysis was performed on a linearized form of the neutron transport equation with coarse mesh diffusion-based acceleration and TH feedback. This analysis showed that partially converging the CMFD calculation during the early iterations has the effect of improving the convergence of the overall system. The *nearly optimally partially converged* (NOPC) CMFD method is implemented in MPACT as an option to improve the performance of multiphysics simulations. Additional details of this method are left to the literature [133, 134]

12.8. ENERGY DEPOSITION

The solution of the Boltzmann transport equation gives neutron flux and related reaction rates. To obtain the pin power distribution, the user must calculate the energy deposition based on these reaction rates. The energy release and recovery in a nuclear reactor system involves several physical mechanisms. In addition

to the direct fission energy release, energy is also deposited by the slowing-down and capture of neutrons. Most recoverable energy is released instantaneously, such as the kinetic energy from fission fragments, prompt gamma rays, and slowing-down of fission neutrons. About 7% of energy is released some time after the reaction event from the radioactive decay of fission products or from activation materials in the form of delayed beta and gamma rays. The delayed energy reaches equilibrium after a period of steady power history. The time-dependence of the delayed energy is usually neglected in most reactor core simulations, except when estimating the decay heat during reactor shutdown. Table 12.1 presents the various energy release mechanisms from neutron fission and capture, along with the characteristics of these mechanisms. To accurately model the energy deposition regarding the spatial and time dependence, special considerations are required for the terms in boldface font.

Table 12.1. Fission energy release from the section of MF1/MT458 in ENDF/B-VII.

Mechanisms		Fraction	U-235 (MeV)	U-238 (MeV)	Range	Time
Fission fragments		80%	169.13	139.80	Local	Instant
Beta decay of fission products		3-4%	6.50	8.48	Local	**Delayed**
Fission	Prompt	3-4%	6.60	6.68	**Global**	Instant
gammas	Delayed	3-4%	6.33	8.25		**Delayed**
Anti-neutrinos		4-5%	8.75	11.39	Unrecoverable	Delayed
Neutron capture[1,2]		3-5%	**6.0-11.0**		**Global**	Instant/**delayed**
Neutron scattering[2]		2-3%	**4.0-5.0**		**Global**	Instant

1. Energy released by capture of neutron of all absorbers in the system via reactions of (n,γ), (n,α), etc.
2. The kinetic energy of fission neutrons is not included in this table because it is accounted for by neutron capture and scattering. The energy releases via neutron capture and scattering are estimated ranges that depend on the specific reactor system.

12.8.1 The Default Energy Deposition Model

Conventionally, MPACT uses the effective isotopic fission energy release for core operation calculations [86]. The effective energy release includes kinetic energy of fission fragments and fission neutrons, prompt gamma energy, equilibrium delayed gamma and beta energies from the Evaluated Nuclear Data File (ENDF), and a typical energy release contribution from neutron capture [78]. With this approach, the slowing-down energy is assumed to be equal to the kinetic energy of fission neutrons. Given a reactor's thermal power P, the neutron flux is normalized by a factor f, as determined in the following equation,

$$P(t) = f \sum_j V_j \sum_{iso} \widetilde{\kappa}_{f,iso} N_{iso,j} \sum_g \sigma_{f,g,iso,j} \phi_{g,j} \,. \tag{12.34}$$

where, $\widetilde{\kappa}_{f,iso}$ is effective fission energy release for isotope iso (unit: J), and j is the region index. When the summation of material regions applies to a pin cell, Eq. (12.34) computes the pin power. Therefore, MPACT assumes local energy deposition at fission sites inside the fuel rods. To consider the heat directly generated in the moderator via neutron and gamma reactions, a direct moderator heating fraction (user-defined) is presently used.

12.8.2 The Explicit Energy Deposition Model

Several limitations of the default energy deposition model in MPACT include (1) time dependence of the delayed energy mechanisms is not considered in the time-dependent problems like depletion and transient calculations; (2) a typical value of the neutron capture and slowing-down heat is built into fission kappa without considering the fuel composition explicitly; (3) all heat generation is assumed to be deposited in

fuel rods when computing the pin power; and (4) no gamma smearing effect is considered. To eliminate these assumptions and account for all the boldface terms in Table 12.1, an explicit energy deposition model is developed in MPACT.

12.8.2.1 Explicit model with time dependence

Accounting for the time dependence of energy deposition involves two considerations. First, in depletion, the overall effective energy release per fission varies with the isotopic concentrations via fission and capture reactions. As described below, this effect is considered in the new model because it accounts for the fission and capture energy releases explicitly. Second, the time dependence of the delayed energy mechanisms should be modeled for depletion and transient calculations. To proceed, the thermal power of a reactor, $P(t)$, is written as the sum of a prompt component, $P_{pr}(t)$, and a delayed component, $P_{de}(t)$:

$$P(t) = P_{pr}(t) + P_{de}(t) \, . \tag{12.35}$$

For a fissionable isotope, a fraction γ^f_{iso} of the fission energy is released sometime after the fission event through fission product decay. Similarly, the delayed fraction, γ^c_{iso} , of the energy for neutron capture can be defined, so $P_{pr}(t)$ is written as

$$\begin{aligned} P_{pr}(t) = f(t) \sum_j V_j \sum_{iso} N_{iso,j}(t)[(1-\gamma^f_{iso})\kappa_{f,iso} \sum_g \sigma_{f,g,iso,j}\phi_{g,j}(t) \\ + (1-\gamma^c_{iso})\kappa_{c,iso} \sum_g \sigma_{c,g,iso,j}\phi_{g,j}(t) + \sum_g \sum_{g'} \sigma_{s,g'\rightarrow g,iso,j}\phi_{g',j}(t)\Delta E(g' \rightarrow g)] \, . \end{aligned} \tag{12.36}$$

In Eq. (12.36), the energy release from neutron fission, capture, and slowing down are explicitly modeled; this corresponds to the three terms in brackets. $\kappa_{f,iso}$ is the direct fission kappa without the contribution from fission neutrons (and the subsequent energy from neutron capture and slowing down). $\kappa_{c,iso}$ is the average energy release from neutron capture of isotope iso. $\Delta E(g' \rightarrow g)$ is the average energy loss of neutrons scattered from group g' to g calculated by the midpoints of the two energy groups. Because the inelastic scattering effect is small for thermal systems, all the slowing-down heat is assumed to be promptly released.

To determine the delayed power $P_{de}(t)$, a common approach is to lump the numerous components of the delayed heat source into a relatively small number of groups [49] [125]. This approach is similar to the treatment of delayed neutron precursors, but in this case, the group is characterized by the half-lives of delayed heat emitters. The constants for the delayed heat groups are obtained by exponentially fitting the delayed energy release as a function of emission time. This approach was implemented in MPACT to investigate the effect of the delayed energy release on the time-dependent calculations [108]. If $D^x_{m,iso,j}$ is defined as the concentration of the delayed heat emitters in group m from isotope iso at region j for induced reaction x (fission or capture), then the balance equation that $D^x_{m,iso,j}$ satisfies is given as

$$\frac{dD^x_{m,iso,j}(t)}{dt} = f(t)V_j\gamma^x_{m,iso}\kappa_{x,iso}N_{iso,j}(t)\sum_g \sigma_{x,g,iso,j}\phi_{g,j}(t) - \lambda^x_{m,iso}D^x_{m,iso,j}(t) \, , \tag{12.37}$$

where $\gamma^x_{m,iso}$ and $\lambda^x_{m,iso}$ are the yield and decay constants of delayed heat group m, so that $\sum_m \gamma^x_{m,iso}$ is equal to γ^x_{iso} in Eq. (12.36). By using $D^x_{m,iso,j}$, the delayed power is calculated as

$$P_{de}(t) = \sum_j \sum_{iso} \sum_{x=f,c} \sum_m \lambda^x_{m,iso}D^x_{m,iso,j}(t) \, . \tag{12.38}$$

To account for the time dependence of delayed energy, MPACT can solve Eqs. (12.35)- (12.38) instead of (12.34).

12.8.2.2 Spatially dependent energy deposition with gamma smearing

Most fission energy is deposited locally in fuel rods in which fission events occur. However, a small but nontrivial fraction of the energy can also be deposited in the nonfuel regions, such as the clad, moderator, control rods, and structural materials. By applying Eqs. (12.36) and (12.38) to a spatial cell (i.e., the summation over the material regions of the cell), the local heat can be estimated explicitly.

The spatial model should also account for the energy deposition of gamma rays, essentially from neutron fission and capture for thermal systems. Liu et al. [111] developed an approximate gamma smearing model based on the following rationale: (1) the spatial gamma energy sources can be estimated according to the fission and capture rates, (2) the gamma energy originated from a pin cell is assumed to be redistributed within a predefined block of $n \times n$ pin cells due to the longer mean free path, (3) pseudo fuel pins with a zero gamma source are placed in the radial reflector (almost transparent to gamma) to account for the leakage effect, and (4) according to Knoll [92], the gamma interaction cross sections are essentially proportional to the power of the isotope atomic number. Considering gamma smearing, the final pin power $\widetilde{P}_{i,j}$ is calculated as

$$\widetilde{P}_{i,j} = P_{i,j} - P^{\gamma}_{i,j} + \sum_{k=1}^{K}\sum_{m=1}^{M} P^{\gamma}_{km\to ij} \,, \tag{12.39}$$

where $P_{i,j}$ is the initial pin power before gamma smearing at the x and y indices (i, j); $P^{\gamma}_{i,j}$ is the gamma source energy originated from pin (i, j), including prompt and delayed gamma from fission and capture; K and M are the total numbers of pins on x and y side of a 2D plane (pseudo pins can be filled for the jagged core such that all rows or columns have the same number of pins); and $P^{\gamma}_{km\to ij}$ is the gamma energy deposition in pin (i, j) due to the gamma source from pin (k, m). Based on the assumption that gamma smearing is within a predefined $n \times n$ (an odd number should be chosen for n to ensure that the gamma source pin is in the center of the block), $P^{\gamma}_{km\to ij}$ is calculated as

$$P^{\gamma}_{km\to ij} = P^{\gamma}_{k,m} \frac{\sum_r \sum_{iso} Z^{l}_{r,iso,i,j} N_{r,iso,i,j} V_{r,i,j}}{\sum_{p=k-x}^{k+x} \sum_{m=q-x}^{q+x} \sum_r \sum_{iso} Z^{l}_{r,iso,p,q} N_{r,iso,p,q} V_{r,p,q}} \,, \tag{12.40}$$

$$\text{if } k - \frac{n-1}{2} \le i \le k + \frac{n-1}{2} \text{ and } m - \frac{n-1}{2} \le j \le m + \frac{n-1}{2} \,, \tag{12.41}$$

where Z is the atomic number of isotope iso in subregion r of pin (i, j), N is the number density of the isotope, and V is the volume of the subregion. For values of k, m, i, and j that do not meet the above conditions, $P^{\gamma}_{km\to ij} = 0$. Figure 12.8 shows the gamma smearing ranges for three pin cells: an inside fuel rod (A), a control rod (B) near the reflective boundary, and a fuel rod (C) near the reflector. For the reflective boundary, the pin cells are unfolded according to the symmetry within the block.

If Eq. (12.39) is summed over all pins of the 2D plane, including the pseudo pins, then

$$\widetilde{P}_{tot} = P_{tot} - P^{\gamma}_{tot} + \sum_{i=1}^{K}\sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{m=1}^{M} P^{\gamma}_{km\to ij} = P_{tot} \,. \tag{12.42}$$

P_{tot} and P^{r}_{tot} are the total power and total gamma power of the 2D plane. The summation term is also the total gamma power, so the gamma smearing scheme preserves the total power.

The coefficients l and n in Eq. (12.40) are determined by trial and error according to the MCNP gamma transport solutions of a set of 2D assembly problems [153]. It was found that $l = 1.6$ and $n = 7$ resulted in good agreement for a range of 2D assembly problems. Also, the Z-dependence of the photoelectric, Compton, and pair production cross sections are in the form of $Z^4 - Z^5$, Z, and Z^2, respectively [92], so $Z^{1.6}$

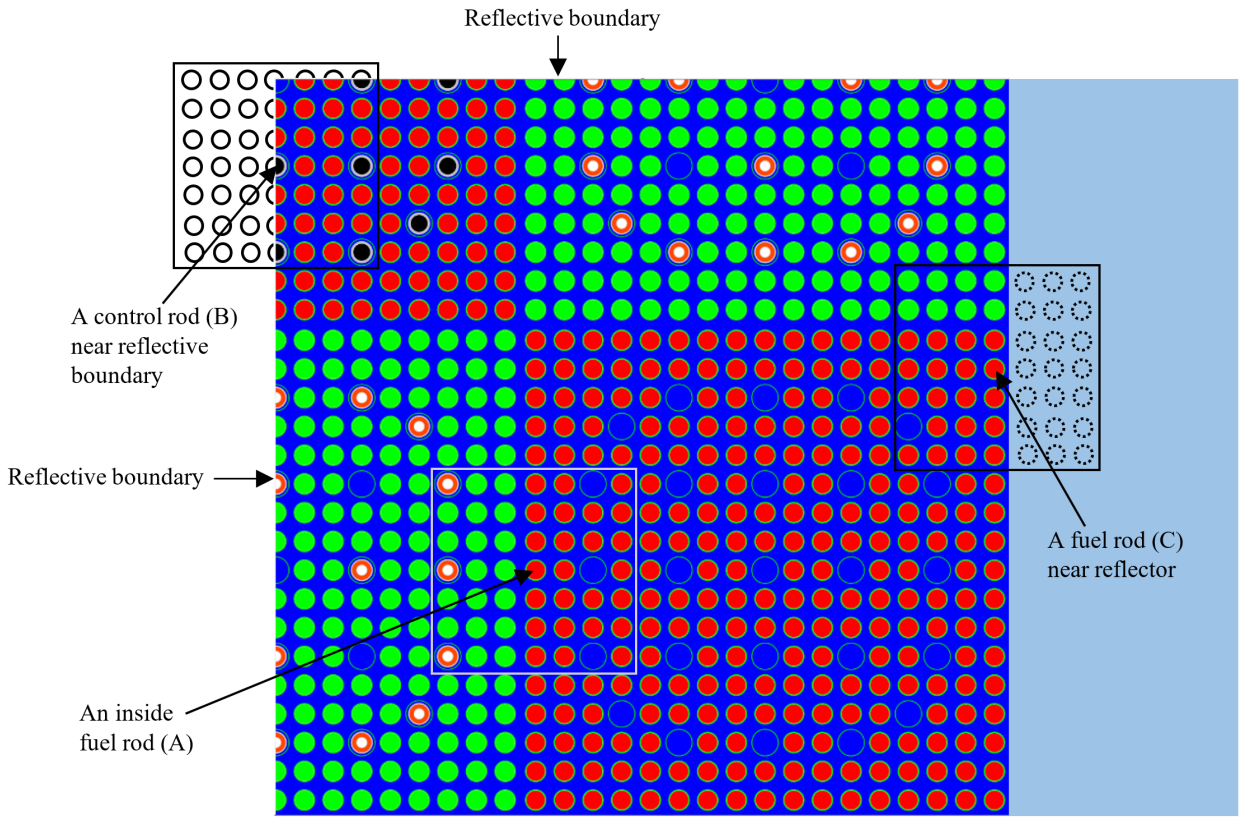


Figure 12.8. Visualization of the gamma smearing ranges with $n = 7$.

is a reasonable approximation because Compton scattering is the dominating gamma reaction in the reactor. Furthermore, only the 2D indices (i,j) are involved in Eq. (12.40), so the axial gamma smearing effect is neglected. When estimating the gamma power distribution within a pin cell, the weighting function of the same form is calculated for each material region.

12.9. DECAY HEAT COUPLING

The ORIGEN application programming interface (API) provides a straightforward function that can be called to calculate decay heat. The function simply takes in a list of isotopic number densities and returns the total heat generation in watts for that set of isotopes. All of the various decay mechanisms and heat generated by each decay are handled internally by ORIGEN. MPACT was modified to call this ORIGEN interface for every mesh region to obtain a 3D decay heat distribution. These values are then converted to decay heat LHRs to match the format of the prompt power production used for the coupling with CTF. The decay heat values are updated at the start of a calculation and after each call to ORIGEN so that the decay heat generation always matches the current isotopics in the MPACT model.

The MPACT-CTF coupling incorporates the decay heat distribution. The distinction between decay and prompt heat is made only on the MPACT side; ultimately, CTF receives only total heat from MPACT. First, the total power is decomposed into prompt and decay power:

$$P_{prompt} = P_{total} - P_{decay} \,, \tag{12.43a}$$

$$P_{decay} = \sum_{r \in R} \sum_{z \in Z} P'_{D,r,z} l_{r,z} \,, \tag{12.43b}$$

where P_{decay} and P_{prompt} are the decay heat and prompt portions of P_{total}, and $P'_{D,r,z}$ is the LHR due to decay heat at axial level z of rod r as calculated by ORIGEN. Using these definitions, we can devise a modified version of Eq. (12.28):

$$P'_{r,z} = P'_{P,r,z} + P'_{D,r,z} \,, \tag{12.44a}$$

$$P'_{P,r,z} = \frac{\sum_{r' \in (r,z)} \sum_{g=1}^{G} \kappa \Sigma_{f,r',g} \phi_{r',g}}{\overline{P_P}'} \,, \tag{12.44b}$$

$$\overline{P_P}' = \frac{P_{prompt}}{\sum_{r \in R} \sum_{zinZ} l_{z,r}} \,. \tag{12.44c}$$

These modifications allow the MPACT-CTF coupling to use the same subroutines in the CTF coupling API while separating out the prompt and delayed portions of the pin power distribution on the MPACT side of the coupling.

12.10. SPECIES TRANSPORT COUPLING

This chapter discusses mass transport models for molten salt reactor (MSR) calculations. The first model is a simplified neutron precursor drift model [31]. This assumes only axial flow, allowing a simple equation for the drift of neutron precursors to be derived as a function of fuel flow velocity and the length of the primary loop external to the core. The second model is a generalized species transport module in CTF. These calculations use the CTF TH solution to transport arbitrary species around the loop [66].

12.10.1 Simplified Neutron Precursor Drift

The standard precursor balance equation is given by

$$\frac{dC_k}{dt} = \beta_k \int \nu \Sigma_f \phi dE - \lambda_k C_k \,, \tag{12.45}$$

where C is the precursor concentration, β is the delayed neutron fission yield, λ is the precursor decay constant, and subscript k denotes the precursor group index, where the standard 6 precursor groups are used. For flowing fuel, the precursors migrate away from their initial location prior to decaying, an effect that cannot be ignored. To account for this, a convection term is added to the previous equation:

$$\frac{dC_k}{dt} = \beta_k \int \nu \Sigma_f \phi dE - \lambda_k C_k - \boldsymbol{\nabla} \cdot (C_k \boldsymbol{v}) \,, \tag{12.46}$$

where v is the fluid fuel velocity vector. This gives the distribution of the precursors, even when the fuel is moving.

The simplified model in MPACT assumes steady-state axial drift. This simplifies the equation to

$$\frac{d(C_k v_z)}{dz} + \lambda_k C_k = \beta_k \int \nu \Sigma_f \phi dE \,, \tag{12.47}$$

where v_z is the axial component of the fluid fuel velocity vector. Additionally, the axial velocity is assumed constant, specified by user input. With this assumption, the previous equation can be integrated axially across a mesh cell to obtain the precursor concentration in the cell as a function of axial position:

$$C_k(z) = C_k(0)\, e^{-\frac{\lambda_k}{v_z} z} + \frac{\beta_k}{\lambda_k} \Psi_T \left(1 - e^{-\frac{\lambda_k}{v_z} z}\right) \,, \tag{12.48}$$

where Ψ_T is the total fission source. The average precursor concentration can be calculated by integrating over a cell of height H:

$$\overline{C}_k = \frac{\left(C_k\left(0\right) - C_k\left(H\right)\right) v_z}{\lambda_k H} + \frac{\beta_k \Psi_T}{\lambda_k} \,. \tag{12.49}$$

Using this formulation, the average and outgoing precursor concentrations can be found in each mesh cell by starting at the core inlet and marching upward across each axial level for each flow channel. The outgoing precursor concentration for each level can be used as the incoming for the level above it. All that remains is to determine the core inlet concentration $C_{k,core,in}$. To do this, the outlet concentration $C_{k,core,out}$ is calculated as the average of outgoing concentrations of all the cells at the top of the core. The inlet concentration can then be calculated from the outlet concentration using the length of the primary loop external to the core L_{ext}:

$$C_{k,core,in} = C_{k,core,out} e^{-\frac{\lambda_k}{v_z} L_{ext}} \,. \tag{12.50}$$

The first time this calculation is done, a guess is used for $C_{k,core,in}$ to update the concentrations in each cell and obtain a better estimate of $C_{k,core,in}$. This process is then repeated for each of the precursor groups until the change in the inlet concentration is small.

The delayed fission source in a cell is defined in terms of the precursor concentrations:

$$\Psi_{delayed} = \sum_{k=1}^{6} \lambda_k \overline{C}_k \,. \tag{12.51}$$

The fission source contribution to each energy group g can then be defined as

$$\Psi_g = (1-\beta)\chi_g \Psi_T + \chi_{d,g} \sum_{k=1}^{6} \lambda_k \overline{C}_k \,, \tag{12.52}$$

where β is the sum of β_k, and χ_g and $\chi_{d,g}$ are the prompt and delayed fission spectra, respectively.

For CMFD acceleration, the fission source must be defined such that there is no inhomogeneous source. To accomplish this, the delayed fission source is divided by the fission source from the previous iteration:

$$\Psi_g = \left((1-\beta)\chi_g \Psi_T + \chi_{d,g} \frac{\sum_{k=1}^{6} \lambda_k \overline{C}_k}{\Psi_{T,old}} \right) \Psi_T \,. \tag{12.53}$$

At convergence, $\Psi_{T,old} = \Psi_T$, returning the typical definition of the fission source.

12.10.2 CTF Species Transport

The species transport coupling with CTF is built on the channel-based coupling from the TH coupling section. MPACT generates source terms and sends them to CTF, which uses the velocity fields to transport the species through the model. The results are then sent back to MPACT to modify the neutron transport mesh appropriately.

First, the user inputs to the code are briefly discussed, as they are useful in describing the coupling between MPACT and CTF. Then, each of five different types of species are described, along with their feedback effects on the MPACT calculation.

12.10.2.1 User inputs

Several user inputs warrant discussion to enlighten the remainder of the species transport coupling discussion. First, the MPACT input exposes options to name certain CTF cells using the "component" option. This allows CTF cells relevant to the species transport calculation to be referenced in other input options. Example names that are used in this work are "bubbler" and "offgas," which identify portions of the loop where helium gas bubbles are inserted and removed, respectively.

Two other options that are noteworthy are the "source" and "sink" options. These allow the user to specify constant source or sink terms at various parts of the loop. These options take a component name, a value, and units. The units can be either a mass insertion or removal rate in $\frac{kg}{s}$ or a percent. This allows, for example, $1 \times 10^{-6} \frac{kg}{s}$ of helium gas to be injected at the bubbler component, or 50% of all gas bubbles to be removed at the location in the loop corresponding to the off-gas system. These inputs can also be used to specify online processing of the fuel salt, online refueling, or other phenomena that are unique to MSRs and relevant to the species transport calculations and coupling.

Finally, there are several input options related to species interacting with gas bubbles. First is the list of interacting species. This list tells MPACT which isotopes can move between the salt and gas bubbles. It tells MPACT to generate two versions of that species and to create a transition between them in CTF. Second, a list of Henry's Law coefficients must be supplied for each species to govern the transitions between the two phases. MPACT does not use these coefficients, but because it is driving the calculation and setting up the species, it must pass the coefficients to CTF.

12.10.2.2 Delayed neutron precursors

The first category of species is delayed neutron precursors, an important subset of fission products that emit neutrons as a result of radioactive decay. Traditionally, and in MPACT, the precursors are divided into six groups based on half-life, with values ranging from approximately 0.2 to approximately 55 seconds. This timescale is significantly longer than that of prompt fission neutrons. The timescale is not an issue in LWRs, in which the fuel is solid, but in MSRs, it gives the precursors time to drift from their point of origin before decaying. This causes shifts in the shape of the power source, and it even causes production of power outside the active core region in the primary loop.

Two pieces of information must be calculated to transport the precursors. First is the production rate from the fission of each precursor group, defined by $S_{F,i}$:

$$S_{F,i} = \beta_i \int_0^\infty \nu\Sigma_f(\boldsymbol{x}, E, t)\, \phi(\boldsymbol{x}, E, t)\, dE\,, \tag{12.54}$$

where subscript i indicates the precursor group, β_i is the delayed neutron yield (the fraction of fission neutrons that are delayed) for the group, ν is the average number of neutrons produced per fission, Σ_f is the fission cross section, and ϕ is the scalar flux.

The other piece of information is the decay constant for each group. This constant is provided as part of MPACT's cross section library and is the same for every region. Thus, the decay constants are simply passed directly to CTF for each region without modification.

The delayed neutron precursors are applied by modifying the fission source. For a steady-state calculation, the steady-state fission source F_r in region r is typically written as follows:

$$F_r^s = \frac{k_{eff}}{4\pi} \chi_r(E)\, F_{t,r}\,, \tag{12.55}$$

where k_{eff} is the system criticality, χ_r is the energy spectrum of fission neutrons, and $F_{t,r}$ is the total fission source in region r:

$$F_{t,r} = \int_0^\infty \nu\Sigma_{f,r}(E)\,\phi_r(E)\,dE\ . \tag{12.56}$$

This expression assumes that the amounts and positions of delayed neutron precursors are unchanging, allowing the delayed neutrons to be lumped in with the prompt neutrons in calculating the fission source. Because the assumption of unmoved precursors does not apply in MSRs, even during steady-state operation, the fission source must be written in a different form that allows the precursor concentrations to directly impact the fission source distribution:

$$\begin{aligned} F_r &= \frac{k_{eff}}{4\pi}\chi_r(E)\left(\int_0^\infty\left(1-\sum_{i\in N_d}\beta_{i,r}\right)\nu\Sigma_{f,r}(E)\,\phi_r(E)\,dE + \sum_{i\in N_d}\lambda_i C_i\right) \\ &= \left(1-\sum_{i\in N_d}\beta_{i,r}\right)F_r^s + \frac{k_{eff}}{4\pi}\chi_r(E)\sum_{i\in N_d}\lambda_i C_i \\ &= \left(1-\sum_{i\in N_d}\left(\beta_{i,r}+\frac{\lambda_i C_i}{F_{t,r}}\right)\right)F_r^s \\ &= \alpha_r F_r^s\ , \end{aligned} \tag{12.57}$$

where N_d is the number of delayed neutron groups, usually six, and C_i is the delayed neutron precursor concentration in region r obtained from the CTF species transport calculation. Instead of explicitly treating the prompt and delayed portions of the fission source separately, a scaling factor α_r can be defined for each region. This scaling factor subtracts the steady-state amounts of each delayed neutron group fission source, and then it adds the actual source back in after accounting for the movement of the precursors prior to decay. When $\alpha_r = 1$, the precursors have not moved, so the expression is equivalent to the steady-state fission source. For $\alpha_r < 1$, more precursors have flowed out of r than into it, whereas the opposite is true for $\alpha_r > 1$. Thus, applying the feedback of the delayed neutron precursor drift is as simple as calculating that scaling factor for each region.

For transient calculations, such a modification to the fission source is unnecessary. Instead, the prompt and delayed portions of the fission source are calculated explicitly. The prompt fission source is calculated from the flux and fission cross section:

$$F_{p,r} = \int_0^\infty \nu_p\Sigma_{f,r}(E)\,\phi_r(E)\,dE\ , \tag{12.58}$$

whereas the delayed portion is calculated from the precursor concentrations:

$$F_{d,r} = \sum_{i\in N_d}\lambda_i C_i\ , \tag{12.59}$$

where subscripts p and d refer to *prompt* and *delayed*, respectively. The precursor concentrations in this case are taken from the transient species transport calculation for the timestep. Currently, an explicit time integration method is applied. Future work could apply more advanced time-stepping techniques.

12.10.2.3 Isotopes

The second category of supported species in MPACT is isotopes. For isotope transport, users can specify any isotope in the cross section library and have CTF transport it around the loop. Isotopes make use of the fission source yield and decay constant preparation just like precursors, but they also require some additional setup. One quantity that is required by isotopes is an absorption rate, which takes into account the fact that some isotopes are removed from the system as a result of nuclear interactions in the presence of a neutron flux. It can be calculated as follows:

$$L_{a,j} = \int_0^\infty \sigma_{a,j}(\boldsymbol{x}, E, t)\, \phi(\boldsymbol{x}, E, t)\, dE \,, \tag{12.60}$$

where $L_{a,j}$ is the loss due to absorption of isotope j and $\sigma_{a,j}$ is the microscopic absorption cross section for the isotope. This term is then included in the species transport matrix as a coefficient for j components of the solution.

Another piece of information unique to isotope transport is production of an isotope from another isotope. When multiple isotopes are being transported and one decays to the other, the loss of the parent isotope from radioactive decay is treated as a source term for the daughter isotope.

Finally, isotopes are assumed to be evenly distributed around the loop at the beginning of a calculation. This choice was made because it is important to capture isotopes flowing into the active core region from other parts of the primary loop. The code essentially assumes "fresh fuel" everywhere in the primary loop at the start of the calculation, resulting in a uniform distribution of all isotopes. After the first species transport calculation, this assumption is no longer made since a distribution is now available from CTF.

Applying the feedback effects for isotopes is much more straightforward than applying the precursor effects. For each region in the MPACT mesh, the concentration of each isotope in the corresponding CTF cell is used to update the number density on the MPACT mesh by simply assigning the new value. This is done for every isotope tracked by CTF. For those not tracked by CTF, the number densities are unchanged from the previous MPACT calculation. After updating the isotope number densities, MPACT calculates macroscopic cross sections as usual. The updated isotopic densities impact the cross sections, thus providing feedback to the next neutron transport calculation.

12.10.2.4 Elements

The third category of species is elements. Elements in species transport are a collection of individual isotopes that are exposed to the user as a convenient shorthand. When a user specifies an element, all isotopes of that element that exist in the MPACT cross section library are prepared and sent to CTF. The only unique preparation for elements, compared with isotopes, is the unfolding of the element into its constituent isotopes based on the cross section library.

12.10.2.5 Chemical species

A fourth type of species is the chemical species. These are other generic species of interest that are not directly produced as a result of the neutron transport and depletion calculations and provide no feedback, but whose movement around the loop could be of some other interest to a user.

12.10.2.6 Gas bubbles

Finally, gas bubbles can also be transported. Some MSR models inject an inert gas such as helium into the salt to aid the uptake of gaseous fission products from the salt, enabling easier removal of those fission

products via an off-gas system. Like isotopes, gas bubbles also assume an initial distribution that is uniform around the loop. This distribution is taken from the input options for the bubble or a similar component.

The only other piece of data that must be supplied to CTF is a list of interacting species. This list tells CTF which species could exist in either the liquid or the gas phase, allowing them to leave the fuel salt and enter the gas bubbles. This mechanism is governed by Henry's Law coefficients, mentioned above, but it requires MPACT to set up two species for an isotope and then inform CTF that transitions can occur between those two species based on Henry's Law behavior.

For gas bubble transport, it is assumed that the MPACT mesh was specified without accounting for the gas bubbles. After the species transport solution, the void fraction v_r from gas bubbles can be retrieved from CTF for each region in the MPACT mesh. To apply this void, the number densities of all non-gas isotopes in the MPACT mesh region are multiplied by a scaling factor $1 - v_r$. The appropriate amount of gaseous material is then added to the mesh, including any isotopes that may have moved from the salt to the gas bubbles. As with the isotope updates, the new number densities then provide feedback into the macroscopic cross section calculations in MPACT.

12.11. CHEMISTRY COUPLING

12.11.1 MAMBA Coupling

Under certain coolant chemistry and power conditions, LWRs exhibit *CRUD-induced power shift* (CIPS). CRUD is a build-up of entrained coolant particulate impurities, mostly Ni-Fe-Oxides, on the surface of the fuel. This layer is porous in nature. CRUD preferentially forms in the upper spans of the reactor where coolant temperatures are largest and where subcooled nucleate boiling occurs on the rod surfaces. The porous layer contains micrometer-sized boiling chimneys which drive an advection process that draws coolant impurities into the CRUD layer. Because the coolant contains soluble boron and lithium, the wicking effect can locally concentrate soluble boron and lithium such that solid boron-containing compounds precipitate in the CRUD layer, thus trapping boron near the rod's surface. If enough of these boron-rich deposits form in the upper spans of the core, a depression in power results in the upper spans of the reactor, leading to a power shift toward the bottom of the core, thus yielding CIPS.

The accurate prediction of CIPS has remained an important challenge problem identified by the nuclear industry. MAMBA [32] coupled with MPACT and CTF provides a capability to simulate CRUD growth with pin-by-pin fidelity, which is important for the prediction of localized boron precipitation in CRUD and for estimating the risk of *CRUD-induced local corrosion* (CILC). The difficulty of the problem stems from its multiphysics nature: correct simulation of CIPS requires coupling neutronics, thermal hydraulics, CRUD deposition, solid boron compound precipitation in the CRUD layer, and the buildup of CRUD precursor impurities in the primary loop from corrosion processes.

Several important neutronic-CRUD feedbacks are considered by coupling MPACT and MAMBA. First, the spatial fission distribution computed by MPACT is used in the larger core TH energy balance to resolve the rod surface heat flux and temperature. These surface conditions are passed to MAMBA and used in conjunction with other coolant properties to determine the CRUD growth rate. Critical to the prediction of CIPS, the power feedback effects caused by the presence of boron inside the CRUD are captured. This is handled by having MAMBA provide the CRUD thickness to MPACT, along with the number densities of B10 and other isotopes in the CRUD layer, such that the neutronic effects of the thin boron-rich CRUD layer can be captured. This feedback is resolved in the fixed-point iteration loop, where for a given operating condition and time step size, a steady-state neutronic, TH, and CRUD solution is found. Additionally, the depletion of B10 in the CRUD layer is also resolved. MPACT supplies a microscopic reaction rate of the B10 n-alpha reaction in the CRUD layer. This information is used in MAMBA to compute the amount of

B10 in the CRUD. Toward the end of cycle, the B10 depletion in the CRUD layer tends to shift the power distribution toward the top of reactor for cases in which heavy CIPS was encountered mid-cycle. Finally, the ability to clean CRUD off of the fuel rod surfaces during an outage and fuel reshuffle is provided.

Overall, the net effects of CRUD buildup on the power distribution and fuel depletion rates can be significant over the course of a cycle, so the coupling of MPACT and MAMBA provides an important capability when assessing a particular core design's CIPS risk over multiple cycles with fuel reshuffling and crud cleaning.

12.11.2 User CRUD

In addition to coupling with MAMBA, MPACT also supports a user-specified CRUD capability, similar to the user TH capabilities discussed above. The user can set up an HDF5 file with pin and assembly distributions of CRUD and boron that are applied to the MPACT mesh. This allows the user to perform analysis on the effects of CRUD (such as CIPS) without coupling to another code.

12.11.3 Thermochimica Coupling

One problem unique to MSRs lies in the chemistry of the reactor. A fresh fuel salt may contain a dozen elements, and the system components have even more, all at high temperature. Additionally, once fission begins occurring, almost the entire periodic table is available in this molten solution. Therefore, it is important to characterize the chemistry of the system, in addition to the neutronics and TH. Such analysis is important to determine whether system components will corrode, whether certain phases will plate out on the heat exchanger, which compounds could leave the liquid phase, and other potentially important phenomena.

To broach this type of analysis, the *Thermochimica* [120, 121] thermochemistry code has been coupled to MPACT and CTF. A detailed discussion of the Thermochimica methods is deferred to the referenced literature, but this section presents a brief overview of the inputs and outputs of Thermochimica in the context of MSR multiphysics simulations. The coupling with MPACT and CTF is described.

12.11.3.1 Thermochimica overview

Thermochimica takes temperature, pressure, and elemental compositions as inputs to a CALculation of PHAse Diagrams (CALPHAD) calculation using a Gibbs energy minimization (GEM) technique. This calculation determines exactly which chemical compounds and phases exist in those conditions, as well as the chemical potentials for each compound and element. Thermochimica calculations are all equilibrium point calculations, so they are limited in capturing rapid changes to the system state. However, at typical operating temperatures, it is expected that chemical equilibrium will be achieved quickly and that steady-state analysis is useful for identifying potential issues in certain system components.

To inform the Thermochimica calculations, a database must be provided. This database contains energetic information about possible states of the system: possible phases, species within each phase, and equations to calculate the Gibbs energy of those species. The GEM calculations then make use of the data to determine which phases and species are present in the system and in what proportions. For this work, the Molten Salt Thermodynamic Database (MSTDB) [12] was used: specifically, the fluoride salt database file from December 10, 2019. This database has been developed—and is still being added to—for analysis of both fluoride- and chloride-based MSR fuel salts.

12.11.3.2 Thermochimica coupling

Because Thermochimica currently provides no feedback to the MPACT calculations, the Thermochimica coupling has been implemented as a post-solution operation instead of as a feedback operation. This means that at the end of each state point, after the neutronics, TH, and species transport have all converged, a single Thermochimica calculation is performed using the converged coupled solution.

For each cell in the CTF solution, MPACT gathers the volume-averaged temperature, pressure, and elemental compositions and sends them to Thermochimica. Thermochimica performs its GEM calculations to determine the phases that exist at that state. MPACT then retrieves data of interest: the total chemical potential of the system, the chemical potential of each element in the system, and concentrations of each solution and pure condensed phase in the system. This information is then written to the MPACT output file.

Databases such as MSTDB often contain a limited number of elements. An MSR will automatically contain the vast majority of the periodic table as results of fission and neutron absorption; therefore, each element that is not present in the database but is present in the reactor in sufficient quantity to be chemically important must use a surrogate element. MPACT has an input option to name pairs of elements, so that the first in the pair is the actual element, and the second is the surrogate that appears in the chemical database. In this work, zirconium is one such element that is chemically important but does not appear in the MSTDB. A commonly used surrogate for zirconium is plutonium, as it has similar oxidation states and forms similar compounds. Thus, in the results presented later in this work, anywhere zirconium is present in the Thermochimica calculations, the data used from MSTDB are actually for plutonium. Using these surrogates can be very important, as zirconium plays a crucial role in ensuring that the salt is actually liquid at the operating conditions of the MSRE. When zirconium is neglected, large quantities of solid compounds are formed, which could not be the case physically.

CHAPTER 13. NODAL CROSS SECTION GENERATION

MPACT can generate nodal cross sections and other values of interest to drive nodal simulator calculations. This chapter presents the details of how these calculations are performed, followed by a discussion of some additional features related to the nodal data generation.

13.1. NODAL DATA CALCULATION METHODOLOGY

The next several sections will describe how nodal data are calculated by MPACT. A number of different subscripts will be needed to distinguish values. These are shown in Table 13.1.

Table 13.1. Nomenclature for subscripts used to describe nodal data generation theory

Subscript	Description
N	node index
g	fine group index
G	coarse group index
d	delayed neutron precursor group index
h	decay heat precursor group index
s	cell surface index
S	node surface index
C	node corner index
j	fine mesh region index
i	isotope index
I	full set of isotopes in a region

13.1.1 Node-Averaged Data

The following node-averaged data are calculated for each node:

- Volume V_N
- Scalar flux $\phi_{N,G}$
- Adjoint flux $\phi^*_{N,G}$
- Cross sections:
  - Transport $\Sigma_{tr,N,G}$
  - Removal $\Sigma_{r,N,G}$
  - Fission $\Sigma_{f,N,G}$
  - Nu-fission $\nu\Sigma_{f,N,G}$

  - Kappa-fission $\kappa\Sigma_{f,N,G}$
  - Fission distribution $\chi_{N,G}$
  - Scattering $\Sigma_{s,N,G\rightarrow G'}$
- Surface-averaged quantities:
  - Surface currents $J_{N,S,G}$
  - Surface flux $\phi_{N,S,G}$
  - Surface discontinuity factors $D_{N,S,G}$
- Thermal hydraulic quantities:
  - Fuel volume fraction $F_{N,FV}$
  - Wetted volume fraction $F_{N,WV}$
  - Moderator density $\rho_{N,M}$
  - Moderator temperature $T_{N,M}$
  - Moderator fluid fraction $F_{N,M}$
  - Coolant density $\rho_{N,C}$
  - Coolant temperature $T_{N,C}$
  - Coolant fluid fraction $F_{N,C} = F_{N,WV} - F_{N,M}$
  - Fuel temperature $T_{N,F}$
  - Fuel surface temperature $T_{N,FS}$
- Isotopic quantities:
  - ^{135}I number density $N_{N,I}$
  - ^{135}Xe number density $N_{N,Xe}$
  - ^{149}Pm number density $N_{N,Pm}$
  - ^{149}Sm number density $N_{N,Sm}$
  - ^{10}B number density $N_{N,B}$
  - ^{135}Xe absorption cross section $\Sigma_{a,N,G,Xe}$
  - ^{149}Sm absorption cross section $\Sigma_{a,N,G,Sm}$
  - ^{10}B absorption cross section $\Sigma_{a,N,G,B}$
- Time-dependent data:
  - Delayed neutron group fission distribution $\chi_{D,d,N,G}$
  - Delayed neutron group decay constants $\lambda_{D,d,N}$
  - Delayed neutron group yields $\beta_{D,d,N}$
  - Decay heat group decay constants $\lambda_{H,h,N}$
  - Decay heat group yields $\beta_{H,h,N}$

  - Neutron group velocities $v_{N,G}$
  - Burnup B_N
  - Heavy metal mass M_N
- Miscellaneous data:
  - Control rod insertion fraction for each node CR_N
  - Axial mesh for the core ΔZ_z
  - Corner point fluxes $\phi_{N,C,G}$
  - Transfer leakage moments $L_{N,A,G}$

13.1.1.1 Volume

The node volumes are calculated by simply summing up the fine mesh region volumes in the node:

$$V_N = \sum_{j \in n} V_j \, . \tag{13.1}$$

13.1.1.2 Scalar and adjoint flux

The node-averaged scalar flux is calculated by volume-averaging the fine mesh fluxes and performing an energy collapse for each coarse group:

$$\phi_{N,G} = \frac{\sum_{j \in N} V_j \sum_{g \in G} \phi_{j,g}}{V_N} \, , \tag{13.2}$$

This same procedure is also followed for the adjoint flux:

$$\phi^*_{N,G} = \frac{\sum_{j \in N} V_j \sum_{g \in G} \phi^*_{j,g}}{V_N} \, . \tag{13.3}$$

13.1.1.3 Cross sections

All cross sections are calculated using a flux-volume weighting of the fine mesh cross sections while also collapsing the fine groups for each coarse group:

$$\Sigma_{x,N,G} = \frac{\sum_{j \in n} V_j \sum_{g \in G} \Sigma_{x,j,g} \phi_{j,g}}{\phi_{N,G} V_N} \, , \tag{13.4}$$

where Σ_x can be Σ_{tr}, Σ_r, Σ_f, $\nu\Sigma_f$, $\kappa\Sigma_f$, or $\Sigma_{s,g \to g'}$. To homogenize the fission distribution, a fission source weighting must be used:

$$\chi_{N,G} = \frac{\sum_{j \in n} V_j \sum_{g \in G} \chi_{j,g} \nu\Sigma_{f,j,g} \phi_{j,g}}{\nu\Sigma_{f,N,G} \phi_{N,G} V_N} \, . \tag{13.5}$$

13.1.1.4 Thermal hydraulic quantities

Several volume fractions are calculated by examining each of the fine mesh regions:

$$F_{N,FV} = \frac{\sum_{j \in N, j \in fuel} V_j}{V_N} , \tag{13.6}$$

$$F_{N,WV} = \frac{\sum_{j \in N, j \in fluid} V_j}{V_N} , \tag{13.7}$$

$$F_{N,M} = \frac{\sum_{j \in N, j \in M} V_j}{V_N} , \tag{13.8}$$

$$F_{N,C} = \frac{\sum_{j \in N, j \in C} V_j}{V_N} . \tag{13.9}$$

Average temperatures are calculated by volume-averaging over the appropriate regions:

$$T_{N,x} = \frac{\sum_{j \in N, j \in x} T_{j,x} V_j}{V_N} , \tag{13.10}$$

where x can refer to moderator M, coolant C, or fuel F. Likewise for the density:

$$\rho_{N,x} = \frac{\sum_{j \in N, j \in x} \rho_{j,x} V_j}{V_N} . \tag{13.11}$$

Finally, the node-averaged fuel surface temperature is calculated. This is done by looping over all fuel rods R_N in node N and using a perimeter weighting (which is equivalent to surface area weighting since all rod segments in N are the same height):

$$T_{N,FS} = \frac{\sum_{i \in R_N} T_{FS,i} P_i}{\sum_{i \in R_N} P_i} , \tag{13.12}$$

where P_i is the perimeter of rod i in node N. In most cases, this is equal to the outer circumference of the clad, but for pins that are split across node boundaries, it can be one half or one quarter of the outer clad's circumference.

13.1.1.5 Isotopic quantities

The isotopic number densities are calculated by volume weighting:

$$N_{N,x} = \frac{\sum_{j \in N} N_{j,x} V_j}{V_N} , \tag{13.13}$$

where x refers to ^{135}I, ^{135}Xe, ^{149}Pm, ^{149}Sm, or moderator-dissolved ^{10}B. The absorption cross sections can also be calculated:

$$\Sigma_{a,N,G,x} = \frac{\sum_{j \in N} V_j \sum_{g \in G} \Sigma_{a,j,g,x} \phi_{j,g}}{\phi_{N,G} V_N} , \tag{13.14}$$

where x refers to the three strong absorbers from the previous list: ^{135}Xe, ^{149}Sm, and ^{10}B.

13.1.1.6 Time-dependent data

The node-averaged burnup is calculated through volume weighting the burnup in the fuel regions:

$$B_N = \frac{\sum_{j\in V, j\in F} V_j B_j}{V_N F_{N,FV}} \,. \tag{13.15}$$

The delayed neutron group spectra $\chi_{D,d,N,G}$ are calculated using a fission source weighting of the fine mesh region delayed χ values. The only difference is that the calculation from Eq. (13.5) must be repeated for each of the 6 delayed neutron groups.

The yields for the delayed neutron precursors are calculated using adjoint fission source weighting for each delayed group d:

$$\beta_{D,d,N} = \frac{\sum_{j\in N} V_j \beta_{D,d,j} \sum_{G'\in G} \sum_{g\in G'} \chi_{D,d,j,g} \phi^*_{j,g} F_j}{\sum_{j\in N} V_j \sum_{G'\in G} \sum_{g\in G'} \chi_{j,g} \phi^*_{j,g} F_j} \,, \tag{13.16a}$$

$$F_j = \sum_{G'\in G} \sum_{g\in G'} \nu\Sigma_{f,j,g} \phi_{j,g} \,, \tag{13.16b}$$

where the fine mesh region betas are calculated as follows, to account for the differences in yields between fissionable isotopes:

$$\beta_{D,d,j} = \frac{\sum_{i\in I} \beta_{D,d,i} N_i \sum_G \sum_{g\in G} \nu\sigma_{f,g,i} \phi_{j,g}}{\nu\Sigma_{f,j,g} \phi_{j,g}} \,, \tag{13.17}$$

where $\beta_{D,d,i}$ is taken from the cross section library.

The delayed neutron precursor group decay constants must also be calculated for each node. Although the precursor isotopes in each group are fixed, the distribution of those isotopes produced by fission depends on which fissionable isotope produced them. Because of this, the precursor group decay constants are isotope-dependent, resulting in node-dependent homogenized decay constants. Thus, the precursor decay constants are calculated via isotope-dependent fission source weighting:

$$\lambda_{D,d,N} = \frac{\sum_{j\in N} V_j \sum_{i\in I} \lambda_{D,d,i} N_{j,i} \sum_G \sum_{g\in G} \nu\sigma_{f,j,g,i} \phi_{j,g}}{\nu\Sigma_{f,N,G} \phi_{N,G} V_N} \,, \tag{13.18}$$

where $\lambda_{D,d,j,i}$ is the precursor group d decay constant (from the cross section library) for fission by isotope i, $N_{j,i}$ is the region j number density of isotope i, and $\nu\sigma_{f,j,g,i}$ is the microscopic nu-fission cross section for isotope i.

The decay heat precursors are calculated using isotope-dependent heat production weighting. This essentially works out to be the same equation as Eq. 13.18, replacing the nu-fission cross section with kappa-fission and the decay constant with the yield:

$$\beta_{H,h,N} = \frac{\sum_{j\in N} V_j \sum_{i\in I} \beta_{H,h,i} N_{j,i} \sum_G \sum_{g\in G} \kappa\sigma_{f,j,g,i} \phi_{j,g}}{\kappa\Sigma_{f,N,G} \phi_{N,G} V_N} \,. \tag{13.19}$$

Likewise, for the decay heat precursor decay constants:

$$\lambda_{H,h,N} = \frac{\sum_{j\in N} V_j \sum_{i\in I} \lambda_{H,h,i} N_{j,i} \sum_G \sum_{g\in G} \kappa\sigma_{f,j,g,i} \phi_{j,g}}{\kappa\Sigma_{f,N,G} \phi_{N,G} V_N} \,. \tag{13.20}$$

The values for $\beta_{H,h,i}$ and $\lambda_{H,h,i}$ are taken from the ANS Standard ANSI/ANS-5.1-2014 [3].

Finally, the group-wise neutron velocities must be homogenized by performing a flux-weighted average of the inverse of the velocities to match how the velocities are used in the time-dependent transport equation:

$$v_{N,G} = \left(\frac{\sum_{j \in N} V_j \sum_{g \in G} \frac{1}{v_g} \phi_{j,g}}{\phi_{N,G} V_N} \right)^{-1} , \tag{13.21}$$

where v_g are the energy group neutron velocities taken from the cross section library.

13.1.2 Surface Flux Moments

Surface flux and surface current are calculated during the 2D/1D calculations on the pin-homogenized mesh. For the pin cell surfaces that are also part of a node surface, the surface flux and current can be averaged over the node surface using the pin cell surface areas as weights:

$$\phi_{S,G} = \frac{\sum_{s \in S} A_s \sum_{g \in G} \phi_{s,g} A_s}{\sum_{s \in S}} , \tag{13.22}$$

$$J_{S,G} = \frac{\sum_{s \in S} A_s \sum_{g \in G} J_{s,g}}{\sum_{s \in S} A_s} , \tag{13.23}$$

where A_s is the surface area of a pin-homogenized cell surface s.

13.1.3 Corner Point Fluxes

In MPACT, it is not strictly possible to calculate the corner fluxes. Instead, MPACT takes the fine mesh region fluxes of the nearest neighbors and volume-averages them to obtain a good approximation of the corner flux. In most cases, there are 4 regions touching the corner that are averaged. However, around the edges of the model, there could be only 1 or 2 regions.

For cases in which a corner lies on a rotational boundary, the 1 or 2 regions that touch the corner are averaged. The rotationally equivalent corner is then identified by looking across the boundary, and the results for the two equivalent corners are averaged as well. This ensures that the rotational symmetry of a problem is properly reflected in the corner fluxes.

13.1.4 Discontinuity Factors

To calculate the discontinuity factors, the same nodal kernels described in Chapter 6 are used. Specifically, the one-node NEM kernel can be used with quartic basis functions, and the one-node SENM kernel can be used with either quadratic or quartic polynomials in addition to the sinh and cosh functions that are always used for SENM.

For each node and each direction (x, y, and z), the user-requested nodal kernel is applied. The cross sections and node-averaged flux are taken from the calculations described earlier in this chapter, and the surface currents $J_{S,G}$ are taken from the fine mesh transport calculation. The diffusion coefficient uses the traditional definition:

$$D_{N,G} = \frac{1}{3\Sigma_{tr,N,G}} . \tag{13.24}$$

The nodal method is applied to all energy groups simultaneously to capture potential up-scatter effects, with the result of the method being the homogeneous surface fluxes $\phi^{hom}_{S,G}$.

The surface discontinuity factor is defined as the ratio of the homogeneous surface flux from the nodal kernel to the transport surface flux:

$$f_{df,S,G} = \frac{\phi^{het}_{S,G}}{\phi_{S,G}} . \tag{13.25}$$

13.2. ADDITIONAL FEATURES

13.2.1 Meshing Options

MPACT supports two primary radial mesh options for nodal data generation. The first is the quarter assembly node, which is written to a NODAL_XS group in the HDF5 file. For this data, all quantities of interest are averaged over every quarter assembly at each axial level of the MPACT model. The second option is the assembly node, which is written to an ASSEMBLY_XS group in the HDF5 file. In this case, the data are averaged over the entire assembly for each axial level. By default, both sets of data are written to the HDF5 file when the nodal data generation is enabled.

MPACT also has an option to collapse data axially in the axial reflector region. By default, the data are generated for every axial level in the MPACT model. However, this option allows the upper and lower axial reflectors to be collapsed into a single level for each. This allows support for nodal codes that only use one node for the reflector regions.

13.2.2 Energy Group Structure

Currently, all nodal data are generated for just two energy groups. The default behavior is for the boundary between the two groups to be set to the upper boundary of the first fine energy group that has up-scatter from lower energy groups. A user option allows this behavior to be changed so that the coarse group structure can be adjusted. The boundary between the two groups can be set to any of the fine group structure energy group boundaries. When the user specifies a boundary, it must align exactly with one of the fine group boundaries. Support for "splitting" fine groups across coarse groups is not implemented.

13.2.3 Adaptive Discontinuity Factors

The normal discontinuity factor calculation makes no guarantee that the discontinuity factors will be positive or otherwise reasonable. In many cases, especially for nodes along a vacuum boundary, the factors can be very large or negative, which can cause problems for nodal core simulators.

To address this, an adjustment is made to the discontinuity factors on vacuum boundaries. The node balance equation is given in Eqs. (6.8a) and (6.8b) and is repeated here for convenience:

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

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

To improve the quality of the discontinuity factor, a correction factor is added to both TL_g^{XY} and $\Sigma_{r,g}$. This reduces the leakage of neutrons across the vacuum boundary, but it increases the amount of absorption in the node to preserve neutron balance. After making this adjustment, the discontinuity factor is recalculated.

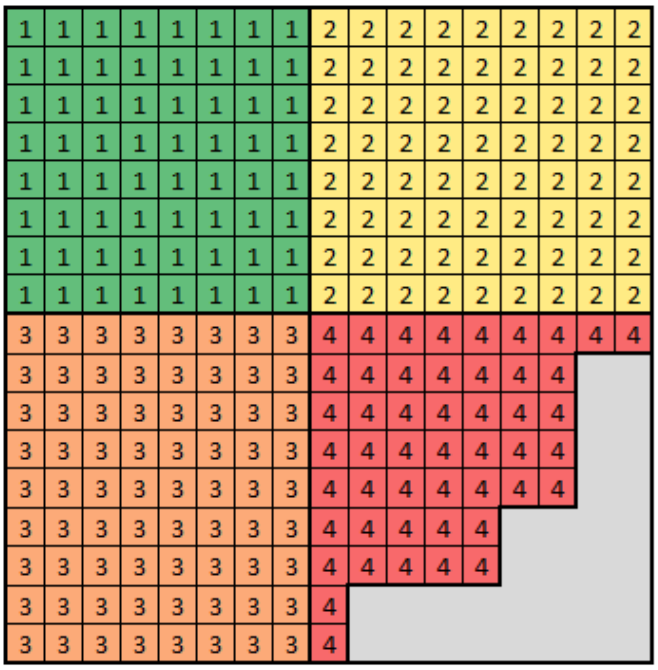
This procedure continues iterating using Newton's method, targeting a discontinuity factor of 1.0. This is done for each energy group until a combination of vacuum boundary current and removal cross section is found that results in discontinuity factors between 0.8 and 1.2 for each energy group. Because of the rapid convergence of Newton's method, convergence is typically reached within a handful of iterations. Between each iteration, the values for all energy groups are updated before the next iteration is performed. This accurately captures the coupling between the groups from up-scattering.

CHAPTER 14. MISCELLANEOUS TOPICS

This chapter addresses topics that do not fit into any of the previous chapters. These topics required modifications to the various solvers (MOC, CMFD, axial transport), so they are best discussed separately.

14.1. 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 14.1 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.

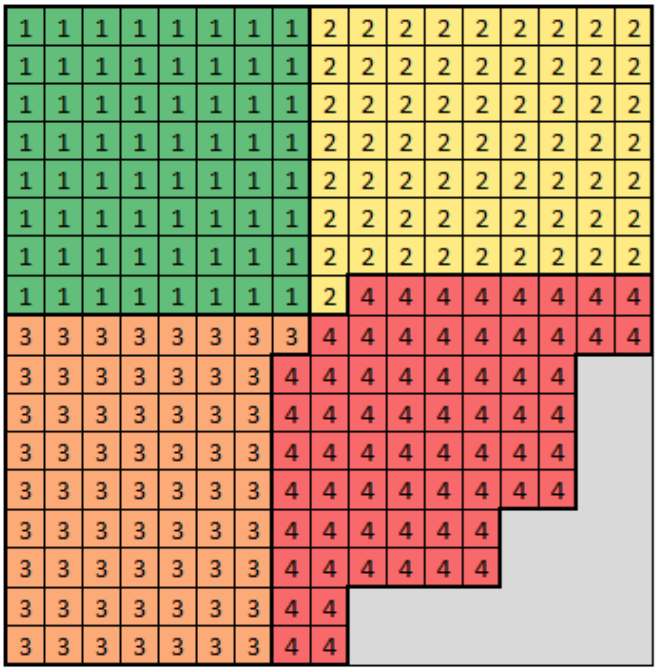


Partition	# Modules
1	64
2	72
3	72
4	49

Figure 14.1. 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 also receives one back. The initial concern with this option 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 14.2 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 14.2. Quarter-core layout with four partitions using new scheme.

14.2. AUTOMATED SPATIAL DECOMPOSITION

With the newly available module-based decomposition strategy, it became possible for a user to manually specify a decomposition. Manual specification allowed the user more freedom in the number of domains compared to previous methods. However, manual specification is a tedious and time-consuming process, and the resulting decomposition may not be well balanced. A graph-based decomposition scheme has been adopted to automate the process of generating these spatial decompositions in an efficient and effective manner.

Graph-based decomposition methods are based on graph partitioning methods. Three graph partitioning methods have been implemented and used in the spatial decomposition process in MPACT. The first method is the *recursive spectral bisection* (RSB) method [122], which uses connectivity of the graph to create a partitioning. The second method is the *recursive inertial nisection* (RIB) method [58], which uses only the

geometric information of the graph to create a partition. Finally, a new *recursive expansion bisection* (REB) method [56] was developed for specific application to rectilinear reactor cores, which takes both connectivity and geometric information of the graph into consideration during partitioning.

For decomposition, the reactor core is abstracted into a weighted graph based on the ray-tracing module mesh. Each module becomes a vertex, with the weight determined by the number of computational cells in the module. In 2D calculations, a radial decomposition is performed. In 3D calculations, weights are averaged over all axial planes, and these average weights are used to decompose a single plane. The resulting decomposition is then applied to all axial planes within the calculation.

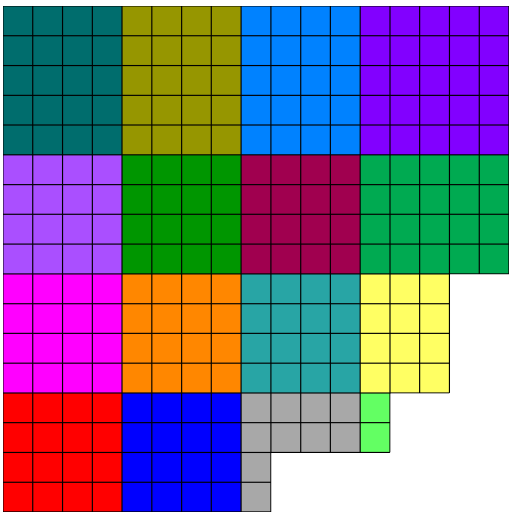


Figure 14.3. Z-tree automated decomposition.

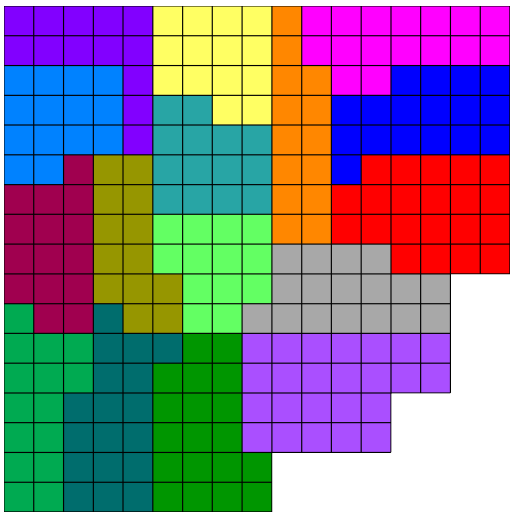


Figure 14.4. Graph-based decomposition using the REB method.

The previous automated decomposition method in MPACT was based on recursive spatial bisection based on Z-order curve (Z-tree). This previous decomposition method and these graph-based methods were used to decompose VERA problem 5a-2D [62] using 16 domains. Graph-based methods were able to give a reduction of 17–24% in overall runtime due to increased parallel efficiency. Each of the graph-based methods results in similarly balanced decompositions. Additionally, these graph-based decomposition methods allow for decomposition into an arbitrary number of domains. The original method only allows for certain numbers of domains to be used.

For highly decomposed cases in which the number of domains is close to the number of modules, the graph-based methods may result in slower converging calculations. This results from re-entrant rays. The Z-Order spatial bisection method avoids this issue by forcing domains to be rectangular.

14.3. ROTATIONAL SYMMETRY BOUNDARY TREATMENT

Several 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.

If the quarter core can be rotated through 90°, 180°, and 270°and the four quarters can then be stitched together, then rotational symmetry holds. Mirror symmetry, on the other hand, holds if the quarter core can be reflected across the x axis to create a half core and then the half core can be reflected across the y axis to create the whole core. Figure 14.5 shows the core layout and control rod bank assignments for

Watts Bar Unit 1 [62]. 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 rather are rotationally symmetric.

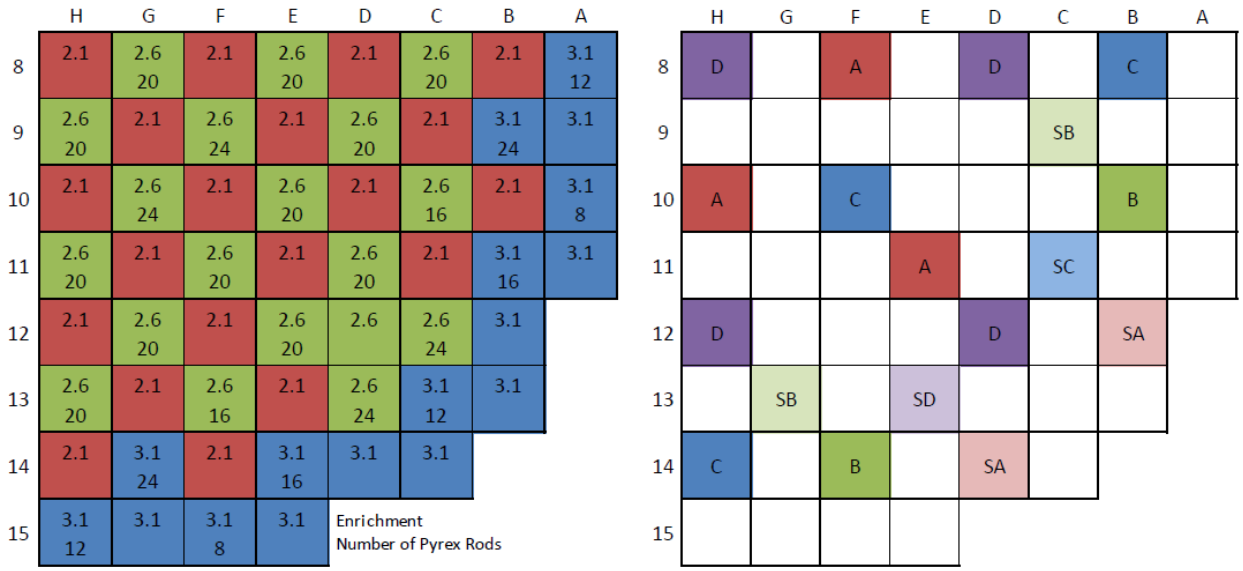


Figure 14.5. Watts Bar Unit 1 - core (left) and rod bank (right) layouts.

To allow these to be appropriately modeled, several modifications to both the CMFD and MOC solvers were necessary, 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 also had to be made 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 because it is most representative of operating reactors. When eighth-core symmetry exists, rotational and mirror symmetry are equivalent.

14.4. POLYNOMIAL ROD CUSPING TREATMENT

A common issue with 2D/1D calculations, as well as many other deterministic methods, is that of *rod cusping*. This occurs when a control rod or control blade is partially inserted into a MOC plane. When axial heterogeneities occur within a plane in MPACT, the materials are volume-homogenized axially, because it is a requirement of the 2D MOC calculations that there be no axial change in the cross sections for a given plane. However, the volume weighting does not accurately preserve the true reaction rates in the rodded and unrodded portions of the plane. This is illustrated in Figure 14.6.

To resolve this, a set of correction curves was generated to help resolve this issue. Control rods were simulated in MPACT at various depths in an MOC plane, with the difference between expected and simulated reactivity calculated in each case. Polynomial curves were then generated to allow MPACT to calculate

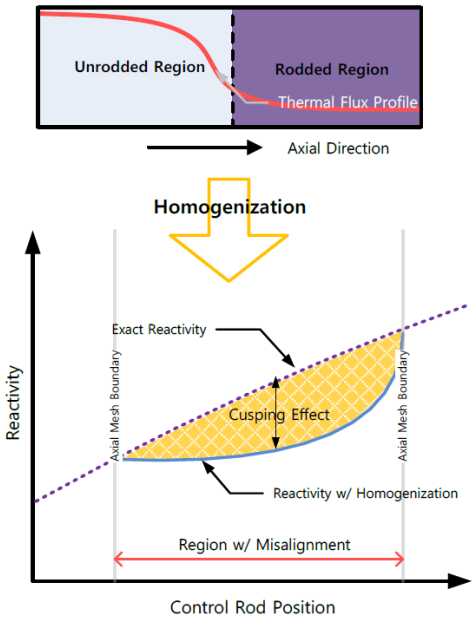


Figure 14.6. Illustration of the rod cusping error. [81]

updated weights to use when homogenizing the materials. This process always results in weighting the control rod region less than its volume fraction and weighting the unrodded region more than its volume fraction. Curves were generated for control rods made of silver-indium-cadmium (AIC), boron carbide (B_4C), and tungsten (W):

$$C_{AIC}(V_U) = 38.348V_U^6 - 95.832V_U^5 + 90.477V_U^4 - 39.326V_U^3 + 7.0843V_U^2 - 0.4708V_U\,, \tag{14.1a}$$

$$C_{B_4C}(V_U) = 46.843V_U^6 - 117.14V_U^5 + 109.78V_U^4 - 47.104V_U^3 + 9.1734V_U^2 - 0.5527V_U\,, \tag{14.1b}$$

$$C_W(V_U) = 4.7734V_U^6 - 14.751V_U^5 + 19.512V_U^4 - 12.762V_U^3 + 4.0769V_U^2 - 0.1505V_U\,, \tag{14.1c}$$

where V_U is the unrodded volume fraction, and C_x is the corresponding percent change in k_{eff} from full insertion to full withdrawal for a given rod type x. An illustration of this process is shown in Figure 14.7. This process and the results are discussed in more detail in the paper by Graham et al. [65].

14.5. SUBPLANE CMFD

Using conventional CMFD, each pin cell is homogenized using the quantities defined in Eq. (7.3) in every slice in the model so that every axial CMFD cell corresponds to a MOC plane. The radial coupling coefficients defined in Eq. (7.8) are obtained by calculating the current at the interface between each pair of pin cells using the transport sweeper, whereas 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 without significantly increasing the computational burden of the calculations, researchers at the Korea Atomic Energy Research Institute developed the *subplane scheme* [26, 126]. While preparing the CMFD and axial calculations, this scheme divides the MOC planes, referred

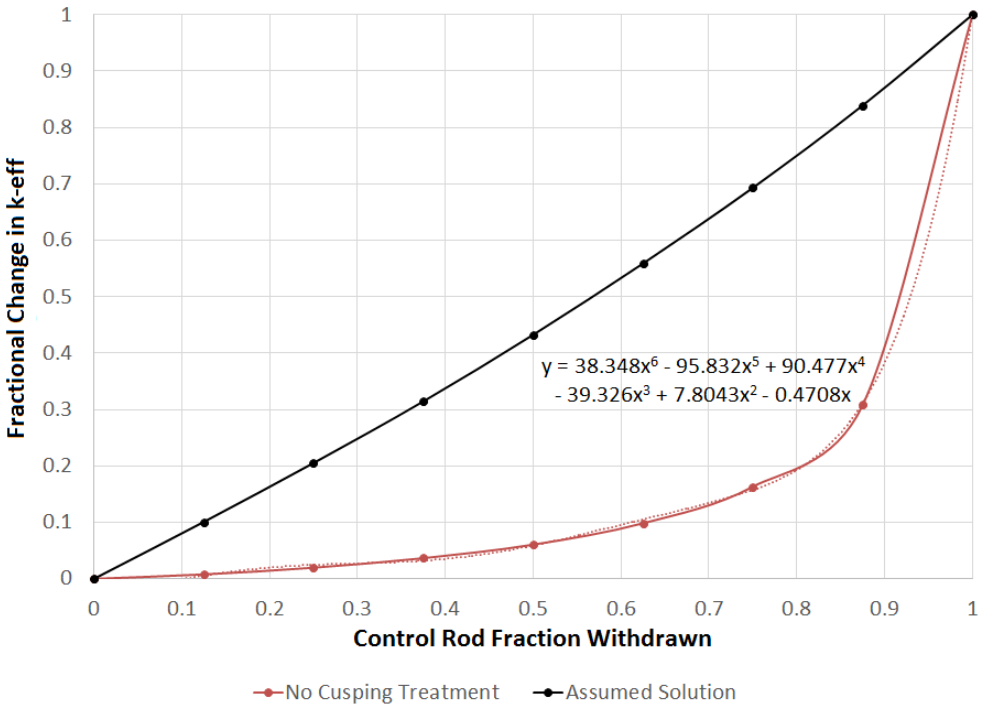


Figure 14.7. Polynomial fits to correct the control rod cusping issue [65].

to as *macroplanes*, into multiple subplanes, as illustrated in Figure 14.8. This axially refines the CMFD and 1D transport meshes to allow them to capture intra-plane detail without increasing the number of 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.

14.5.1 Homogenization

For 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 MOC 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:

$$c_{g,i}^{(l)} = \frac{\phi_{g,i}^{(l-1)}}{\phi_{g,i}^{(l-1)}} = \frac{\phi_{g,i}^{(l-1)} \sum_{i'=1}^{N_{sp}} V_{i'}}{\sum_{i'=1}^{N_{sp}} \phi_{g,i'}^{(l-1)} V_{i'}}, \tag{14.2}$$

where superscripts indicate from which iteration the values are taken, and N_{sp} is the number of subplanes on each macroplane 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 to update the radial shape of the fine mesh flux. 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 $\phi_{g,i}^{(l)}$ and fission source distribution $\chi_{g,i}^{(l)}$ are changed, providing an axial shape for the source term in the eigenvalue calculation.

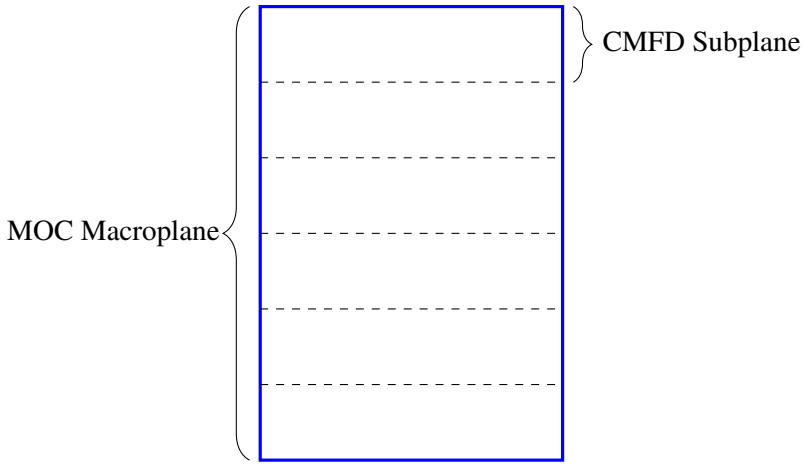


Figure 14.8. Illustration of multiple subplanes in one macroplane.

14.5.2 Coupling Coefficients

In addition to the homogenized cell terms, the coupling coefficients described by Eqs. (7.6) and (7.8) 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.

14.5.3 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 Eq. (7.9):

$$\phi^{(k)}_{\text{trans},g,j} = \left(\frac{\sum_{i'=1}^{N_{sp}} \phi^{(k)}_{\text{CMFD},g,i'} V'_i}{\sum_{i'=1}^{N_{sp}} \phi^{(k-1)}_{\text{CMFD},g,i'} V'_i} \right) \phi^{(k-1)}_{\text{trans},g,j} \,. \tag{14.3}$$

The surface fluxes can also be homogenized axially in the same way to scale the angular flux boundary conditions using Eq. (7.10).

14.6. SUBPLANE-BASED SUBGRID SOLVERS

While the subplane scheme allows for stable, faster calculations using 2D/1D, the thicker macroplanes used by the MOC calculations can lead to degradation of the solution accuracy. This is especially true when intraplane heterogeneities start to change significantly enough within a macroplane that the coarser macroplane mesh can no longer accurately capture the effects in the MOC calculation. Examples of such heterogeneities are partially inserted control rods, intraplane temperature and density distributions resulting from TH feedback, and intraplane isotopic distributions during depletion calculations. To address these

issues, *subgrid* methods have been implemented to improve the accuracy of the 2D/1D calculations and to make use of the subplane scheme. The methods presented in this section have been exercised on control rod cusping [65] and on spacer grids, TH feedback, depletion calculations, and axial reflectors [140].

14.6.1 Subgrid Heterogeneity Treatment

The first method to deal with intraplane heterogeneity is to simply extend the homogenization and projection steps described in the previous section. For the homogenization step without subplane, Eq. 7.3 is applied. However, this equation assumes that the cross section for each fine mesh region is axially constant in the macroplane. To apply the subgrid treatment, this assumption must be removed, allowing the cross section to vary axially, leading to a modified homogenization equation:

$$\Sigma_{x,j,g} = \frac{\sum_{i\in j}\sum_{k\in i}\Sigma_{x,k,g}\phi_{k,g}V_k}{\sum_{i\in j}\phi_{i,g}V_i} \,, \tag{14.4a}$$

$$\phi_{k,g} = \phi_{i,g}\frac{\phi_{z,g}^{l-1}}{\frac{\sum_{z'\in Z}\phi_{z',g}V_z}{V_i}} \,, \tag{14.4b}$$

where subscript z refers to the subplane index in region i of a macroplane, and Z is the collection of subplanes in a macroplane. Effectively, the previous subplane CMFD solution is used as an axial shaping function on the current fine mesh flux. Combining this axially shaped fine mesh flux with the axially heterogeneous cross sections provides unique cross sections for each subplane CMFD cell. This allows the CMFD calculation to more accurately resolve axial heterogeneities in a macroplane.

To complete the treatment, it is desired to improve the macroplane cross sections as well. Typically, when there is an axial heterogeneity in the macroplane, the cross sections are volume weighted, which leads to poor estimates of the reaction rates. To resolve this issue, the projection step of the CMFD acceleration should also be augmented to address the cross sections instead of only addressing the fine mesh flux. This leads to a new projection equation for cross sections:

$$\Sigma_{x,i,g} = \frac{\sum_{k\in i}\Sigma_{x,k,g}\phi_{k,g}V_k}{\sum_{k\in i}\phi_{k,g}V_k} \,, \tag{14.5}$$

where each variable has the same meaning as that of the homogenization step. This improves the accuracy of the cross sections for the MOC calculations.

14.6.2 Subplane Collision Probabilities Rod Decusping

The subgrid heterogeneity treatment presented above significantly improves accuracy in the vicinity of objects such as control rods and spacer grids. However, the radial calculations are still completed using only homogenized cross sections. Thus, improvements can be made to this treatment by introducing corrections to the radial portion of the calculation. This is done by using an auxiliary collision probabilities (CP) calculation to generate radial flux profiles in each subplane.

14.6.2.1 Collision Probabilities Method

The CP method can be used to calculate flux spectra inside a pin cell [93]. The transport equation is written with the entire source on the RHS:

$$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi_g + \Sigma_{tr,g}(\boldsymbol{x})\,\varphi_g(\boldsymbol{x},\boldsymbol{\Omega}) = q_g(\boldsymbol{x},\boldsymbol{\Omega}) \,, \tag{14.6a}$$

$$q_g(\boldsymbol{x},\boldsymbol{\Omega}) = \frac{1}{4\pi}\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\to g}(\boldsymbol{x},\boldsymbol{\Omega}'\to\boldsymbol{\Omega})\,\varphi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')\,d\Omega' + \frac{1}{k_{eff}}\frac{\chi_g}{4\pi}\sum_{g'=1}^{G}\int_{4\pi}\nu\Sigma_{f,g'}(\boldsymbol{x})\,\varphi_{g'}(\boldsymbol{x},\boldsymbol{\Omega}')\,d\Omega' \ . \tag{14.6b}$$

The source can be assumed to be exclusively composed of scatter and fission, and the scattering source can be assumed to be isotropic, giving a simplified form of the source term:

$$q_g(\boldsymbol{x},\boldsymbol{\Omega}) \approxeq q_g(\boldsymbol{x}) = \frac{1}{4\pi}\Sigma_{g'=1}^{G}\Sigma_{s,g'\to g}(\boldsymbol{x})\,\phi_{g'}(\boldsymbol{x}) + \frac{1}{k_{eff}}\frac{\chi_g}{4\pi}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{x})\,\phi_{g'}(\boldsymbol{x}) \ . \tag{14.7}$$

The problem is now discretized into a 1D radial problem, assuming that the source $q_g(\boldsymbol{x})$ is flat inside each region r. The total source of neutrons in each region r can now be calculated simply by multiplying the source by the region volume V_r. If the probability $T_{g,r'\to r}$ of a neutron born uniformly and isotropically in region r' reaching region r is known for all regions r', then the scalar flux can be calculated as follows:

$$\phi_{g,r} = \sum_{r'=1}^{R} T_{g,r'\to r} q_{g,r'} V_{r'} \ . \tag{14.8}$$

All that then remains is to determine the components of the transport matrix $\underline{\underline{T_g}}$. These components are functions of both geometry and cross sections, but for the 1D cylindrical problem of interest here, they take the following form:

$$T_{ii} = \Sigma_{tr,i}V_i - P_{ii}C_i = \Sigma_{tr,i}V_i - \Sigma_{s,i}V_i - 2c_j\left(S_{ii} + S_{i-1\,j-1} - S_{ij-1} - S_{i-1\,j}\right) \ , \tag{14.9a}$$

$$T_{ij} = -P_{ji}c_j = -2c_j\left(S_{ii} + S_{i-1\,j-1} - S_{ij-1} - S_{i-1\,j}\right) \ , \tag{14.9b}$$

$$S_{ij} = \int_0^R \left(Ki_3\left(\tau_{ij}^{+}(y)\right) - Ki_3\left(\tau_{ij}^{-}(y)\right)\right)dy \ , \tag{14.9c}$$

$$c_i = \frac{\Sigma_{s,i}}{\Sigma_{tr,i}} \ , \tag{14.9d}$$

where Ki_3 is the third-order Bickley-Naylor function which is numerically integrated to calculate the transport matrix elements for the collision probabilities method.

As derived here, the CP method has several important approximations. First, a white boundary is used. This means that all exiting neutrons re-enter the cell, but not at the same angle. In other words, all escaping neutrons are isotropically returned into the cell. Second, it is assumed that the scalar flux varies in the radial direction only. No axial or azimuthal dependence is determined. Finally, no axial leakage sources from the greater 2D/1D calculation are handled. The 1D CP calculation is performed in isolation from the rest of the MPACT calculation. Since the CP method is only used to generate shape functions for the subplane flux, these assumptions are reasonable and do not hinder the efficacy of the CP method for this particular application.

14.6.2.2 Subgrid CP

When a macroplane contains an axial heterogeneity, the CP method can be used to improve the cross sections going into both the CMFD and the MOC calculations. The first step is to identify the pin cells in each macroplane that have axial heterogeneities such as control rods or spacer grids. Once these pin cells are identified, a 1D CP calculation is performed for each subplane in each pin cell, the results being a radial flux profile for each subplane of the pin cell.

Applying these radial profiles is straightforward. The main drawback in Eq. 14.4 was that the subplane fluxes used in the cross section homogenization have no radial shape. Thus, simply replacing the subplane fluxes used to calculate $\phi_{k,g}$ with the subplane CP fluxes for each radial region i gives improved cross sections for the CMFD calculation. This in turn improves the accuracy of the subplane CMFD flux solution, which provides a more accurate axial shape to use during the projection process. This procedure is illustrated in Figure 14.9 for a partially inserted control rod. Figure 14.10 illustrated the meshing when applying subgrid methods to spacer grids.

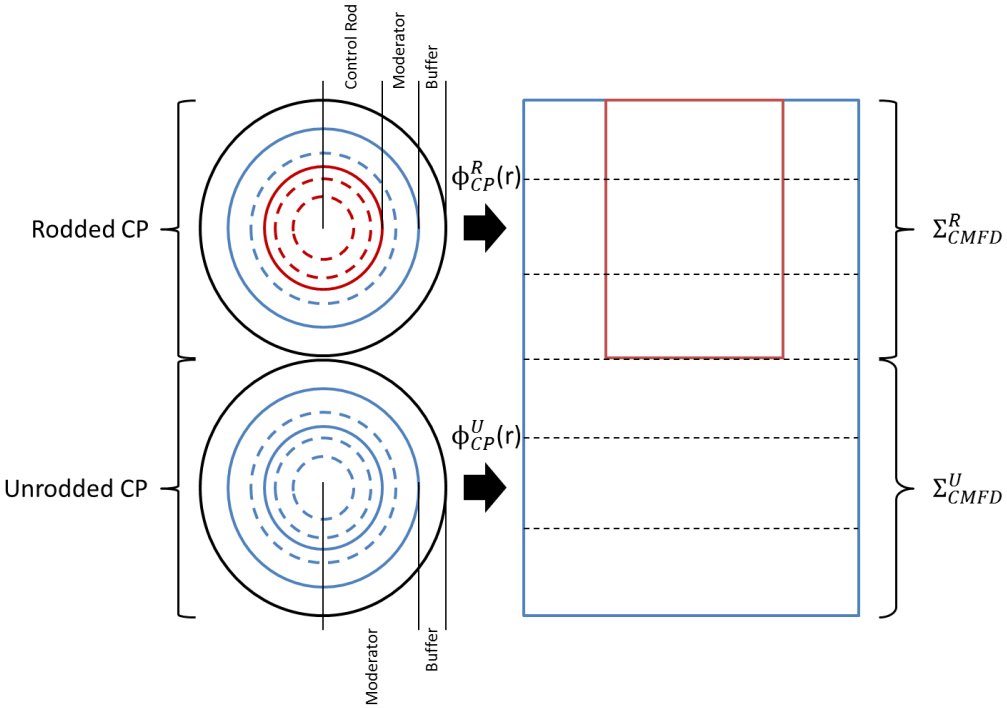


Figure 14.9. Illustration of the subgrid CP treatment [65].

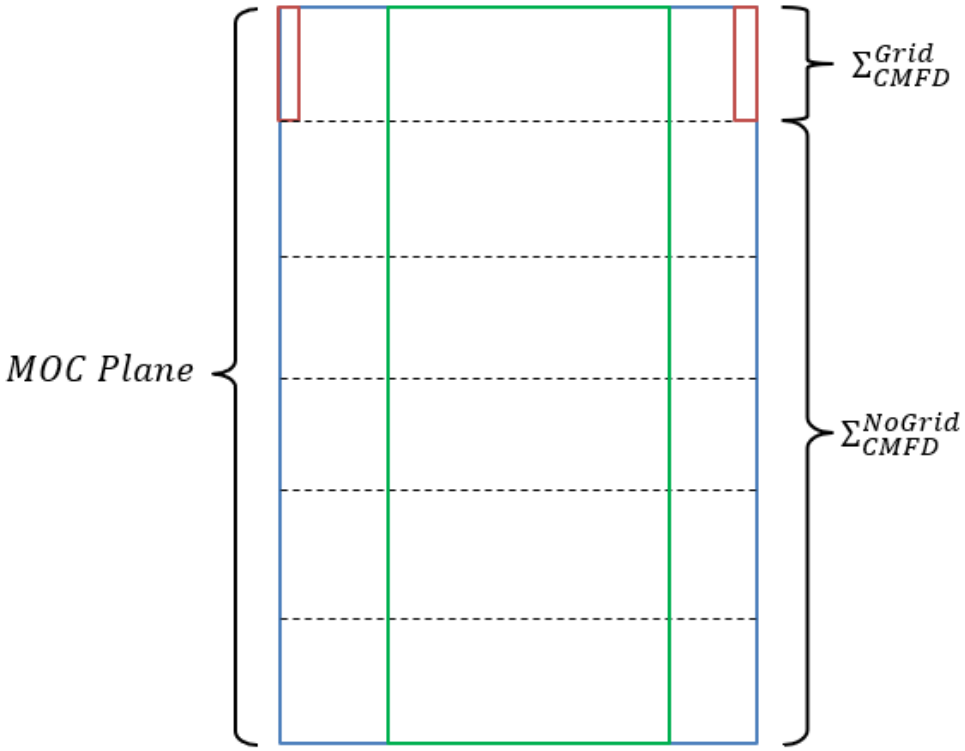


Figure 14.10. Illustration of the subplane meshing applied to spacer grids.

14.6.3 Other Subgrid Applications

The subplane subgrid treatments can also be applied to several other situations in MPACT to accelerate the solution time. First, the subgrid treatments can be applied to TH feedback. This allows the TH code to solve on a finer axial mesh without the requirement that MPACT use the same number of MOC planes. This can be important for some calculations, especially for accident scenarios for which the TH solution is important. A second application is for depletion calculations in which the depletion mesh is allowed to be more refined than the MOC transport mesh. For subgrid calculations applied to TH feedback and depletion, the TH parameters and isotopics are stored for each subplane. For the subplane CMFD calculation, these quantities are used directly; for the MOC calculations, they are homogenized within a macroplane to obtain the MOC cross sections. Application of the subgrid method to TH feedback (with a spacer grid explicitly meshed) is shown in Figure 14.11.

Finally, subgrid methods have also been applied to the axial reflector regions. Because no fission occurs in these regions, it is often not necessary to use multiple MOC planes to resolve the flux shape. Subplane-based subgrid solvers allow a single MOC plane to be used for each axial reflector, with subplane CMFD and the subgrid method resolving the intraplane shape. This is illustrated in Figure 14.12.

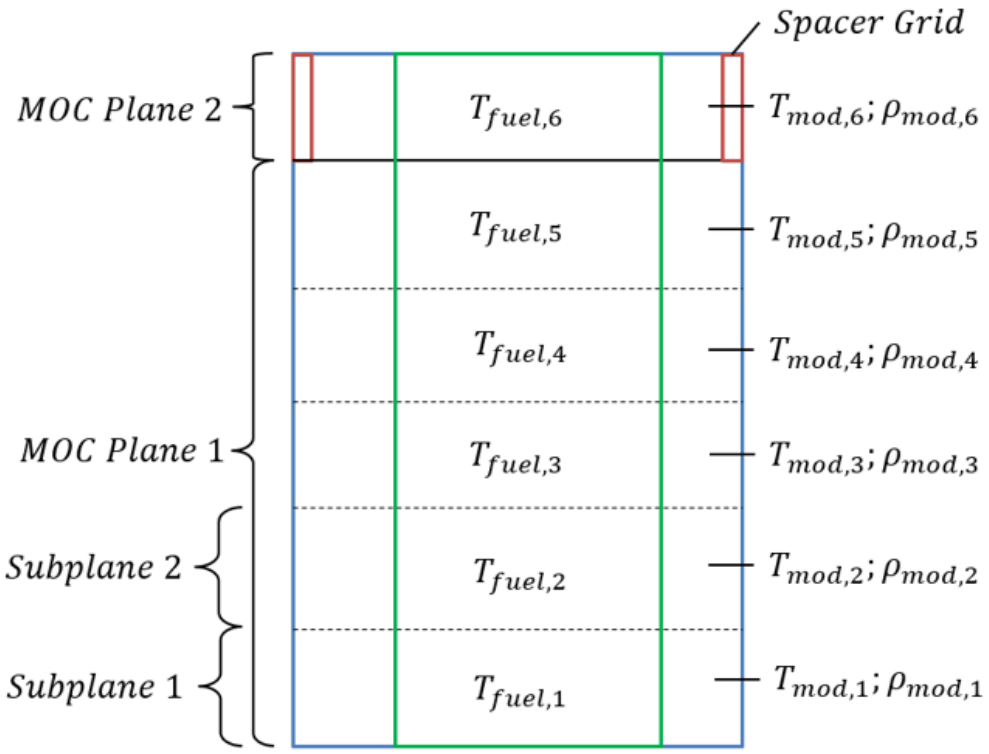


Figure 14.11. Illustration of the subplane meshing applied to TH feedback.

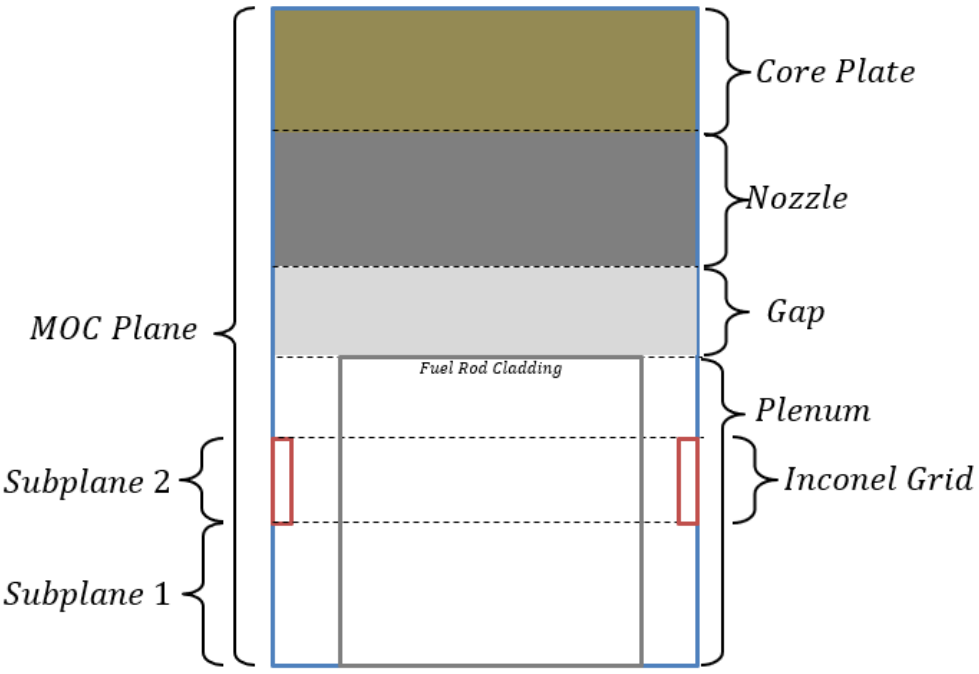


Figure 14.12. Illustration of the subplane meshing applied to axial reflectors.

14.7. 2D/1D 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. (4.12):

$$\Omega_x \frac{\partial \psi_{n,k}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{n,k}}{\partial y}(x,y) + \Sigma_t \psi_{n,k}(x,y) = \frac{\Sigma_s}{4\pi} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m + \frac{\nu\Sigma_f}{4\pi k_{\text{eff}}} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m - \frac{1}{4\pi\Delta_k} [J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y)], \tag{14.10}$$

using a finite difference approximation and Fick's Law to calculate the currents:

$$J_{z,k+1/2}(x,y) = -D_{k+1/2}(x,y) \frac{\phi_{k+1}(x,y) - \phi_k(x,y)}{\Delta_{k+1/2}}, \tag{14.11a}$$

where

$$D_{k+1/2}(x,y) = \frac{\Delta_{k+1} + \Delta_k}{3\Sigma_{t,k+1}(x,y)\Delta_{k+1} + 3\Sigma_{t,k}(x,y)\Delta_k}, \tag{14.11b}$$

$$\Delta_{k+1/2} = \frac{1}{2}(\Delta_{k+1} + \Delta_k). \tag{14.11c}$$

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 [83], [82]. 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 [84].

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 require under-relaxations in certain circumstances. The 2D/1D method based on Eqs. (14.10) and (14.11) is *not* commonly used in MPACT.

The Fourier analysis is based on the following fixed-source version of Eqs. (14.10) and (14.11):

$$\left(\Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Sigma_{t,k}\right) \psi_k(x,y,\mathbf{\Omega}) = \frac{1}{4\pi} \left\{ \Sigma_{s,k}\phi_k(x,y) + Q_k(x,y) + \frac{1}{\Delta_k} \left[\frac{D_{k+1/2}}{\Delta_{k+1/2}} \left(\phi_{k+1}(x,y) - \phi_k(x,y)\right) - \frac{D_{k-1/2}}{\Delta_{k-1/2}} \left(\phi_k(x,y) - \phi_{k-1}(x,y)\right) \right] \right\}, \tag{14.12a}$$

$$\phi_k(x,y) = \int \psi_k(x,y,\mathbf{\Omega}') d\Omega'. \tag{14.12b}$$

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:

$$\left(\Omega_x\frac{\partial}{\partial x}+\Omega_y\frac{\partial}{\partial y}+\Sigma_{t,k}\right)\psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega})=\frac{1}{4\pi}\Big\{\Sigma_{s,k}\phi_k^{(\ell)}(x,y)+Q_k(x,y)$$
$$+\frac{1}{\Delta_k}\left[\frac{D_{k+1/2}}{\Delta_{k+1/2}}\Big(\phi_{k+1}^{(\ell)}(x,y)-\phi_k^{(\ell)}(x,y)\Big)-\frac{D_{k-1/2}}{\Delta_{k-1/2}}\Big(\phi_k^{(\ell)}(x,y)-\phi_{k-1}^{(\ell)}(x,y)\Big)\right]\Big\}, \tag{14.13a}$$

$$\phi_k^{(\ell+1/2)}(x,y)=\int\psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega}')d\Omega' , \tag{14.13b}$$

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

$$\phi_k^{(\ell+1)}(x,y)=\theta\phi_k^{(\ell+1/2)}(x,y)+(1-\theta)\phi_k^{(\ell)}(x,y) . \tag{14.13c}$$

In the Fourier analysis of this method, Eqs. (14.13) 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. (14.13c) is to be determined; if $\theta = 1$, then the method defined by Eqs. (14.13) 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 [83, 84] for details, for $\theta = 1$ (the standard source iteration method), the growth factor ω is bounded from above and below by

$$\omega_{min} \le \omega \le \omega_{max} , \tag{14.14}$$

where

$$\omega_{max} = c = \frac{\Sigma_s}{\Sigma_t} = \text{ scattering ratio} , \tag{14.15a}$$

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

$$\omega_{min}=\begin{cases} 0 & , \quad \Sigma_t\Delta_z \ge \frac{2}{\sqrt{3}c}, (\text{ "large" } \Delta_z) , \\ c-\frac{4}{3(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z < \frac{2}{\sqrt{3}c}, (\text{ "small" } \Delta_z) , \end{cases} \tag{14.15b}$$

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 it is slowly converging when $\rho < 1$ but $\rho \approx 1$, and it is stable and rapidly converging when $\rho \ll 1$.

Using Eq. (14.13c), the Fourier analysis for $0 \le \theta \le 1$ yields

$$\theta\omega_{min} + 1 - \theta \le \omega \le \theta\omega_{max} + 1 - \theta . \tag{14.16}$$

For $\theta = 1$ (the source iteration method originally in DeCART), Eqs. (14.15) and (14.16) give

$$\rho = |\omega|_{max} = \begin{cases} c & , \quad \sqrt{\frac{2}{3c}} < \Sigma_t\Delta_z , \\ \frac{4}{3(\Sigma_t\Delta_z)^2} - c & , \quad \Sigma_t\Delta_z \le \sqrt{\frac{2}{3c}} . \end{cases} \tag{14.17}$$

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. (14.16) is the value for which the left and right sides are equal in magnitude but opposite in sign:

$$\theta_{opt}\omega_{min} + 1 - \theta_{opt} = -[\theta_{opt}\omega_{max} + 1 - \theta_{opt}]. \tag{14.18a}$$

Thus,

$$\theta_{opt} = \frac{2}{2 - (\omega_{max} + \omega_{min})} , \tag{14.18b}$$

and then

$$\begin{aligned} \rho = |\omega|_{max} &= \theta_{opt}\omega_{max} + 1 - \theta_{opt} \\ &= \frac{\omega_{max} - \omega_{min}}{2 - (\omega_{max} + \omega_{min})} . \end{aligned} \tag{14.18c}$$

Combining Eqs. (14.15) and (14.18) results in

$$\theta_{opt} = \begin{cases} \frac{2}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t \Delta_z , \\ \frac{3(\Sigma_t \Delta_z)^2}{2+3(1-c)(\Sigma_t \Delta_z)^2} & , \quad \Sigma_t \Delta_z \le \frac{2}{\sqrt{3c}} , \end{cases} \tag{14.19a}$$

$$\rho = \begin{cases} \frac{c}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t \Delta_z , \\ \frac{2}{2+3(1-c)(\Sigma_t \Delta_z)^2} & , \quad \Sigma_t \Delta_z \le \frac{2}{\sqrt{3c}} . \end{cases} \tag{14.19b}$$

Equation (14.19b) shows that the iterative method defined by Eqs. (14.13) with θ defined by Eq. (14.19a) is stable for all scattering ratios $0 \le c \le 1$ 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) [100]. The DSA equivalent of this CMFD system yields an update term $\delta\phi_k^{(\ell+1/2)}$ for the scalar flux. (Kelley and Larsen [84] provide an explicit definition of this correction term.) A CMFD iteration then consists of

(i) an axial sweep, as shown in Eq. (14.13a),
(ii) calculation of $\phi_k^{\ell+1/2}$, Eq. (14.13b),
(iii) the CMFD calculation of $\delta\phi_k^{(\ell+1/2)}$ [84], and
(iv) the relaxed update equation

$$\phi_k^{(\ell+1)}(x,y) = \theta\left(\phi_k^{(\ell+1/2)}(x,y) + \delta\phi_k^{(\ell+1/2)}\right) + (1-\theta)\phi_k^{(\ell)}(x,y) . \tag{14.20}$$

These equations constitute 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 synthesis scheme with $\theta = 1$ (no relaxation) may be written in the following form:

$$\omega_1(\lambda_r, \zeta) = 1 - \frac{1 - \zeta I_0(\lambda_r)}{1 - \zeta\left(1 + \frac{\lambda_r^2}{3}\right)^{-1}}, \tag{14.21a}$$

$$\zeta = c - \frac{\Lambda_z^2}{3}, \tag{14.21b}$$

$$\Lambda_z = \frac{\sin(\lambda_z \Sigma_t \Delta_z/2)}{\Sigma_t \Delta_z/2}. \tag{14.21c}$$

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. (14.18b) 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. (14.21). This function is evaluated numerically, and these extrema can be seen in Figure 14.13. 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).

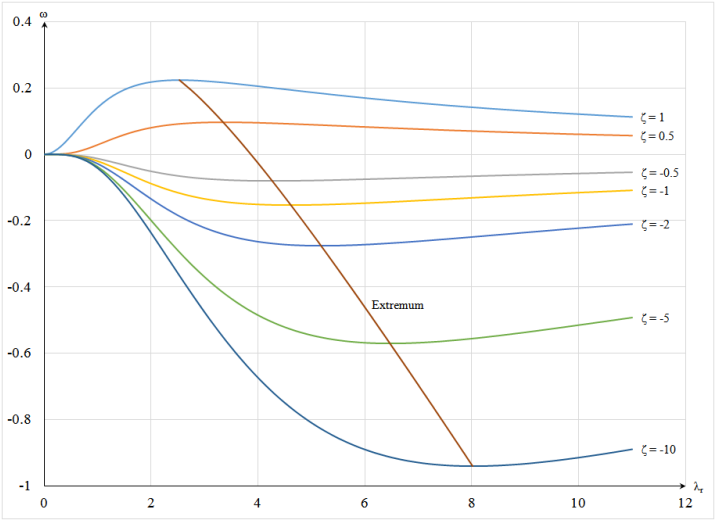


Figure 14.13. Growth factor ω for various radial frequencies λ_r for sample of parameterized values ζ.

Figure 14.13 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

$$\begin{aligned} \lambda_{ext} &\approx S + M(F - \zeta)^P, \\ S &= 1.93801895412889, \quad M = 1.88037759461481, \\ F &= 1.07821249297909, \quad P = 0.487975837139675. \end{aligned} \tag{14.22}$$

Now that λ_{ext} is known, ω_{min} can be calculated from Eqs. (14.21), and thus θ_{opt} can be calculated from Eq. (14.18b).

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

$$\Sigma_t \Delta_z \gtrapprox \frac{2}{\sqrt{3(c + 10.947)}}, \tag{14.23}$$

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

Figure 14.14 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 than 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. (14.12), 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. (14.13a) is used.

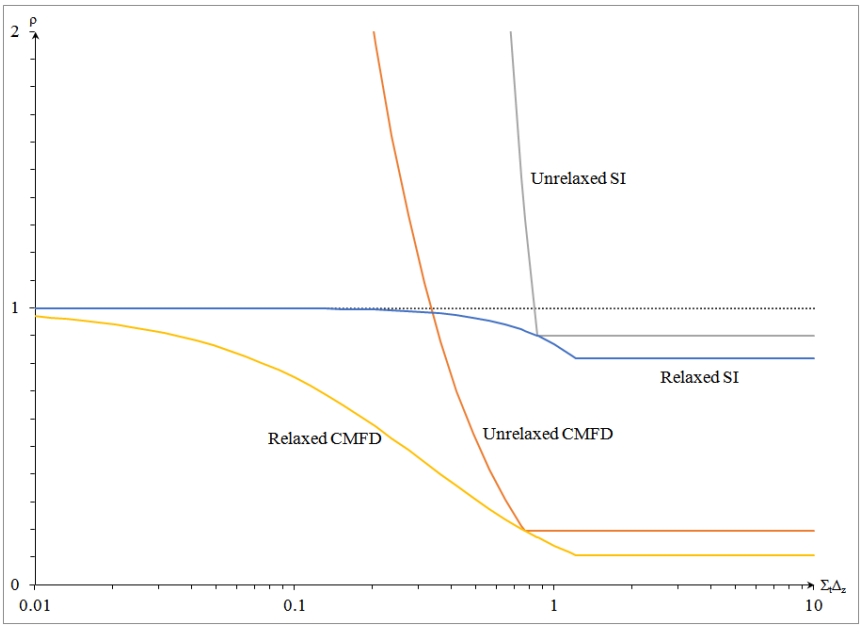


Figure 14.14. Spectral radius ρ for SI and CMFD vs. optical thickness $\tau_z = \Sigma_t\Delta_z$, with c =0.9.

14.8. 3D MOC CMFD ACCELERATION

CMFD becomes unstable when accelerating transport methods, but not when accelerating 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, then the iterations may converge very slowly due to large axial optical thickness. Although 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.

14.9. CRITICAL SEARCH

MPACT can perform a critical search capability by searching on a number of variables. The most common of these variables is soluble boron concentration, but it is also possible to search for critical control bank position, inlet flow temperature, core power, or core flow. The theory for critical boron search is described briefly; the procedure is generally the same for any of the other variables.

For a given iteration k, the reactivity worth estimate is calculated using previous boron concentrations b:

$$\Delta\rho^k = \begin{cases} 0.0\,, & k = 1 \\ -10^{-5}\,, & k = 2 \\ \frac{k_{eff}^{k-2} - k_{eff}^{k-1}}{b^{k-1} - b^{k-2}}\,, & k > 2 \end{cases}. \tag{14.24}$$

Using this worth, the new boron concentration can be calculated as

$$b^k = \min\left(u * \Delta\rho^k + (1-u)\,\Delta\rho^{k-1}\,,\ 0\right)\,, \tag{14.25}$$

where u is an underrelaxation factor to ensure stable convergence of the search algorithm.

BIBLIOGRAPHY

[1] BT Adams and JE Morel. A Two-Grid Acceleration Scheme for the Multigroup SN Equations with Neutron Upscattering. *Nuclear Science and Engineering*, 1993.

[2] A. C. Aldous. Numerical Studies of the Hydrogen Equivalent of Some Structural Materials in the Effect on U-235 Resonance Capture. Technical report, UKAEA Report AEEW-M 860, 1969.

[3] American Nuclear Society Standards Committee Working Group ANS-5.1. Decay Heat Power in Light Water Reactors. Technical report, American National Standards Institute, Inc., 2014.

[4] J. R. Askew. A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries. *UKAEA Report*, AEEW-M-1108, 1971.

[5] 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.

[6] 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.

[7] 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.

[8] R. S. Baker and K. R. Koch. An S_N Algorithm for the Massively Parallel CM-200 Computer. *Nucl. Sci. Eng.*, 128:312, 1998.

[9] 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.

[10] 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.

[11] M. J. Bell. ORIGEN–The ORNL Isotope Generation and Depletion Code. Technical report, ORNL-4628, 1973.

[12] T. M. Besmann, J. Ard, S. Utlak, J. W. McMurray, and R. A. Lefebvre. Status of the Salt Thermochemical Database. Technical Report ORNL/SPR-2019/1208, Oak Ridge National Laboratory, August 2019.

[13] I. I. Bondarenko et al. *Group Constants for Nuclear Reactor Calculations*. Consultants Bureau, New York, 1964.

[14] S. M. Bowman. SCALE 6: Comprehensive Nuclear Safety Analysis Code System. *Nucl. Technol.*, 174:126–148, 2011.

[15] W. Boyd, S. Shaner, L. Li, B. Forget, and K. Smith. The OpenMOC Method of Characteristics Neutral Particle Transport Code. *Annals of Nuclear Energy*, 68:43–52, 2014.

[16] William Boyd, Nathan Gibson, Benoit Forget, and Kord Smith. An analysis of condensation errors in multi-group cross section generation for fine-mesh neutron transport calculations. *Annals of Nuclear Energy*, 112:267–276, 2018.

[17] X. M. Chai, K. Wang, and D. Yao. The Linear Source Approximation in Three Dimension Characteristics Method. In *International Conference on Mathematics, Computational Methods \& Reactor Physics (M\&C 2009)*, volume 31, 6 2010.

[18] Y. I. Chang. Technical Rationale for Metal Fuel in Fast Reactors. *Nuclear Engineering and Technology*, 39(3):161–170, 2007.

[19] B. Chexal and G. Lellouche. Full-range Drift-flux Correlation for Vertical Flows. Technical Report EPRI-NP-3989-SR, ElectricPower Research Institute, 1985.

[20] D. Chisholm. Void Fraction during Two-Phase Flow. *Journal of Mechanical Engineering Science*, 15(3):235–236, 1973.

[21] J. Y. Cho et al. Cell Based CMFD Formulation for Acceleration of Whole-Core Method of Characteristics Calculations. In *Proceedings of KNS Fall Meeting, Suwon*, 2001.

[22] 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.

[23] 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.

[24] 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.

[25] N. Z. Cho. Fundamentals and Recent Developments of Reactor Physics Methods. *Nucl. Eng. Tech.*, 37:25–78, 2005.

[26] N. Z. Cho and S. G. Hong. Neutron Transport Theory; Computational Algorithms and Applications. In *Chungmoon-Gak, Seoul, Korea*, 2000.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] B. Collins, C. Gentry, A. Wysocki, and R. Salko. Molten Salt Reactor Simulations Using MPACT-CTF. *Transactions of the American Nuclear Society*, 6 2017.

[32] B. Collins, W. Gurecky, B. Okhuysen, D. Andersson, A. Elliot, J. Galloway, and B. Kendrick. MAMBA v.2.0 Theory Manual. Technical Report CASL-U-2019-1836-000, Oak Ridge National Laboratory, March 2019.

[33] B. Collins., S. Stimpson, B.W. Kelley, M.T.H. Young, B. Kochunas, A. Graham, E.W. Larsen, T. Downar, and A. Godfrey. Stability and Accuracy of 3D Neutron Transport Simulations Using the 2D/1D Method in MPACT. *Journal of Computational Physics*, 326:612–628, 2016.

[34] D. E. Cullen. Application of the Probability Table Method to Multigroup Calculations of Neutron Transport. *Nucl. Sci. Eng.*, 55:387, 1974.

[35] J. W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997.

[36] 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.

[37] T. J. Downar et al. PARCS: Purdue Advanced Reactor Core Simulator. In *Proc. PHYSOR 2002, Seoul, South Korea*, 2002.

[38] S. Dulla, E. Mund, and P. Ravetto. The Quasi-Static Method Revisited. *Prog. Nucl. Energy*, 50:908, 2008.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] B. Collins et al. Demonstration of Full Core Reactor Depletion with MPACT. Technical Report CASL-U-2014-0140-000, CASL, 2014.

[45] 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.

[46] 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.

[47] J. Cho et al. Transient Capability for a MOC-Based Whole Core Transport Code DeCART. *Trans. Am. Nucl. Soc.*, 92:721, 2005.

[48] 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.

[49] P. J. Turinsky et al. NESTLE, Few-Group Neutron Diffusion Equation Solver Utilizing the Modal Expansion Method for Eigenvalue, Adjoint, Fixed-Source Steady-State and Transient Problems. Technical report, North Carolina State University, 1994.

[50] R. J. J. Stammler et al. HELIOS Methods. Technical report, Studsvik Scandpower, 2003.

[51] 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.

[52] X. Fang et al. New Correlations of Single-phase Friction Factor for Turbulent Pipe Flow and Evaluation of Existing Single-phase Friction Factor Correlations. *Nuclear Engineering and Design*, 241:897–902, 2011.

[53] R. M. Ferrer and J. D. Rhodes. A Linear Source Approximation Scheme for the Method of Characteristics. *Nuclear Science and Engineering*, 182(2):151–165, 2016.

[54] R. M. Ferrer and J. D. Rhodes. The Linear Source Approximation and Particle Conservation in the Method of Characteristics for Isotropic and Anisotropic Sources. *Annals of Nuclear Energy*, 115:209–219, 2018.

[55] H. Finnemann, F. Bennewitz, and M. Wagner. Interface Nodal Current Technique for Multi-Dimensional Reactor Calculation. *Atomkernenergie*, 30:123, 1977.

[56] A. Fitzgerald, Z. Dodson, S. Stimpson, and B. Kochunas. Automated Decomposition of a Structured Grid. In *Transactions of the American Nuclear Society*, volume 117, pages 731–734, Washington DC, 2017.

[57] A. Fitzgerald and B. Kochunas. Fast Exponential Function Approximations for the Method of Characteristics with Linear Source. In *Transactions of the American Nuclear Society*, pages 645–648, Orlando, USA, 2018.

[58] N. Floros, J. S. Reeve, J. Clinckemaillie, S. Vlachoutsis, and G. Lonsdale. Comparative Efficiencies of Domain Decompositions. *Parallel Computing*, 21(11):1823–1835, 1995.

[59] D. Gaston et al. MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations. *Nuclear Engineering Design*, 239:1768–1778, 2009.

[60] 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.

[61] J. Gehin. *A Quasi-Static Polynomial Nodal Method for Reactor Analysis*. PhD thesis, Massachusetts Institute of Technology, 1992.

[62] A. Godfrey. VERA Core Physics Benchmark Progression Problem Specifications. Technical report, CASL-U-2012-0131-004, 2014.

[63] R. Goldstein and E. R. Cohen. Theory of Resonance Absorption of Neutrons. *Nucl. Sci. Eng.*, 13:132–140, 1962.

[64] A. Graham, B. Collins, R. Salko, and M. Asgari. Multiscale thermal hydraulic coupling methods for boiling water reactor simulation. In *Proceedings of PHYSOR 2022*, 2022.

[65] A. Graham, B. Collins, S. Stimpson, and T. Downar. Subplane Collision Probabilities Method Applied to Control Rod Cusping in 2D/1D. *Annals of Nuclear Energy*, 118:1–14, 2018.

[66] A. Graham, Z. Taylor, B. Collins, R. Salko, and M. Poschmann. Multiphysics Coupling Methods for Molten Salt Reactor Modeling and Simulation in VERA. *Nuclear Science and Engineering*, 2021.

[67] J. D. Hales et al. BISON Theory Manual: The Equations Behind Nuclear Fuel Analysis. Technical report, Idaho National Laboratory, 2015.

[68] M. J. Halsall. CACTUS, A Characteristics Solution to the Neutron Transport Equations in Complicated Geometries. *UKAEA Report*, AEEW-R-1291, 1980.

[69] M. J. Halsall. Neutron Transport in WIMS by the Characteristics Methods. In *Transactions of the American Nuclear Society1*, pages 454–455, 1993.

[70] A. Hebert. *Applied Reactor Physics*. Presses Internationales Polytechnique, 2009.

[71] A. Hébert. High-Order Linear Discontinuous and Diamond Differencing Schemes Along Cyclic Characteristics. *Nuclear Science and Engineering*, 184(4):591–603, 2016.

[72] A. Hébert. Acceleration of Step and Linear Discontinuous Schemes for the Method of Characteristics in DRAGON5. *Nuclear Engineering and Technology*, 49(6):1135–1142, 2017.

[73] Nicholas F. Herring, Benjamin S. Collins, and Thomas J. Downar. Native Natural Circulation in VERA. In *Proc. of PHYSOR 2022*, 2022.

[74] A. Hoffman. *A Time-Dependent Method of Characteristics Formulation with Time Derivative Propagation*. PhD thesis, University of Michigan, 2013.

[75] S. G. Hong and N. Z. Cho. CRX: A Code for Rectangular and Hexagonal Lattice Based on the Method of Characteristics. *Ann. Nucl. Energy*, 25, 1998.

[76] 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.

[77] M. Hursin. *Full Core, Heterogeneous, Time Dependent Neutron Transport Calculations with the 3D Code DeCART*. PhD thesis, University of California at Berkeley, 2010.

[78] M. James. Energy Released in Fission. *Journal of Nuclear Energy*, 34:517–536, 2019.

[79] 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.

[80] Y. S. Jung and H. G. Joo. Direct Whole Core Calculation with Thermal Feedback Using Planar MOC Generated Cross Sections Functions. In *Proc. M&C 2011, Rio de Janeiro, Brazil*, May 8-12, 2011.

[81] Y. S. Jung and H. G. Joo. Control Rod Decusping Treatment Based on Local 3-D CMFD Calculation for Direct Whole Core Transport Solvers. In *Proc. ICAPP 2014*, 2014.

[82] 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.

[83] 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.

[84] 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.

[85] K. S. Kim. Generation of the V4.2m1 AMPX and MPACT 51-Group Libraries with ENDF/B-VII.0 and VII.1. Technical report, CASL-U-2016-1177-000, Oak Ridge National Laboratory, 2016.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] Kang-Seog Kim, Aaron Graham, and Matthew Jessee. Dancoff-based Wigner-Seitz Approximation for the Subgroup Resonance Self-Shielding in the VERA Neutronics Simulator MPACT. In *Proc. of PHYSOR 2022*, 2022.

[92] G. Knoll. *Radiation Detection and Measurement*. John Wiley & Sons. Inc., New York City, 2000.

[93] D. Knott and A. Yamamoto. *Handbook of Nuclear Engineering (edited by D.G. Cacuci)*, volume 1, chapter 9: Lattice Physics Computations. Springer, 2010.

[94] B. Kochunas, B. Collins, T. Downar, and W. Martin. Overview of Development and Design of MPACT: Michigan Parallel Characteristics Transport Code. In *Proc. M&C 2013, Sun Valley, ID, USA*, May 5–9 2013.

[95] Brendan Kochunas, Benjamin Collins, Shane Stimpson, Robert Salko, Daniel Jabaay, Aaron Graham, Yuxuan Liu, Kang Seog Kim, William Wieselquist, Andrew Godfrey, Kevin Clarno, Scott Palmtag, Thomas Downar, and Jess Gehin. VERA Core Simulator Methodology for Pressurized Water Reactor Cycle Depletion. *Nucl. Sci. Eng.*, 185:217–231, 2017.

[96] Brendan Kochunas, Zhouyu Liu, and Thomas Downar. Parallel 3-D Method of Characteristics in MPACT. In *Proc. M&C 2013, Sun Valley, ID, USA*, May 5–9 2013.

[97] 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.

[98] S. Kosaka and T. Takeda. Verification of 3D Heterogeneous Core Transport Calculation Using Nonlinear Iteration Technique. *J. Nucl. Sci. Tech.*, 41:645–654, 2004.

[99] L. Lapidus and R. Luus. *Optimal Control of Engineering Processes*. Blaisdell Publishing Company, Waltham, Massachusetts, 1967.

[100] 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.

[101] E. W. Larsen and W. F. Miller. Convergence Rates of Spatial Difference Equations for the Discrete-Ordinates Neutron Transport Equations in Slab Geometry. *Nuclear Science and Engineering*, 73(1):76–83, 1980.

[102] R. Le Tellier and A. Hébert. On the Integration Scheme along a Trajectory for the Characteristics Method. *Annals of Nuclear Energy*, 33(14-15):1260–1269, 2006.

[103] 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.

[104] 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.

[105] F. Leszczynski. Neutron Resonance Treatment with Details in Space and Energy for Pin Cells and Rod Clusters. *Ann. Nucl. Energy*, 14:589–601, 1987.

[106] 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.

[107] Y. Liu et al. Development of explicit heat calculation and coupling between MPACT and CTF. Technical report, CASL-U-2019-1807-000, 2019.

[108] 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.

[109] 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.

[110] 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.

[111] Y. Liu, R. Salko, K. Kim, X. Wang, M. Kabelitz, S. Choi, B. Kochunas, B. Collins, and W. Martin. Improved MPACT Energy Deposition and Explicit Heat Generation Coupling with CTF. *Annals of Nuclear Energy*, 152:107999, 2021.

[112] 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.

[113] E. Masiello, R. Clemente, and S. Santandrea. High-Order Method of Characteristics for 2-D Unstructured Meshes. In *International Conference on Mathematics, Computational Methods & Reactor Physics*, 6 2009.

[114] R. G. McClarren. Theoretical Aspects of the Simplified P_N Equations. *Transport Theory and Statistical Physics*, 39:73–109, 2011.

[115] 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.

[116] K. Ott and R. Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, La Grange Park, IL, 1985.

[117] Hansol Park and Han Gyu Joo. Practical resolution of angle dependency of multigroup resonance cross sections using parametrized spectral superhomogenization factors. *Nuclear Engineering and Technology*, 49(6):1287–1300, 2017.

[118] P. Petkov, T. Takeda, and T. Mori. Comparison of the Flat and Linear Source Variants of the Method of Characteristics. *Annals of Nuclear Energy*, 26(10):935–942, jul 1999.

[119] P. T. Petkov and T. Takeda. Transport Calculations of MOX and UO_2 Pin Cells by the Method of Characteristics. *Journal of Nuclear Science and Technology*, 35(12):874–885, 1998.

[120] M.H.A. Piro, S. Simunovic, T.M. Besmann, B.J. Lewis, and W.T. Thompson. The Thermochemistry Library Thermochimica. *Computational Materials Science*, 67:266 – 272, 2013.

[121] M. Poschmann, B. Fitzpatrick, S. Simunovic, and M. Piro. Recent Development of Thermochimica for Simulations of Nuclear Materials. In *TMS 2020 149th Annual Meeting & Exhibition Supplemental Proceedings, 1003-12*, 2020.

[122] A. Pothen, H. D. Simon, and K.-P. P. Liu. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1989.

[123] A. K. Prinja and E. W. Larsen. Chapter 5: General Principles of Neutron Transport. In D. G. Cacuci, editor, *Handbook of Nuclear Engineering*, volume 1. Springer, 2010.

[124] C. Rabiti, G. Palmiotti, W. S. Yang, M. A. Smith, D. K. Kaushik, and A. Wollaber. Quasi Linear Representation of the Isotropic Scattering Source for the Method of Characteristics. In *International Conference on Mathematics, Computational Methods & Reactor Physics*, pages 3–7, January 2009.

[125] M. A. Rathbun and D. P. Griesheimer. Effect of Delayed Energy Release on Power Normalization in Reactor Depletion Calculations. *Trans. Am. Nucl. Soc.*, 116:1144, 2017.

[126] 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.

[127] R. K. Salko and M. N. Avramova. CTF Theory Manual. Technical report, The Pennsylvania State University, 2016.

[128] Robert Salko, Belgacem Hizoum, David Kropaczek, and Mehdi Asgari. Verification and validation of the alternative nonlinear two-phase subchannel (ants) code. *Nuclear Engineering and Design*, 397:111930, 2022.

[129] S. Santandrea and R. Sanchez. Positive Linear and Nonlinear Surface Characteristic Schemes for the Neutron Transport Equation in Unstructured Geometries. In *Physor*, 2002.

[130] S. Santandrea, R. Sanchez, and P. Mosca. A Linear Surface Characteristics Approximation for Neutron Transport in Unstructured Meshes. *Nuclear Science and Engineering*, 160(1):23–40, 2008.

[131] 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.

[132] 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.

[133] Q. Shen, S. Choi, and B. Kochunas. A Robust Relaxation-Free Multiphysics Iteration Scheme for CMFD-Accelerated Neutron Transport k-Eigenvalue Calculations — II: Numerical Results. *Nuclear Science and Engineering*, 195:1202–1235, 2021.

[134] Q. Shen and B. Kochunas. A Robust Relaxation-Free Multiphysics Iteration Scheme for CMFD-Accelerated Neutron Transport k-Eigenvalue Calculations — I: Theory. *Nuclear Science and Engineering*, 195:1176–1201, 2021.

[135] K. S. Smith. Nodal Method Storage Reduction by Nonlinear Iteration. *Trans. Am. Nucl. Soc.*, 44:265, 1983.

[136] K. S. Smith. Practical and Efficient Iterative Method for LWR Fuel Assembly Homogenization. *Trans. Am. Nucl. Soc.*, 71:238, 1994.

[137] R. J. J. Stammler and M. J. Abbate. *Methods of Steady-State Reactor Physics in Nuclear Design*. Academic Press, London, 1983.

[138] S. Stimpson and B. Collins. Implementation of a Red-Black SOR CMFD Solver in MPACT. In *Transactions of the American Nuclear Society*, volume 115, Las Vegas, Nevada, United States of America, 2016.

[139] 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].

[140] S. Stimpson, A. Graham, and B. Collins. Extended Applications of Subgrid Representation in the 2D/1D Method. *Nuclear Science and Engineering*, 195:778–793, 2020.

[141] 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.

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

[143] 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.

[144] 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.

[145] 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.

[146] 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.

[147] C. Tang and S. Zhang. Development and Verification of an MOC Code Employing Assembly Modular Ray Tracing and Efficient Acceleration Techniques. *Annals of Nuclear Energy*, 36(8):1013–1020, 2009.

[148] L. S. Tong and J. Weisman. *Thermal Analysis of Pressurized Water Reactors (2nd edition)*. American Nuclear Society, Le Grange Park, Illinois, 1979.

[149] 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.

[150] T. J. Trahan and E. W. Larsen. An Asymptotic Homogenized Neutron Diffusion Approximation. II. Numerical Comparisons. In *Proc. PHYSOR 2012, Knoxville, TN*, April 15–20 2012.

[151] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.

[152] E. L. Wachspress. *Iterative Solution of Elliptic Systems*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1966.

[153] X. Wang, Y. Liu, W. Martin, and Kim K. Energy Deposition Analysis for VERA Progression Problems by MCNP. In *Proceedings of PHYSOR 2018, Cancun, Mexico*, April 22-26 2018.

[154] M. L. Williams and M. Asgari. Computation of Continuous-Energy Neutron Spectra with Discrete Ordinates Transport Theory. *Nucl. Sci. Eng.*, 121:173–201, 1974.

[155] M. L. Williams and K. S. Kim. The Embedded Self-shielding Method. In *Proc. PHYSOR 2012, Knoxville, TN*, April 15–20, 2012.

[156] R. Williamson et al. Multidimensional Multiphysics Simulation of Nuclear Fuel Behavior. *Nuclear Materials*, 423:149–163, 2012.

[157] 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.

[158] 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.

[159] 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.

[160] B. C. Yee, E. W. Larsen, B. Kochunas, and Y. Xu. Space-Dependent Wielandt Shift Methods for Multigroup Diffusion Eigenvalue Problems. In *ANS Annual Meeting, New Orleans, LA, USA*, 2016.

[161] Ben C Yee, Brendan Kochunas, and Edward W Larsen. A Multilevel In Space and Energy Solver for Multigroup Diffusion Eigenvalue Problems. *Nuclear Engineering and Technology*, 2017.

[162] 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.

[163] 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.

[164] Zhaopeng Zhong, Thomas J. Downar, Yunlin Xu, Mark D. DeHart, and Kevin T. Clarno. Implementation of Two-level Coarse-mesh Finite Difference Acceleration in An Arbitrary Geometry, Two-dimensional Discrete Ordinates Transport Method. *Nuclear Science and Engineering*, 2008.

[165] A. Zhu et al. An Optimally Diffusive Coarse Mesh Finite Difference Method to Accelerate Neutron Transport Calculations. *Annals of Nuclear Energy*, 95:116–124, 2016.