DISCLAIMER

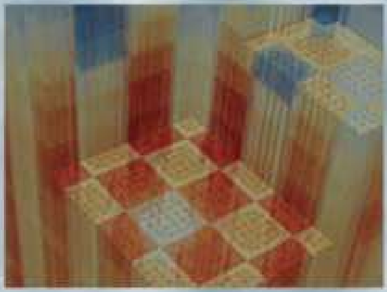
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness,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

CASL-U-2015-0078-000

MPACT Theory Manual

Version 2.0.0

Dr. Benjamin Collins
Prof. Thomas J. Downar
Dr. Jess Gehin
Dr. Andrew Godfrey
Aaron Graham
Daniel Jabaay
Blake Kelley
Dr. Kevin Clarno
Dr. Kang Seog Kim
Dr. Brendan Kochunas
Prof. Edward Larsen
Dr. Yuxuan Liu
Dr. Zhouyu Liu
Prof. William R. Martin
Dr. Scott Palmtag
Michael Rose
Thomas Saller
Dr. Shane Stimpson
Dr. Travis Trahan
Jipu Wang
Dr. Will Wieselquist
Mitchell T.H. Young
Ang Zhu



DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via US Department of Energy (DOE) SciTech Connect.

Website www.osti.gov

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website http://classic.ntis.gov/

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website http://www.osti.gov/contact.html

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Theory Manual

Version 2.0.0

March 11, 2015

CASL-U-2015-0078-000

Contributors (in alphabetical order)

- Dr. Benjamin Collins
- Prof. Thomas J. Downar
- Dr. Jess Gehin
- Dr. Andrew Godfrey
- Aaron Graham
- Daniel Jabaay
- Blake Kelley
- Dr. Kevin Clarno
- Dr. Kang Seog Kim
- Dr. Brendan Kochunas
- Prof. Edward Larsen
- Dr. Yuxuan Liu
- Dr. Zhouyu Liu
- Prof. William R. Martin
- Dr. Scott Palmtag
- Michael Rose
- Thomas Saller
- Dr. Shane Stimpson
- Dr. Travis Trahan
- Jipu Wang
- Dr. Will Wieselquist
- Mitchell T.H. Young
- Ang Zhu

Acknowledgements:

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 U.S. Department of Energy Contract No. DE-AC05-00OR22725.

Abstract

This theory manual describes the three-dimensional (3-D) whole-core, pin-resolved transport calculation methodology employed in the MPACT code. To provide sub-pin level power distributions with sufficient accuracy, MPACT employs the method of characteristics (MOC) solutions in the framework of a 3-D coarse mesh finite difference (CMFD) formulation. MPACT provides a 3D MOC solution, but also a 2D/1D solution in which the 2D planar solution is provided by MOC and the axial coupling is resolved by one-dimensional (1-D) lower order (diffusion or P_3) solutions.

In Chapter 2 of the manual, the MOC methodology is described for calculating the regional angular and scalar fluxes from the Boltzmann transport equation. In Chapter 3, the 2D/1D methodology is described, together with the description of the CMFD iteration process involving dynamic homogenization and solution of the multigroup CMFD linear system. A description of the MPACT depletion algorithm is given in Chapter 4, followed by a discussion of the subgroup and ESSM resonance processing methods in Chapter 5. The final Chapter 6 describes a simplified thermal hydraulics model in MPACT.

Contents

Chapter 1

Introduction

MPACT is a three-dimensional (3-D) whole core transport code that is capable of generating sub-pin level power distributions. This is accomplished by solving an integral form of the Boltzmann transport equation for the heterogeneous reactor problem in which the detailed geometrical configuration of fuel components, such as the pellet and cladding, are explitly retained. The cross section data needed for the neutron transport calculation are obtained directly from a multigroup cross section library, which has traditionally been used by lattice physics codes to generate few-group homogenized cross sections for nodal core simulators. Hence, MPACT involves neither *a priori* homogenization nor group condensation for the full core spatial solution.

The integral transport solution is obtained using the method of characteristics (MOC) [1], [9], and employs discrete ray tracing within each fuel pin. MPACT provides a 3-D MOC solution; however, for practical reactor applications, the direct application of MOC to 3-D core configuration requires considerable amounts of memory and computing time associated with the large number of rays. Therefore, an alternative approximate 3-D solution method is implemented in MPACT for practical full core calculations, based on a “2D/1D" method in which MOC solutions are performed for each radial plane and the axial solution is performed using a lower-order one-dimensional (1-D) diffusion or P_3 approximation. The core is divided into several planes, each on the order of 5-10 cm thick, and the planar solution is obtained for each plane using 2D MOC. The axial solution is obtained for each pin, and the planar and axial problems are coupled through a transverse leakage. The use of a lower order 1-D solution, which is most often the nodal expansion method (NEM) with the diffusion or P_3 approximation, is justified by the fact that most heterogeneity in the core occurs in the radial direction rather than the axial direction. Alternatively, a full 3D MOC solution can be performed, if the computational resources are available.

The Coarse Mesh Finite Difference (CMFD) acceleration method, which was originally introduced to improve the efficiency of the nodal diffusion method, is used in MPACT for the acceleration of the whole core transport calculation. The basic mesh in the CMFD formulation is a pin cell, which is much *coarser* than the flat source regions defined for MOC calculations (typically there are on the order of fifty (50) flat source regions in each fuel pin). The concept of dynamic homogenization of group constants for the pin cell is the basis for the effectiveness of the CMFD formulation to accelerate whole core transport calculations. The intra-cell flux distribution determined from the MOC calculation is used to generate the homogenized cell constants, while the MOC cell surface-averaged currents are used to determine the radial nodal coupling coefficients. The equivalence formalism makes it possible to generate the same transport solution with CMFD as the one obtained with the MOC calculation. In addition to the acceleration aspect of the CMFD formulation, it provides the framework for the 3-D calculation in which the global 3-D neutron balance is performed through the use of the MOC generated cell constants, radial coupling coefficients, and the NEM generated axial coupling coefficients.

The 2D/1D flux solver in MPACT thus consists of three primary modules: 2-D MOC, 1-D NEM, and 3-D CMFD. The 2-D MOC methodology, for solving the standard 2D linear Boltzmann (neutron transport) equation, is discussed in Chapter 2. The basic presentation of the 2D/1D methodology – in which transport theory is used to describe neutron transport in the radial directions of flight, but diffusion or P_3 theory is used to describe neutron transport in the axial direction – is presented in Chapter 3. The iteration strategy to solve the discrete 2D/1D equations, which uses 3-D CMFD with dynamic homogenization and relaxations, is also discussed in Chapter 3.

In Chapter 4, the MPACT depletion methodology is described. Chapter 5 discusses MPACT's treatment of energy resonance & self-shielding, and Chapter 6 outlines MPACT's simplified thermal hydraulics model.

Chapter 2

The Method of Characteristics

This chapter provides a detailed derivation of the MOC equations and introduces several of the concepts and algorithms used in the MPACT MOC solvers. First, a detailed derivation is provided, which highlights important approximations at each step. Next, the algorithm for the iterative solution of these equations is described. Then, the techniques required to discretize a problem that are common to any multi-dimensional MOC transport solver are described. The descriptions primarily focus on 2-D solvers but can be extended to 3-D.

2.1 Method of Characteristics for Solution Boltzmann Transport Equation

The derivation of the MOC solution to the Boltzmann neutron transport equation starts with the steady-state continuous form of the equation given in Eq. (2.1) below.

$$\begin{aligned}\Omega \cdot \nabla\varphi(\mathbf{r},\mathbf{\Omega},E) + \Sigma_t(\mathbf{r},\mathbf{\Omega},E) = \\ \frac{\chi(E)}{4\pi k_{\text{eff}}}\int_0^{\infty}\nu\Sigma_f(\mathbf{r},E')\int_0^{4\pi}\varphi(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE' + \\ \int_0^{\infty}\int_0^{4\pi}\Sigma_s(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\to E)d\Omega' dE'.\end{aligned} \tag{2.1}$$

The variable q is introduced to simplify the right hand side

$$q(\mathbf{r},\mathbf{\Omega},E)=\frac{\chi(E)}{4\pi k_{eff}}\int_0^\infty \nu\Sigma_f(\mathbf{r},E')\int_0^{4\pi}\varphi(\mathbf{r},\mathbf{\Omega}',E')d\Omega'dE'+ \int_0^\infty\int_0^{4\pi}\Sigma_s(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\rightarrow E)d\Omega'dE', \tag{2.2}$$

yeilding

$$\Omega\cdot\nabla\varphi(\mathbf{r},\mathbf{\Omega},E)+\Sigma_t(\mathbf{r},\mathbf{\Omega},E)=q(\mathbf{r},\mathbf{\Omega},E). \tag{2.3}$$

The method of characteristics is then applied in which the spatial and angle variables of the partial differential equation, Eq. (2.3), are transformed into the *characteristic direction* using the following identities.

$$\begin{aligned} & x(s)=x_0+s\Omega_x \\ \mathbf{r}=\mathbf{r}_0+s\mathbf{\Omega}\Rightarrow & y(s)=y_0+s\Omega_y. \\ & z(s)=z_0+s\Omega_z \end{aligned} \tag{2.4}$$

This leads to the characteristic form of Eq. (2.3) shown in Eq. (2.5).

$$\frac{d\varphi}{ds}\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E\right)+\Sigma_t\left(\mathbf{r}_0+s\mathbf{\Omega},E\right)\varphi\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E\right)=q\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E\right), \tag{2.5}$$

where

$$\begin{aligned} q(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E) =& \frac{\chi\left(\mathbf{r}_0+s\mathbf{\Omega},E\right)}{4\pi k_{\text{eff}}}\int_0^\infty\int_0^{4\pi}\nu\Sigma_f\left(\mathbf{r}_0+s\mathbf{\Omega}',E'\right)\varphi\left(\mathbf{r}_0+s\mathbf{\Omega}',\mathbf{\Omega}',E'\right)d\Omega'dE' \\ &+\int_0^\infty\int_0^{4\pi}\Sigma_s\left(\mathbf{r}_0+s\mathbf{\Omega}',\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\rightarrow E\right)\varphi\left(\mathbf{r}_0+s\mathbf{\Omega}',\mathbf{\Omega}',E'\right)d\Omega'dE'. \end{aligned} \tag{2.6}$$

Eq. (2.5) can then be solved analytically using the following integrating factor:

$$\exp\left(-\int_0^s\Sigma_t(\mathbf{r}_0+s'\mathbf{\Omega},E)ds'\right),$$

which leads to:

$$\begin{aligned} \varphi\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E\right) =& \varphi\left(\mathbf{r}_0,\mathbf{\Omega},E\right)\exp\left(-\int_0^s\Sigma_t(\mathbf{r}_0+s'\mathbf{\Omega},E)ds'\right) \\ &+\int_0^s q\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega},E\right)\exp\left(-\int_{s'}^s\Sigma_t(\mathbf{r}_0+s''\mathbf{\Omega},E)ds''\right)ds'. \end{aligned} \tag{2.7}$$

Eq. (2.7) is the solution of the characteristics form of the continuous Boltzmann transport equation. Next, this equation is discretized so that it may be solved numerically. Some reasonable approximations are introduced to accomplish this, and also so that the integrals of Eq. (2.7) may be evaluated more easily.

2.1.1 The Multigroup Approximation

The first approximation to be introduced, which is common for almost all deterministic methods, is the *multigroup approximation*. This discretizes the energy variable by defining discrete neutron energy groups. The multigroup cross sections are determined exactly by Eq. (2.8). However, $\varphi(\mathbf{r},\mathbf{\Omega},E)$ is generally not known *a priori*, therefore the approximation of Eq. (2.9) is introduced. The multigroup cross sections are then defined as shown in Eq. (2.10) using a weighting factor $\Psi(\mathbf{r},E)$ in energy. This weighting factor should typically be representative of the neutron energy spectrum of the system, which cannot be known exactly for all potential problems *a priori*. As long as the energy distribution of the neutron flux in the system to be simulated is reasonably consistent with the weighting spectrum used to collapse the continuous energy cross section, the multigroup approximation is accurate since it preserves the reaction rates within each energy group.

$$\Sigma_{x,g}(\mathbf{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_x(\mathbf{r},E)\varphi(\mathbf{r},\mathbf{\Omega},E)dE}{\int_{E_g}^{E_{g-1}} \varphi(\mathbf{r},\mathbf{\Omega},E)dE} \tag{2.8}$$

$$\varphi(\mathbf{r},\mathbf{\Omega},E) \approx \Psi(\mathbf{r},E)f(\mathbf{r},\mathbf{\Omega}) \tag{2.9}$$

$$\Sigma_{x,g}(\mathbf{r}) \approx \frac{\int_{E_g}^{E_{g-1}} \Sigma_x(\mathbf{r},E)\Psi(\mathbf{r},E)dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{r},E)dE} \tag{2.10}$$

$$\chi_g(\mathbf{r}) = \int_{E_g}^{E_{g-1}} \chi(\mathbf{r},E)dE \tag{2.11}$$

Eq. (2.11) does not use the weighting factor since the fission spectrum is not strictly a cross section. In Eq. (2.10), the subscript x is to indicate a reaction type. Applying the multigroup approximation to Eq. (2.7) leads to the steady-state multigroup MOC solution of Boltzmann neutron transport equation shown in Eq. (2.10), where the subscript g, is introduced to denote the neutron energy group index.

$$\begin{aligned}\varphi_g\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega}\right) &= \varphi_g\left(\mathbf{r}_0,\mathbf{\Omega}\right)\exp\left(-\int_0^s \Sigma_{t,g}(\mathbf{r}_0+s'\mathbf{\Omega})ds'\right)\\ &+\int_0^s q_g\left(\mathbf{r}_0+s\mathbf{\Omega},\mathbf{\Omega}\right)\exp\left(-\int_{s'}^s \Sigma_{t,g}(\mathbf{r}_0+s''\mathbf{\Omega})ds''\right)ds'.\end{aligned} \tag{2.12}$$

2.1.2 The Discrete Ordinates Approximation

The next approximation that is introduced after the multigroup approximation is the *discrete ordinates approximation* for the angular variable. This is essentially a quadrature approximation, which for a given function of angle is written as:

$$\int_{4\pi} f(\mathbf{\Omega})d\Omega \approx \sum_{m=1}^{M} w_m f(\mathbf{\Omega_m}). \tag{2.13}$$

Applying this approximation to Eq. (2.12) and Eq. (2.6) leads to:

$$\begin{aligned}\varphi_{g,m}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right) &= \varphi_{g,m}\left(\mathbf{r}_0\right)\exp\left(-\int_0^s \Sigma_{t,g}(\mathbf{r}_0 + s'\mathbf{\Omega}_m)ds'\right) \\ &+ \int_0^s q_{g,m}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right)\exp\left(-\int_{s'}^s \Sigma_{t,g}(\mathbf{r}_0 + s''\mathbf{\Omega}_m)ds''\right)ds',\end{aligned} \tag{2.14}$$

where

$$\begin{aligned}q_{g,m}(\mathbf{r}_0{+}s\mathbf{\Omega}_m, E) &= \frac{\chi_g\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right)}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G}\sum_{m'=1}^{M} \nu\Sigma_{f,g'}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right) w_{m'}\varphi_{g',m'}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right) \\ &+ \sum_{g'=1}^{G}\sum_{m'=1}^{M} w_m \Sigma_{s,g'\to g}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m, \mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m\right)\varphi_{g',m'}\left(\mathbf{r}_0 + s\mathbf{\Omega}_m\right).\end{aligned} \tag{2.15}$$

So long as the error introduced by Eq. (2.13) is minimal, the discrete ordinates approximation is valid. This approximation has been used extensively by the S_N and MOC methods, and in practice it is observed to be very accurate if a sufficient number of angles are used.

2.1.3 Constant Material Properties in a Discrete Region

To discretize the spatial domain, the problem is divided into arbitrarily shaped discrete regions. Within each region it is assumed that the material properties are constant with respect to the spatial variable. This spatial discretization, as illustrated by Fig. 2.1, essentially leads to a spatial discretization scheme that is first order accurate.

From these definitions and the constant material property, Eq. (2.14) and Eq. (2.15) are reduced to the following for each characteristic ray passing through each discrete region denoted by the subscripts k and i, respectively.

$$\varphi_{i,g,m,k}^{out} = \varphi_{i,g,m,k}^{in}\exp(-\Sigma_{t,i,g}s_{i,m,k}) + \int_0^{s_{i,m,k}} q_{i,g,m}(s')\exp\left(-\Sigma_{t,i,g}(s_{i,m,k} - s')\right)ds', \tag{2.16}$$

$$\begin{aligned}q_{i,g,m}(s) &= \frac{\chi_{i,g}}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\nu\Sigma_{f,i,g'}\sum_{m'=1}^{M} w_{m'}\varphi_{i,g',m'}(s) \\ &+ \sum_{g'=1}^{G}\sum_{m'=1}^{M} w_{m'}\Sigma_{s,i,g'\to g}(\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m)\varphi_{i,g',m'}(s), 0 \le s \le s_{i,m,k}.\end{aligned} \tag{2.17}$$

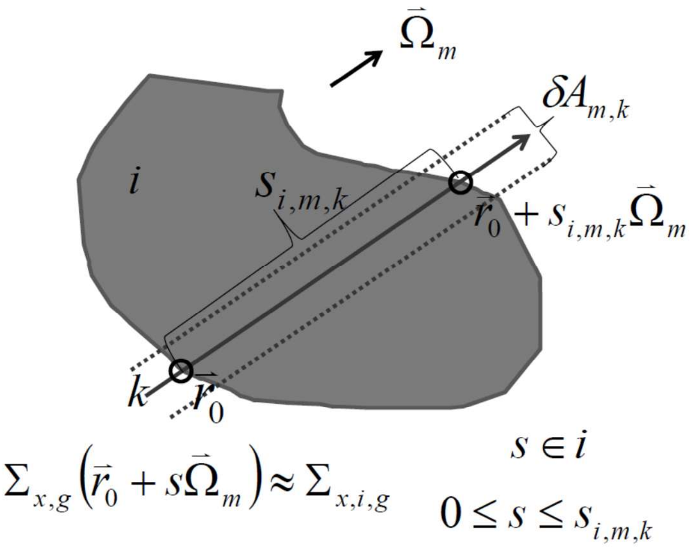


Figure 2.1: Spatial discretization with constant properties

In Eq. (2.16) the shorthand notation, $\varphi^{in}_{i,g,m,k} = \varphi_{i,g,m}(\mathbf{r}_0) = \varphi_{i,g,m,k}(s = 0)$ and $\varphi^{out}_{i,g,m,k} = \varphi_{i,g,m}(\mathbf{r}_0 + s_{i,m,k}\mathbf{\Omega}_m) = \varphi_{i,g,m,k}(s = s_{i,m,k})$, is used. For adjacent regions i and $i+1$ the identity, $\varphi^{out}_{i,g,m,k} = \varphi^{in}_{i+1,g,m,k}$, is also true.

Next the source, $q_{i,g,m}(s)$, is assumed to be constant within each discrete spatial region. This is commonly referred to as the *flat source* approximation. It is the lowest order approximation for the spatial dependence of the source and is accurate in the fine limit of the spatial mesh. Other approximations, such as linear and quadratic have been developed, but for the work here only the flat source is considered. With the flat source approximation, the remaining integral over s' can be evaluated analytically, which leads to Eq. (2.18) and Eq. (2.19):

$$\varphi^{out}_{i,g,m,k} = \varphi^{in}_{i,g,m,k}\exp(-\Sigma_{t,i,g}s_{i,m,k}) + \frac{q_{i,g,m}}{\Sigma_{t,i,g}}\left[1 - \exp(-\Sigma_{t,i,g}s_{i,m,k})\right], \tag{2.18}$$

$$q_{i,g,m} = \frac{\chi_{i,g}}{4\pi k_{\text{eff}}}\sum_{g'=1}^{G}\nu\Sigma_{f,i,g'}\sum_{m'=1}^{M}w_{m'}\overline{\varphi}_{i,g',m'} + \sum_{g'=1}^{G}\sum_{m'=1}^{M}w_{m'}\Sigma_{s,i,g'\to g}(\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m)\overline{\varphi}_{i,g',m'}. \tag{2.19}$$

Equation (2.19) introduces a new term, the region average angular flux, $\overline{\varphi}_{i,g,m}$. This is computed from the segment-average angular flux which is defined as:

$$\tilde{\varphi}_{i,g,m,k} = \frac{\int_0^{s_{i,m,k}}\varphi_{i,g,m}(s')ds'}{\int_0^{s_{i,m,k}}ds'} \Rightarrow \frac{\varphi^{in}_{i,g,m,k} - \varphi^{out}_{i,g,m,k}}{\Sigma_{t,i,g}s_{i,m,k}} + \frac{q_{i,g,m}}{\Sigma_{t,i,g}}. \tag{2.20}$$

The region average angular flux is then computed from the segment-average angular fluxes as shown in Eq. (2.21), where $\delta A_{m,k}$ denotes the cross-sectional area of the characteristic ray as illustrated in Fig. 2.1.

$$\overline{\varphi}_{i,g,m} = \frac{\sum\limits_{k\in i} \tilde{\varphi}_{i,g,m,k} s_{i,m,k} \delta A_{m,k}}{\sum\limits_{k\in i} s_{i,m,k} \delta A_{m,k}}. \tag{2.21}$$

Equations (2.18) and (2.20) are the fundamental discretized MOC equations that must be evaluated to obtain a solution of the angular flux for a given source q.

2.1.4 Scattering Source Approximation

The final approximation is the treatment of the scattering source. The usual way to represent the scattering source is to expand the differential scattering cross section in a series of Legendre polynomials of the cosine of the scattering angle.

In the first order, or P$_1$ scattering approximation, the source term of energy group g is approximated as:

$$\begin{aligned} q_g^n &= q_{g0}^n + q_{g1}^n \\ q_{g0}^n &= \frac{1}{4\pi}\left(\chi_g^n \sum_{g'} \nu\Sigma_{f,g'}^n \phi_{g'}^n + \sum_{g'} \Sigma_{s0,g'\to g}^n \phi_{g'}^n\right). \\ q_{g1}^n &= \frac{3}{4\pi}\left(\sum_{g'} \Sigma_{s1,g'\to g}^n \int_{4\pi} \Omega\cdot\Omega' \varphi_{g'}^n(\Omega') d\Omega'\right) \end{aligned} \tag{2.22}$$

Here, q_{g0}^n is the isotropic source that includes the fission and P$_0$ scattering source and q_{g1}^n is the anisotropic scattering source, and $\varphi_g^n(\Omega)$ is the regional angular flux that is determined by Eq. (2.14). The anisotropic scattering source in Eq. (2.19) can be further simplified by defining the regional currents in the x and y directions as:

$$q_{g1}^n(\Omega) = \frac{3}{4\pi}\left(\sum_{g'} \Sigma_{s1,g'\to g}^n (\Omega_x J_{g,x}^n + \Omega_y J_{g,y}^n)\right), \tag{2.23}$$

where

$$J_{g,x}^n \equiv \int_{4\pi} \Omega_x' \varphi_g^n(\Omega') d\Omega' \quad \text{and} \quad J_{g,y}^n \equiv \int_{4\pi} \Omega_y' \varphi_g^n(\Omega') d\Omega'.$$

The above two equations indicate that the regional currents besides the scalar flux must be stored in order to determine the anisotropic scattering source in P$_1$ scattering source approximation.

For many applications a suitable approximation based on the "transport correction" is applied which replaces an explicit treatment of anisotropic scattering by a reduced scattering cross section that is then taken to be isotropic. The transport corrected cross sections are calculated by merely subtracting the quantity $\bar{\mu}_0\Sigma_s$ from Σ_s, while in the scattering kernel the same quantity is subtracted from the diagonal elements. Here $\bar{\mu}_0$ is the average cosine of the scattering angles. The transport corrected cross sections can be expressed as:

$$\begin{aligned} \Sigma^{tr}_{s,g} &= (1-\bar{\mu}_0)\Sigma_{s,g} = \Sigma_{s0,g} - \Sigma_{s1,g} \\ \Sigma^{tr}_{s,g'\to g} &= \Sigma_{s0,g'\to g}(\text{for} g \neq g') \\ \Sigma^{tr}_{s,g\to g} &= \Sigma_{s0,g\to g} - \bar{\mu}_{0,g}\Sigma_{s0,g} = \Sigma_{s0,g\to g} - \Sigma_{s1,g} \end{aligned} \tag{2.24}$$

Here the P$_1$ anisotropic scattering cross section is the sum of the differential scattering cross sections, namely:

$$\Sigma_{s1,g} = \sum_{g'} \Sigma_{s1,g'\to g}. \tag{2.25}$$

The total cross section is now modified to the transport corrected one as:

$$\Sigma_{tr,g} = \Sigma_{t,g} - \Sigma_{s1,g}. \tag{2.26}$$

Assuming isotropic scattering allows the source term to be written in the form:

$$q_{i,g} = \frac{\chi_{i,g}}{4\pi k_{\text{eff}}} \sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'} + \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,i,g'\to g}\phi_{i,g'}, \tag{2.27}$$

where the scalar flux, $\phi_{i,g}$ is computed as:

$$\phi_{i,g} = \int_{4\pi} \overline{\varphi}_{i,g}(\mathbf{\Omega}) \approx \sum_{m=1}^{M} w_m \overline{\varphi}_{i,g,m}. \tag{2.28}$$

2.1.5 Iteration Scheme

In general the quantities of interest for reactor analysis such as reaction rates are determined from the scalar flux and not the angular flux, therefore the scalar flux is typically the primary solution variable updated by a transport kernel. Since any MOC kernel will make use of the above equations in some form, the kernel can be based on the concept of a 1-group fixed source problem which makes it possible to abstract the kernel into the following functional form:

$$\left(\varphi_g^{in,(n+1)}, \phi_g^{(n+1)}\right) = f\left(\varphi_g^{in,(n)}, \bar{\mathbf{q}}_g^{(n)}\right), \tag{2.29}$$

$$\bar{q}_{g,i}^{(n+1)} = \frac{1}{4\pi}\left(Q_{ext,i,g} + \Sigma_{s0,i,g\to g}\phi_{i,g}^{(n+1)}\right). \tag{2.30}$$

In Eq. (2.29), $\phi_g^{in,(n)}$ is a vector of the discrete incoming angular flux boundary conditions in all space and angle for a single group, g. $\bar{\mathbf{q}}_g^{(n)}$ is a vector of the 1-group source computed as shown in Eq. (2.30) for all regions and $\phi_g^{(n)}$ is a vector of the scalar fluxes for all regions. By alternately evaluating the function of Eq. (2.29) and updating the source defined by Eq. (2.30), it is possible to first define an *inner iteration* scheme to solve the 1-group fixed source problem for a given external source $Q_{ext,i,g}$. In these equations n is the inner iteration index and the algorithm for solving the 1-group fixed source problem is described by the process described in Algorithm 2.1.

Algorithm 2.1 Iterative algorithm for the MOC solution of 1-group fixed source problem

```
Guess initial source.
while not converged do
    Compute outgoing angular fluxes by evaluating Eq. (2.18) for all segments
    Compute segment-average angular flux by evaluating Eq. (2.20) for all segments.
    Compute region-wise angular flux by evaluating Eq. (2.21) for all regions
    Compute scalar flux by evaluating Eq. (2.28) for all regions
    Update 1-group source by evaluating Eq. (2.30) for all regions
end while
```

In the 1-group fixed source problem, the external source, $Q_{ext,i,g}$, is assumed to be known which is essentially the source from fission and *in-scatter*, or scattering from group g' to group g and it is a function of the scalar flux as shown in Eq.

$$Q_{ext,i,g} = \frac{\chi_{i,g}}{k_{\text{eff}}} \sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'} + \sum_{g'=1,g'\neq g}^{G} \Sigma_{s0,i,g'\to g}\phi_{i,g'}. \tag{2.31}$$

In Eq. (2.31), $1/k_{\text{eff}}$ is the eigenvalue of the system and must also be determined as a part of the solution. This is traditionally calculated using the power method, which is an iterative algorithm for finding the largest eigenvalue of a system that follows naturally from the source iteration scheme described thus far. Briefly, the general form of the eigenvalue value problem in reactor physics can be written in operator notation as:

$$\mathbf{T}\phi = \frac{1}{k_{\text{eff}}}\mathbf{F}\phi \tag{2.32}$$

where $\mathbf{F}$ represents the fission and $\mathbf{T}$ represents the streaming, absorption, and scattering of neutrons. Applying the power method to solve Eq. (2.32) results in the following iterative scheme:

$$\phi^{(l-1)} = \mathbf{T}^{-1}\frac{1}{k_{\text{eff}}^{(l)}}\mathbf{F}\phi^{(l)}, \tag{2.33}$$

$$k_{\text{eff}}^{(l+1)} = \frac{\|\mathbf{F}\phi^{(\mathbf{l+1})}\|_1}{\frac{1}{k_{\text{eff}}^{(l)}}\|\mathbf{F}\phi^{(l)}\|_1} \tag{2.34}$$

This iterative scheme used in the second level of iteration for the eigenvalue, which is referred to as the *outer* iteration. For the outer iteration, denoted by the index l, the total fission source is computed as shown in Eq. (2.35) and Eq. (2.31) is rewritten as shown in Eq. (2.36).

$$\Psi_i^{(l)} = \frac{1}{k_{\text{eff}}^{(l)}} \sum_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}^{(l)}, \tag{2.35}$$

$$Q_{ext,i,g}^{(l,n)} = \chi_{i,g}^{(l)}\Psi_i^{(l)} + \sum_{g'=1,g'\neq g}^{G} \Sigma_{s0,y,g'\to g}\phi_{i,g'}^{(l,n)}. \tag{2.36}$$

The equation to update the eigenvalue based on the power method is shown in Eq. (2.37), where V_i is the region volume.

$$k_{\text{eff}}^{(l+1)} = \frac{\sum\limits_{i=1}^{I} V_i \sum\limits_{g'=1}^{G} \nu\Sigma_{f,i,g'}\phi_{i,g'}^{(l+1)}}{\sum\limits_{i=1}^{I} V_i\Psi_i^{(l)}}. \tag{2.37}$$

The overall iterative procedure for solving the eigenvalue problem is shown in Algorithm 2.2 and is sometimes referred to as *source iteration*. In the source iteration technique described in this section, there is an inner iteration for the converging self-scattering and an outer iteration for the eigenvalue which is equivalent to the power iteration.

Algorithm 2.2 Iterative algorithm for the MOC solution of steady-state eigenvalue problem

```
Guess initial $k_{\text{eff}}$ and scalar flux.
while not converged do
    Compute total fission source by evaluating Eq. (2.36) for all regions
    for all groups do
        Compute 1-group source by evaluating Eq. (2.30) for all regions
        Solve 1-group fixed source problem with Algorithm 2.1
    end for
    Update $k_{\text{eff}}$ by evaluating Eq. (2.37) for all regions
end while
```

This iteration scheme has the advantage of reducing memory usage, by allowing the transport method to only allocate data for a single group, with the exception of the boundary condition. In the loop over groups in Algorithm 2.2 the in-scatter source of Eq. (2.31) is also updated in a Gauss-Seidel fashion. This helps to improve convergence for reactor problems since most LWRs are thermal reactors and the physics of the slowing down source involves primarily the down-scatter of neutrons.

2.2 Discretization of the Characteristics Equations

In Section 2.1 several different discretization methods were introduced for the different variables of the phase space. The discretization techniques for energy (the multigroup approximation) and space will not be discussed in this section since they are not specific to the MOC method, rather the focus here will be on the specific discretization techniques required by the MOC method.

In the method of characteristics, the fundamental way that the problem is discretized is to choose a set of rays that traverse the problem domain to represent the flight paths (characteristic tracks) of the neutrons. This is illustrated in Fig. 2.2. The end goal is to determine the segment lengths from each ray that pass through each discrete region, which are then used as the variable $s_{i,m,k}$ in the evaluation of Eq. (2.18), Eq. (2.20), and Eq. (2.21). In general, one may choose any set of rays so long as the intersection between the ray and the spatial region boundaries can be determined. At this point there are several design choices possible for the algorithm that performs the ray tracing.

The first consideration is whether or not to store the ray tracing data (e.g. the segment lengths and mapping of ray segment index to region index). This information could be computed on the fly as a ray is swept during the transport sweep, or it can be stored. The criterion that is used to make this choice is to minimize computational time, and the tradeoff is essentially increased memory storage versus repetitive computation. The choice that is almost invariably made for any MOC implementation is to compute the ray tracing information once and then to store it. This has the benefit of decoupling the sweep algorithm from the ray trace algorithm, which allows each to be further optimized and developed essentially independently of the other. Another benefit of storing the ray tracing data is that during a normal calculation one may perform on the order of 1000 sweeps, 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, so other design choices for the algorithm must be made to address this issue. Despite the memory requirements for storing the ray tracing data, this is the approach used in this work.

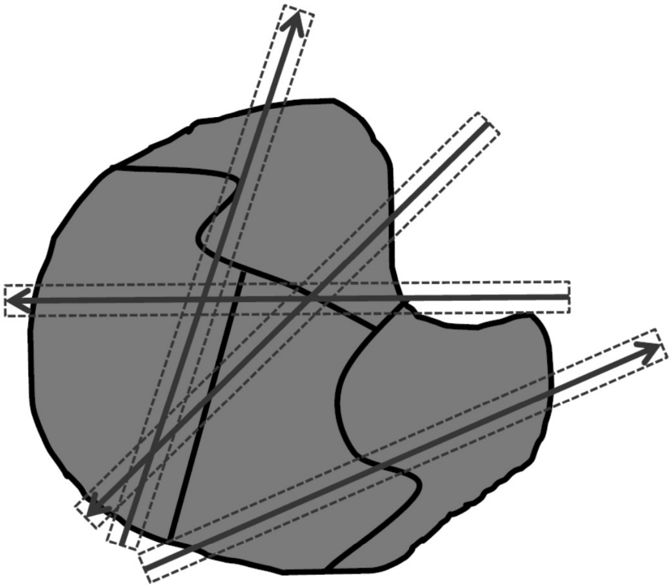


Figure 2.2: Characteristic rays intersection a set of discrete spatial regions

The next consideration is the choice of rays, specifically, which directions of flight should be considered. Since the discrete ordinates approximation described in Section 2.1.2 is used, it logically follows that these should be used for the directions of flight for the rays. In order 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. There are a myriad other considerations that go into developing a good quadrature, but this will not be discussed in great detail. The usual terms in the transport equation that are functions of angle and are integrated over angle are the angular flux and scattering cross section.

Once the directions of flight are chosen, it is then necessary to set up rays for a given angle. Ideally, one would want a few rays from each angle to intersect every spatial region in the problem, but at the very least a single ray should intersect each region. One of the most common design choices, which is used in MPACT, is to choose equally-spaced rays. The advantage of this choice is that it minimizes the need to store ray-dependent quantities such as the cross sectional area. Instead of storing the discrete cross sectional area of each characteristic ray in a problem, this quantity can simply be stored once for all rays or for all rays of a given angle. However, a potential disadvantage of uniform rays is one may place extra rays

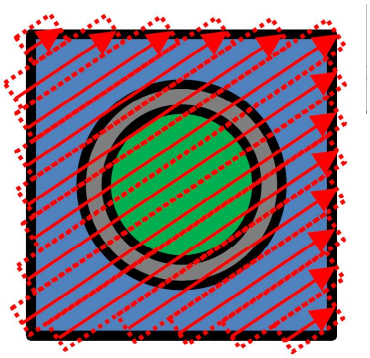


Figure 2.3: Uniformly spaced characteristic rays through a pin cell

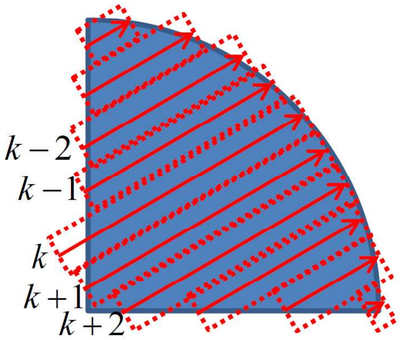


Figure 2.4: Numerical integration of a region volume by ray segments

in regions that do not necessarily require them. This is illustrated in Fig. 2.3.

Since the segment volumes in a given region represent a numerical integration of the region volume, which will have some error, the segment lengths within a given region are renormalized so that they integrate the region volume exactly. This is shown in Fig. 2.4 and Eq. (2.38).

$$\bar{s}_{i,m,k} = s_{i,m,k} \frac{V_i}{\sum_{k \in i} s_{i,m,k} \delta A_{m,k}}. \tag{2.38}$$

In the design of 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 a ray tracing algorithm for the entire domain by replicating this information for the entire core. The technique for this has several names but is referred to as *modular ray tracing* here and is illustrated in Fig. 2.5 which

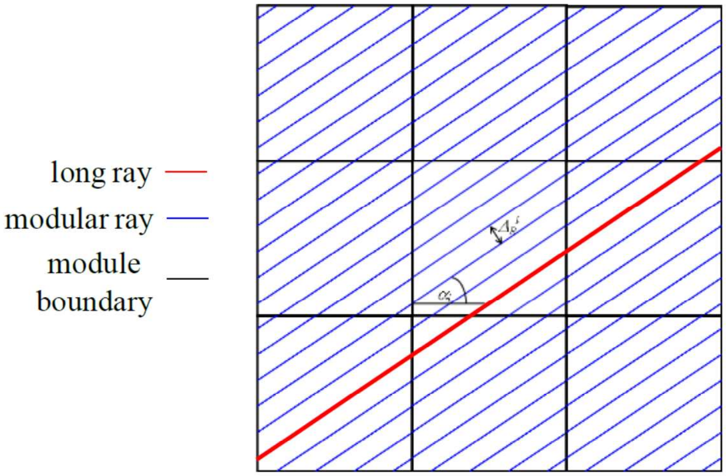


Figure 2.5: Modular ray tracing concept in 2-D

has nine *ray tracing modules* denoted as the black squares. The *modular rays*, depicted as blue lines, are defined only within the ray tracing module and connect at the ray tracing module boundaries. The *long ray* is shown by the red line and extends through the entire problem domain and consists of a particular sequence of modular rays.

It should be noted that the use of modular ray tracing introduces new requirements on the choice of the angles and also creates other subtle issues. The first requirement for modular ray tracing is that one be able to overlay a structured grid on their problem geometry. For the light water reactor problem this is a Cartesian grid, and ideally 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 intersections of rays on opposing surfaces of a ray tracing module. This second requirement is basically satisfied by choosing to have equally spaced rays within a given angle. 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 up to as much as $O(10^7)$ for a problem using pin modular or quarter pin modular ray tracing that is approximately as large as a full core PWR. Should quarter assembly modular ray tracing be used, then the savings in memory requirements could be as much as $O(10^5)$ for a full core PWR.

If modular ray tracing is used, then there is a second step in the setup to determine the long ray information for the entire problem domain. This information essentially makes it possible to connect the modular rays in order to traverse through the entire domain. Once the ray tracing is completed, the MOC transport sweeps

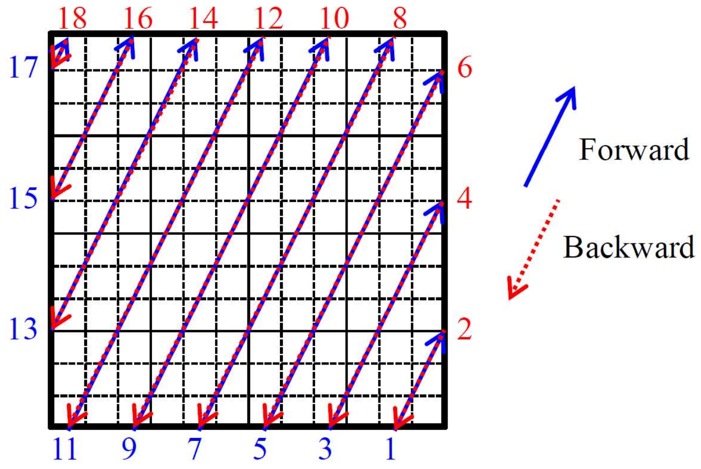


Figure 2.6: Sequential sweep algorithm

can be performed. MPACT employs a bi-directional sweep which loops over the angles and then within each angle loop over all the long rays. The advantage of this kind of sweep is that it becomes easier to order all the associated data structures to have good cache coherency. If modular ray tracing is being performed, then the construction of the long ray can be done once and swept for both the forward and backward direction (e.g. a bi- directional sweep), further improving performance through better cache coherency. Figure 2.6 illustrates how the rays can be swept sequentially.

The disadvantages for this kind of sweep ordering are; there is increased memory storage when there are reflective or periodic boundary conditions, and the convergence of the angular flux boundary condition will be more like Jacobi than Gauss-Seidel.

Chapter 3

The 2D/1D Method in MPACT

3.1 Introduction

The MPACT code is designed around a novel computational “2D/1D" method for simulating approximate solutions of the 3D linear Boltzmann equation. For simplicity, we describe here the 2D/1D method for the 3D monoenergetic Boltzmann equation with isotropic scattering in a homogeneous medium:

$$\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\Psi(\boldsymbol{x},\boldsymbol{\Omega})+\Sigma_t\Psi(\boldsymbol{x},\boldsymbol{\Omega})=\frac{\Sigma_s}{4\pi}\int_{4\pi}\Psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega'+\frac{\nu\Sigma_f}{4\pi k}\int_{4\pi}\Psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \ , \qquad \boldsymbol{x}\in V \ , \quad \boldsymbol{\Omega}\in 4\pi \ . \tag{3.1}$$

However, the extension of the 2D/1D method for Eq. (3.1) to general multigroup, anisotropic scattering problems in heterogeneous media is straightforward. (In the following discussion, certain important steps that must be followed in heterogeneous media are discussed.)

The notation in Eq. (3.1) is standard:

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

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

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

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

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

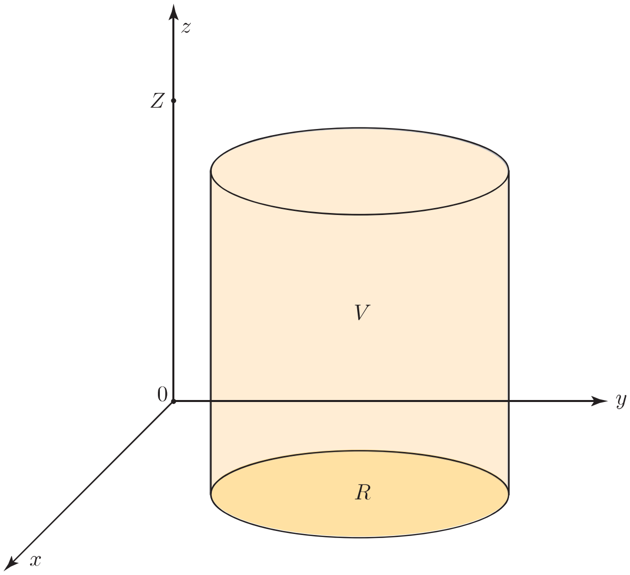


Figure 3.1: The 3D System V

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

$$\Psi(\boldsymbol{x}, \boldsymbol{\Omega}) = 0 \; , \quad \boldsymbol{x} \in \partial V \; , \quad \boldsymbol{\Omega} \cdot \boldsymbol{n} < 0 \; , \tag{3.2}$$

where $\boldsymbol{n}$ is the unit outer normal vector on ∂V. However, other boundary conditions (e.g. reflecting) are permitted.

3.2 The Basic 2D/1D Equations

The 2D/1D method does not directly solve Eq. (3.1); instead, it solves a somewhat elaborate system of equations that approximate Eq. (3.1). 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. Many different possibilities exist for approximating the axial transport process. Although the method in MPACT is not simple, it is probably about as simple as it could be.

To begin, a “2D radial" transport equation is formulated from Eq. (3.1) with an “isotropized" axial leakage term:

$$\boxed{\begin{aligned}\Omega_x\frac{\partial\psi}{\partial x}(\boldsymbol{x},\boldsymbol{\Omega})+\Omega_y\frac{\partial\psi}{\partial y}(\boldsymbol{x},\boldsymbol{\Omega})+\Sigma_t\psi(\boldsymbol{x},\boldsymbol{\Omega}) &= \frac{\Sigma_s}{4\pi}\int_{4\pi}\psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &+\frac{\nu\Sigma_f}{4\pi k}\int_{4\pi}\psi(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega'-\frac{1}{4\pi}\left[\frac{\partial J_z}{\partial z}(\boldsymbol{x})\right]\ .\end{aligned}} \tag{3.3}$$

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

$$J_z(\boldsymbol{x})\approx\int_{4\pi}\Omega_z\psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega\ ; \tag{3.4}$$

the explicit definition of J_z is given below [see Eqs. (3.12) and (3.14)]. The boundary condition for Eq. (3.3) consists of a vacuum boundary condition on the “cylindrical" portion of ∂V:

$$\boxed{\psi(\boldsymbol{x},\boldsymbol{\Omega})=0\ \text{ for }\ (x,y)\in\partial R\ ,\quad 0\le z\le Z\ .} \tag{3.5}$$

Operating on the solution of Eq. (3.3) by $\int_{4\pi}\Omega_x(\cdot)d\Omega$ and $\int_{4\pi}\Omega_y(\cdot)d\Omega$, we obtain the radial currents:

$$\boxed{\begin{aligned}J_x(\boldsymbol{x}) &= \int_{4\pi}\Omega_x\psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega\ , \\ J_y(\boldsymbol{x}) &= \int_{4\pi}\Omega_y\psi(\boldsymbol{x},\boldsymbol{\Omega})d\Omega\ .\end{aligned}} \tag{3.6}$$

Next, a “1D axial" transport equation is formulated from Eq. (3.1), with “isotropized" radial leakage terms described by J_x and J_y:

$$\begin{aligned}\mu\frac{\partial\hat{\psi}}{\partial z}(\boldsymbol{x},\boldsymbol{\Omega})+\Sigma_t\hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}) &= \frac{\Sigma_s}{4\pi}\int_{4\pi}\hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega' \\ &+\frac{\nu\Sigma_f}{4\pi k}\int_{4\pi}\hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega}')d\Omega'-\frac{1}{4\pi}\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x})+\frac{\partial J_y}{\partial y}(\boldsymbol{x})\right]\ .\end{aligned} \tag{3.7}$$

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

$$\hat{\psi}(\boldsymbol{x},\mu)=\int_0^{2\pi}\hat{\psi}(\boldsymbol{x},\boldsymbol{\Omega})d\omega\ , \tag{3.8}$$

we get:

$$\begin{aligned}\mu\frac{\partial\hat{\psi}}{\partial z}(\boldsymbol{x},\mu)+\Sigma_t\hat{\psi}(\boldsymbol{x},\mu) &= \frac{\Sigma_s}{2}\int_{-1}^{1}\hat{\psi}(\boldsymbol{x},\mu')d\mu' \\ &+\frac{\nu\Sigma_f}{2k}\int_{-1}^{1}\hat{\psi}(\boldsymbol{x},\mu')d\mu'-\frac{1}{2}\left[\frac{\partial J_x}{\partial x}(\boldsymbol{x})+\frac{\partial J_y}{\partial y}(\boldsymbol{x})\right]\ .\end{aligned} \tag{3.9}$$

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

$$\hat{\psi}(\boldsymbol{x},\mu) = 0 \ \text{ for } \begin{cases} z = Z \quad , \quad -1 \le \mu < 0 \,, \\ z = 0 \quad , \quad\ \ 0 < \mu \le 1 \,. \end{cases} \tag{3.10}$$

In MPACT, the axial transport Eqs. (3.9) and (3.10) are not directly solved. Instead, the simpler 1D P_1 (or P_3) approximation to these equations is formulated and solved, and the radial current J_z is expressed in terms of the approximate P_1 (or P_3) solution. The P_1 and P_3 approximations are defined next.

The standard P_1 approximation to Eqs. (3.9) and (3.10) is given by the 1D axial diffusion equation:

$$\boxed{-\frac{\partial}{\partial z}\frac{1}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial z}(\boldsymbol{x}) + \Sigma_a\hat{\phi}(\boldsymbol{x}) = \frac{\nu\Sigma_f}{k}\hat{\phi}(\boldsymbol{x}) - \left[\frac{\partial J_x}{\partial x}(\boldsymbol{x}) + \frac{\partial J_y}{\partial y}(\boldsymbol{x})\right] \,.} \tag{3.11a}$$

with boundary conditions:

$$\boxed{\begin{aligned} \hat{\phi}(x,y,Z) + \frac{2}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial x}(x,y,Z) = 0 \,, \quad (x,y)\in R \,, \\ \hat{\phi}(x,y,0) - \frac{2}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial x}(x,y,0) = 0 \,, \quad (x,y)\in R \,. \end{aligned}} \tag{3.11b}$$

After Eqs. (3.11) are solved, the axial current J_z is defined by:

$$\boxed{J_z(\boldsymbol{x}) = -\frac{1}{3\Sigma_t}\frac{\partial\hat{\phi}}{\partial z}(\boldsymbol{x}) \,.} \tag{3.12}$$

Next, the standard P_3 approximation to Eqs. (3.9) and (3.10) is given by the axial diffusion equations:

$$\boxed{\begin{aligned} -\frac{\partial}{\partial z}\frac{1}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(\boldsymbol{x}) + 2\hat{\phi}_2(\boldsymbol{x})\right] + \Sigma_a\hat{\phi}(\boldsymbol{x}) = \frac{\nu\Sigma_f}{k}\hat{\phi}(\boldsymbol{x}) - \left[\frac{\partial J_x}{\partial x}(\boldsymbol{x}) + \frac{\partial J_y}{\partial y}(\boldsymbol{x})\right] \,, \\ -\frac{\partial}{\partial z}\frac{9}{35\Sigma_t}\frac{\partial\hat{\phi}_2}{\partial z}(\boldsymbol{x}) + \Sigma_t\hat{\phi}_2(\boldsymbol{x}) = \frac{2}{5}\left[\Sigma_a\hat{\phi}(\boldsymbol{x}) - \lambda\nu\Sigma_f\hat{\phi}(\boldsymbol{x})\right] \,, \end{aligned}} \tag{3.13a}$$

with boundary conditions at $z = Z$:

$$\boxed{\begin{aligned} \hat{\phi}(x,y,Z) + \frac{5}{4}\hat{\phi}_2(x,y,Z) + \frac{2}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(x,y,Z) + 2\hat{\phi}_2(x,y,Z)\right] = 0 \,, \\ -\frac{1}{5}\hat{\phi}(x,y,Z) + \hat{\phi}_2(x,y,Z) + \frac{24}{35\Sigma_t}\frac{\partial\hat{\phi}_2}{\partial z}(x,y,Z) = 0 \,, \end{aligned}} \tag{3.13b}$$

and at $z = 0$:

$$\boxed{\begin{aligned} \hat{\phi}(x,y,0) + \frac{5}{4}\hat{\phi}_2(x,y,0) - \frac{2}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(x,y,0) + 2\hat{\phi}_2(x,y,0)\right] = 0 \,, \\ -\frac{1}{5}\hat{\phi}(x,y,0) + \hat{\phi}_2(x,y,0) - \frac{24}{35\Sigma_t}\frac{\partial\hat{\phi}_2}{\partial z}(x,y,0) = 0 \,. \end{aligned}} \tag{3.13c}$$

After Eqs. (3.13) are solved, the axial current J_z is defined by

$$\boxed{J_z(\boldsymbol{x}) = -\frac{1}{3\Sigma_t}\frac{\partial}{\partial z}\left[\hat{\phi}(\boldsymbol{x}) + 2\hat{\phi}_2(\boldsymbol{x})\right]\ .} \tag{3.14}$$

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

- The 2D/1D method with the P_1 axial leakage solver is defined by Eqs. (3.3), (3.6), (3.11a), and (3.12). These constitute a total of five equations for the five functions $\psi(\boldsymbol{x}, \boldsymbol{\Omega})$, $J_x(\boldsymbol{x})$, $J_y(\boldsymbol{x})$, $\hat{\phi}(\boldsymbol{x})$, and $J_z(\boldsymbol{x})$. The boundary conditions for the 2D radial transport equation and the 1D axial diffusion equation are Eqs. (3.5) and (3.11b).
- The 2D/1D method with the P_3 axial leakage solver is defined by Eqs. (3.3), (3.6), (3.13a), and (3.14). These constitute a total of six equations for the six functions $\psi(\boldsymbol{x}, \boldsymbol{\Omega})$, $J_x(\boldsymbol{x})$, $J_y(\boldsymbol{x})$, $\hat{\phi}(\boldsymbol{x})$, $\hat{\phi}_2(\boldsymbol{x})$, and $J_z(\boldsymbol{x})$. The boundary conditions for the 2D radial transport equation and the two 1D axial diffusion equations are Eqs. (3.5), (3.13b), and (3.13c).

The description of the underlying integro-differential equations and boundary conditions solved by the 2D/1D method in MPACT is now complete. These equations constitute a 2D radial transport equation, and one or two axial diffusion equations. The radial transport and axial diffusion equations are coupled through the isotropized axial and radial current terms. If the underlying problem is 2D with no axial variation, then the axial currents are zero, and the 2D/1D equations reduce to the correct 2D radial transport equation. If the underlying problem is 1D with no radial variation, then the radial currents are zero, and the 2D/1D equations reduce to the 1D axial P_1 or P_3 equations. Thus, the 2D/1D equations describe a process in which transport physics governs the radial variables, and P_1 or P_3 physics governs the axial variable.

3.3 Discretizations

In the 2D/1D equations, the energy variable is discretized using the conventional multigroup approximation. The details of this discretization are independent of the 2D/1D methodology and will not be discussed here. The key point is that in the derivation of the underlying multigroup 2D/1D equations from the multigroup transport equation, one obtains 2D/1D equations for each energy group that closely resemble the one-group 2D/1D equations described above. The only extra feature is that additional source terms appear, caused by neutrons that scatter into a given group g from other groups g'.

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 will be discussed elsewhere. In this report, the *order* of a quadrature set is denoted by N, and the number of direction vectors in the quadrature set is denoted by N_q. Thus, a quadrature set of order N consists of a set of N_q discrete 2D vectors $\mathbf{\Omega}_n = (\Omega_{x,n}, \Omega_{y,n})$ and weights w_n, with $1 \leq n \leq N_q$. The modular quadrature sets are used to angularly discretize the 2D radial transport Eq. (3.3), and to construct the radial currents in Eqs. (3.6).

It remains to describe the discretization of the spatial variables in the 2D/1D equations. In the following, we only discuss the 2D/1D equations with the P_1 axial leakage solver. (The P_3 axial leakage solver is similar; it involves two 1D axial diffusion equations rather than one.)

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

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

The 2D radial transport Eq. (3.3) and the 1D axial diffusion Eq. (3.11a) are now integrated over the k^{th} slice $z_{k-1/2} < z < z_{k+1/2}$. The integration of Eq. (3.3) produces a 2D radial transport equation for the axially-integrated flux

$$\psi_{n,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} \psi(x,y,z,\mathbf{\Omega}_n) dz \tag{3.15}$$

and the axial currents

$$J_{z,k\pm 1/2}(x,y) = J_z(x,y,z_{k\pm 1/2}) \; ; \tag{3.16}$$

this equation is:

$$\begin{aligned} \Omega_x \frac{\partial \psi_{n,k}}{\partial x}(x,y) + \Omega_y \frac{\partial \psi_{n,k}}{\partial y}(x,y) + \Sigma_t \psi_{n,k}(x,y) &= \frac{\Sigma_s}{4\pi} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m \\ + \frac{\nu \Sigma_f}{4\pi k} \sum_{m=1}^{N_q} \psi_{m,k}(x,y) w_m &- \frac{1}{4\pi \Delta z_k} \left[J_{z,k+1/2}(x,y) - J_{z,k-1/2}(x,y) \right] \; . \end{aligned} \tag{3.17}$$

The integration of the diffusion Eq. (3.11a) produces a 1D balance equation for the axially-integrated flux:

$$\hat{\phi}_k(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} \hat{\phi}(x,y,z) dz \; , \tag{3.18}$$

and the axially-integrated radial currents:

$$J_{x,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} J_x(x,y,z) dz \; , \tag{3.19a}$$

$$J_{y,k}(x,y) = \frac{1}{\Delta z_k} \int_{z_{k-1/2}}^{z_{k+1/2}} J_y(x,y,z) dz \; . \tag{3.19b}$$

Using Eq. (3.12), this balance equation is:

$$\begin{aligned}\frac{1}{\Delta z_k}\left[J_{z,k+1/2}(x,y)-J_{z,k-1/2}(x,y)\right]+\Sigma_a\hat{\phi}_k(x,y)\\ =\frac{\nu\Sigma_f}{k}\hat{\phi}_k(x,y)-\left[\frac{\partial J_{x,k}}{\partial x}(x,y)+\frac{\partial J_{y,k}}{\partial y}(x,y)\right]\ .\end{aligned} \tag{3.20}$$

We note that operating on Eq. (3.17) by

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

we obtain

$$\begin{aligned}\frac{\partial J_{x,k}}{\partial x}(x,y)+\frac{\partial J_{y,k}}{\partial y}(x,y)+\Sigma_a\phi_k(x,y)\\ =\frac{\nu\Sigma_f}{k}\phi_k(x,y)-\frac{1}{\Delta z_k}\left[J_{z,k+1/2}(x,y)-J_{z,k-1/2}(x,y)\right]\ .\end{aligned} \tag{3.21}$$

where

$$\phi_k(x,y)=\sum_{m=1}^{N_q}\psi_{m,k}(x,y)w_n\ . \tag{3.22}$$

Since Eqs. (3.20) and (3.21) have identical leakage terms, it follows that

$$\phi_k(x,y)=\hat{\phi}_k(x,y)\ . \tag{3.23}$$

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

Eq. (3.17) 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 determined, Eq. (3.17) completely determines $\psi_{n,k}(x,y)$.

However, Eqs. (3.11), (3.12), and (3.20) are insufficient for determining $\hat{\phi}_k(x,y)$ and $J_{z,k\pm1/2}(x,y)$. In MPACT, two types of extra equations have been considered: a finite difference method, and a nodal method. The simpler and less accurate finite difference method uses the Fick's law relation

$$J_{z,k+1/2}(x,y)=-\frac{1}{3\Sigma_t\Delta z_{k+1/2}}\left[\phi_{k+1}(x,y)-\phi_k(x,y)\right]\ ,$$

where

$$\Delta z_{k+1/2}=\frac{1}{2}\left(\Delta z_{k+1}+\Delta z_k\right)\ .$$

The nodal method uses other more sophisticated formulations based on standard nodal diffusion expansions. These have been employed in the nuclear industry for years, and require no modification for use in the 2D/1D method. The key fact is that all the axial discretization methods that have been considered make use of the balance Eq. (3.20). In principle, any reasonable axial diffusion solver could be implemented in MPACT without changing the overall structure of the code, as long as the method employs the balance Eq. (3.20).

It remains to describe the discretizations of Eqs. (3.17) and (3.20) in the radial spatial variables x and y. These discretizations are summarized next.

First, the 2D region R is divided into a Cartesian “coarse grid" (typically, one coarse cell = one pin cell) and an unstructured “fine grid" (used to discretize a pin cell; typically, about 40 fine cells per coarse cell are used).

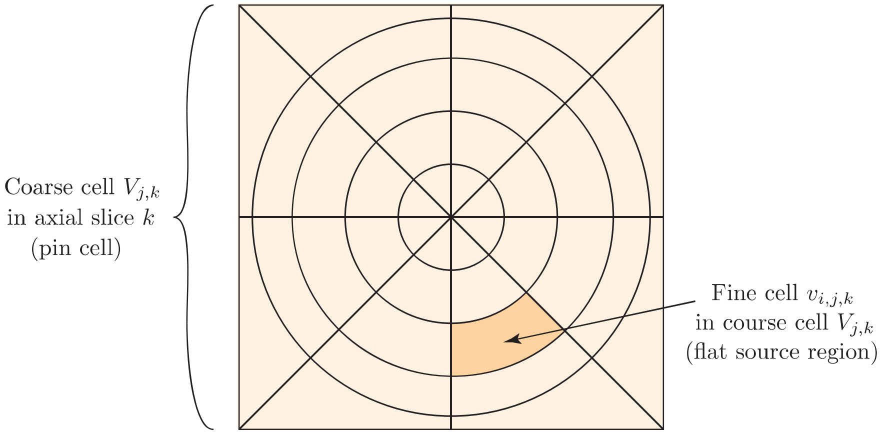


Figure 3.2: Fine and Coarse Spatial Cells

Second, the axial diffusion Eq. (3.20) is integrated radially over each coarse cell. Using radially coarse-cell flux-weighted cross sections (the origin of these cross sections is described in the next paragraph), the equation becomes one in which the unknowns are averaged radially (i.e. are constant) over each coarse cell. More precisely, the solution of the fully discrete axial diffusion equation depends on k (the slice index) and j (the coarse cell index), but not in 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. (3.17) is discretized on the fine radial grid for each coarse cell $V_{j,k}$ using the method of characteristics (MOC). However, the axial current terms in this equation are obtained from the axial diffusion calculation, and these are constant over the coarse cell. Thus, in coarse cell $V_{j,k}$, the axial current terms depend on j and k, but not on i. The MOC method for Eq. (3.17) utilizes balance equations obtained by integrating Eq. (3.17) over each fine cell $v_{i,j,k}$. Summing these balance equations over all i (i.e., over all fine cells in a single coarse cell), we obtain the balance equation for the coarse cell. Operating on this coarse cell balance equation by

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

we obtain the angularly-integrated balance equation 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, obtained by integrating the axial diffusion Eq. (3.20) over the cell. In addition, the flux-weighted coarse-cell homogenized cross sections used in the axial diffusion calculation are obtained from the 2D radial transport calculations.

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

3.4 Iteration Strategy

The iteration strategy used to solve the discretized 2D/1D equations involves (i) radial sweeps of the 2D radial transport equation, (ii) axial “sweeps" of the 1D axial diffusion equation, and (iii) solving a low-order coarse-mesh 3D CMFD “diffusion" equation. The discretized 2D radial transport equation and 1D axial diffusion equations are described above, but the low-order coarse-mesh 3D CMFD (or, more simply, CMFD) equation has not yet been described. We do this next.

The essential element in the derivation of the CMFD equation is the angularly-integrated coarse-mesh balance equation. To simplify the discussion, we consider a 1D system with coarse meshes labeled sequentially by the index j. Thus, the coarse cell to the left of cell j is labeled $j-1$, and the coarse cell to the right of cell j is cell $j+1$. However, because this detail is important, we do not assume that the cross sections are independent of j. The left and right edges of cell j are labeled $j-1/2$ and $j+1/2$. The balance equation for coarse cell j is:

$$\frac{1}{\Delta_j}\left(J_{j+1/2} - J_{j-1/2}\right) + \Sigma_{t,j}\Phi_k = \Sigma_{s,j}\Phi_j + \frac{\nu\Sigma_{f,j}}{k}\Phi_j \ , \tag{3.24}$$

where Φ_j is the course-mesh scalar flux for cell j, $J_{j\pm1/2}$ are the currents on the right and left edges of cell j, and Σ_j denotes the flux-weighted cross sections for cell j. (The actual 3D coarse mesh balance equation used in MPACT would have six current terms rather than two.)

Now each current term is written:

$$J_{j+1/2} = -\frac{\Phi_{j+1} - \Phi_j}{3\Sigma_{t,j+1/2}\Delta_{j+1/2}} + \left[\frac{J_{j+1/2} + \frac{\Phi_{j+1}-\Phi_j}{3\Sigma_{t,j+1/2}\Delta_{j+1/2}}}{\Phi_{j+1} + \Phi_j}\right]\left(\Phi_{j+1} + \Phi_j\right) ,$$

or as

$$J_{j+1/2} = -\frac{\Phi_{j+1} - \Phi_j}{3\Sigma_{t,j+1/2}\Delta_{j+1/2}} + \hat{D}_{j+1/2}\left(\Phi_{j+1} + \Phi_j\right) , \tag{3.25a}$$

where

$$\hat{D}_{j+1/2} = \frac{J_{j+1/2} + \frac{\Phi_{j+1}-\Phi_j}{3\Sigma_{t,j+1/2}\Delta_{j+1/2}}}{\Phi_{j+1} + \Phi_j} , \tag{3.25b}$$

and

$$\Sigma_{t,j+1/2}\Delta_{j+1/2} = \frac{1}{2}\left[\Sigma_{t,j+1}\Delta_{j+1} + \Sigma_{t,j}\Delta_j\right] . \tag{3.25c}$$

Inroducing Eq. (3.25a) into Eq. (3.24), we obtain the foliowing low-order coarse-grid CMFD “diffusion" equation:

$$\begin{aligned} -\frac{1}{3\Sigma_{t,j+1/2}\Delta_{j+1/2}}\left(\Phi_{j+1} - \Phi_j\right) &+ \frac{1}{3\Sigma_{t,j-1/2}\Delta_{j-1/2}}\left(\Phi_j - \Phi_{j-1}\right) \\ +\big[\hat{D}_{j+1/2}(\Phi_{j+1} + \Phi_j) &- \hat{D}_{j-1/2}(\Phi_j + \Phi_{j-1}) + \Sigma_{t,j}\Phi_j\big] \\ &= \Sigma_{s,j}\Phi_j + \lambda\nu\Sigma_{f,j}\Phi_j . \end{aligned} \tag{3.26}$$

This equation holds at all “interior" coarse cells, which do not have a boundary on the outer edge of the system. The derivation of the proper equation for coarse cells that adjoin the outer boundary of V is straightforward.

In the CMFD method, estimates for the coarse cell-edge currents and coarse cell-averaged scalar fluxes are used in Eq. (3.25b) to construct a “lagged" estimate of $\hat{D}_{j+1/2}$. Using the lagged estimates of $\hat{D}_{j+1/2}$,

- Eq. (3.26) is solved to determine new estimates of Φ_j and eigenvalue k, and:
- The new estimates of Φ_j and the lagged estimates of $\hat{D}_{j+1/2}$ are used in Eq. (3.25a) to determine new estimates of $J_{j+1/2}$.

We can now describe the iterative CMFD process used in MPACT. An iteration is outlined in the following figure, and is explained in detail 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.)

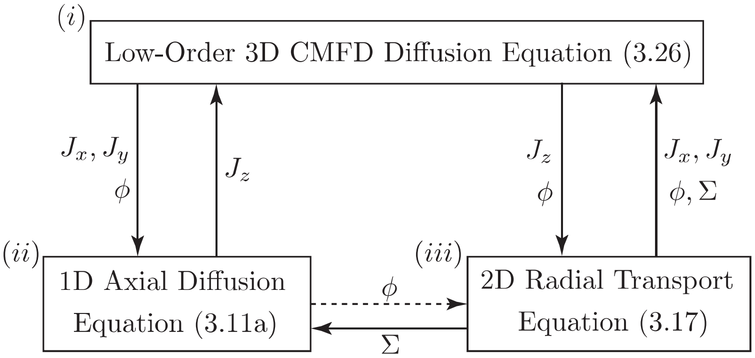


Figure 3.3: MPACT Iteration Strategy

(i) First, the $\hat{D}$ terms are calculated from Eq. (3.25b) using coarse grid scalar flux and current estimates obtained from the previous 1D axial diffusion and 2D radial transport calculations. (At the beginning of the first iteration, the $\hat{D}$ terms are estimated to be zero.) Then, a global 3D low-order CMFD diffusion calculation [solution of Eq. (3.26)] is performed. In this calculation, the homogenized flux-weighted cross sections are obtained from the previous 2D radial transport calculation. 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. The radial current and scalar flux estimates are sent to the “diffusion" processors that will perform the 1D axial diffusion equations, and the axial currents and scalar fluxes are sent to the “transport" processors that will perform the 2D radial transport sweeps.

(ii) Next, a “solve" of the 1D axial diffusion equations is performed. This “solve" can consist of local calculations that involve no axial sweeping; in this case, relaxation is needed to stabilize the iterations for small Δz. Alternatively, the “solve" can consist of several axial sweeps, to partly converge the discrete 1D diffusion equations. (If this sweeping is done, relaxation is most likely not needed.) This calculation results in new coarse-cell scalar fluxes, and new estimates of the axial currents. The axial currents are sent to the “CMFD" processors, and the scalar fluxes are sent to the “transport" processors if the axial sweeping in the “diffusion" processors is performed. If the axial sweeping in the diffusion processors is not performed, the scalar fluxes are not sent to the “transport" processors.

(iii) Next, The “transport" processors use the most recent available coarse-grid scalar flux estimates to update (renormalize) their estimates of the fine-grid scalar fluxes. Then, using the axial current information from the CMFD processors, these processors perform a specified number (typically, three) of 2D sweeps on each axial slice, 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 sent to the “CMFD" processors. Also, new estimates of the coarse-grid flux-weighted cross sections are sent to the “CMFD" and the “diffusion" processors; these homogenized cross sections are used in all the coarse-grid calculations.

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

3.5 Appendix A: Relaxation Strategy

One of the deficiencies of the early 2D/1D method (in the DeCART code) was the instability of the iteration method for problems having thin axial planes. To understand and prevent this issue, we performed a Fourier analysis of the iteration method using a simple finite difference axial diffusion solver [11], [10]. We found that by properly under-relaxing the iterations, we could guarantee unconditional stability with a source iteration-like scheme for the discretized system. These Fourier analysis results were confirmed in numerical simulations. Later, we demonstrated (again, with Fourier analysis and numerical testing) that the CMFD-accelerated iterations could be stabilized using under-relaxation [12]. In the following, we provide the results of these Fourier analyses.

Our Fourier analysis is based on the following 2D/1D equation – which uses a simple Fick’s Law finite difference approximation to describe the axial currents at the top and bottom of the k^{th} slice, in terms of the slice-averaged fluxes in the slices directly above and below that slice edge:

$$\begin{aligned}\left(\Omega_x\frac{\partial}{\partial x}+\Omega_y\frac{\partial}{\partial y}+\Sigma_{t,k}\right)\psi_k(x,y,\mathbf{\Omega}) &= \frac{1}{4\pi}\Bigg\{\Sigma_{s,k}\phi_k(x,y)+Q_k(x,y)\\ &+\frac{1}{\Delta_k}\left[\frac{D_{k+1/2}}{\Delta_{k+1/2}}\Big(\phi_{k+1}(x,y)-\phi_k(x,y)\Big)-\frac{D_{k-1/2}}{\Delta_{k-1/2}}\Big(\phi_k(x,y)-\phi_{k-1}(x,y)\Big)\right]\Bigg\}\,, \end{aligned}\tag{3.27a}$$

$$\phi_k(x,y)=\int\psi_k(x,y,\mathbf{\Omega}')d\Omega'\;. \tag{3.27b}$$

To solve this equation, we consider what is likely the simplest possible “Source Iteration" scheme that can be made stable for all $\Delta_z > 0$. Noting that the right side of the above equation depends only on the scalar fluxes ϕ in adjacent axial slices, we consider a simple 2D sweep on each slice to update the scalar flux:

$$\left(\Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Sigma_{t,k}\right) \psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega}) = \frac{1}{4\pi}\Bigg\{\Sigma_{s,k}\phi_k^{(\ell)}(x,y) + Q_k(x,y)$$

$$+\frac{1}{\Delta_k}\left[\frac{D_{k+1/2}}{\Delta_{k+1/2}}\Big(\phi_{k+1}^{(\ell)}(x,y) - \phi_k^{(\ell)}(x,y)\Big) - \frac{D_{k-1/2}}{\Delta_{k-1/2}}\Big(\phi_k^{(\ell)}(x,y) - \phi_{k-1}^{(\ell)}(x,y)\Big)\right]\Bigg\} , \quad (3.28a)$$

$$\phi_k^{(\ell+1/2)}(x,y) = \int \psi_k^{(\ell+1/2)}(x,y,\mathbf{\Omega}')d\Omega' , \quad (3.28b)$$

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

$$\phi_k^{(\ell+1)}(x,y) = \theta\phi_k^{(\ell+1/2)}(x,y) + (1-\theta)\phi_k^{(\ell)}(x,y) . \quad (3.28c)$$

In Fourier-analyzing this method, we do not treat Eq. 3.28 with any angular or radial spatial discretizations; our experience is that although 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. 3.28c is to be determined; if $\theta = 1$, the method defined by Eqs. 3.28 is basically Source Iteration. We note that 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 $1 \leq k \leq K$ to sufficiently “communicate." For an infinite, homogeneous medium with uniform $\Delta_k = \Delta_z$, the iterative performance of the above method can be assessed by a Fourier analysis. Referring to the papers [11] and [12] for details, we find that for $\theta = 1$ (the standard “Source Iteration" method), the growth factor ω is bounded from above and below by

$$\omega_{min} \leq \omega \leq \omega_{max} , \quad (3.29)$$

where

$$\omega_{max} = c , \quad (3.30a)$$

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

$$\omega_{min} = \begin{cases} 0 & , \quad \Sigma_t\Delta_z \geq \frac{2}{\sqrt{3c}} , \quad (\text{“large"}\ \Delta_z) , \\ c - \frac{4}{3(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z < \frac{2}{\sqrt{3c}} , \quad (\text{“small"}\ \Delta_z) , \end{cases} \quad (3.30b)$$

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 $\rho \geq 1$, is stable but slowly converging when $\rho < 1$ and $\rho \approx 1$, and is stable and rapidly converging when $\rho \ll 1$.

Using Eq. 3.28c, we find that the Fourier analysis for $0 \leq \theta \leq 1$ yields

$$\theta\omega_{min} + 1 - \theta \leq \omega \leq \theta\omega_{max} + 1 - \theta \ . \tag{3.31}$$

For $\theta = 1$ (the “Source Iteration" method originally in DeCART), Eqs. 3.30 and 3.31 give

$$\rho = |\omega|_{max} = \begin{cases} c & , \quad \sqrt{\frac{2}{3c}} < \Sigma_t\Delta_z \ , \\ \frac{4}{3(\Sigma_t\Delta_z)^2} - c & , \quad \Sigma_t\Delta_z \leq \sqrt{\frac{2}{3c}} \ . \end{cases} \tag{3.32}$$

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. 3.31 is the value for which the left and right sides are equal in magnitude but opposite in sign:

$$\theta_{opt}\omega_{min} + 1 - \theta_{opt} = -\left[\theta_{opt}\omega_{max} + 1 - \theta_{opt}\right] \ . \tag{3.33a}$$

Thus,

$$\theta_{opt} = \frac{2}{2 - (\omega_{max} + \omega_{min})} \ , \tag{3.33b}$$

and then

$$\begin{aligned} \rho = |\omega|_{max} &= \theta_{opt}\omega_{max} + 1 - \theta_{opt} \\ &= \frac{\omega_{max} - \omega_{min}}{2 - (\omega_{max} + \omega_{min})} \ . \end{aligned} \tag{3.33c}$$

Combining Eqs. 3.30 and 3.33, we obtain:

$$\theta_{opt} = \begin{cases} \frac{2}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t\Delta_z \ , \\ \frac{3(\Sigma_t\Delta_z)^2}{2+3(1-c)(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z \leq \frac{2}{\sqrt{3c}} \ , \end{cases} \tag{3.34a}$$

$$\rho = \begin{cases} \frac{c}{2-c} & , \quad \frac{2}{\sqrt{3c}} < \Sigma_t\Delta_z \ , \\ \frac{2}{2+3(1-c)(\Sigma_t\Delta_z)^2} & , \quad \Sigma_t\Delta_z \leq \frac{2}{\sqrt{3c}} \ . \end{cases} \tag{3.34b}$$

Equation 3.34b shows that the iterative method defined by Eqs. 3.28 with θ defined by Eq. 3.34a is stable for all scattering ratios $0 \leq c \leq 1$ and all axial grids $\Delta_z > 0$. Like standard Source Iteration applied to the

S_N equations, this method becomes slowly converging as $c \to 1$. It also becomes slowly converging as $\Delta_z \to 0$. However, like Source Iteration, it does not become unstable. However, the DeCART-like method with $\theta = 1$ is stable only for sufficiently large axial grids.

CMFD acceleration can be applied to the above Source iteration method, 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 converge the transport solution. However, before applying a Fourier analysis, it is necessary to linearize this method. From previous work, we know that the linearized form of CMFD is equivalent to Diffusion Synthetic Acceleration (DSA) [13]. The DSA equivalent of our CMFD system yields an update term $\delta\phi_k^{(\ell+1/2)}$ for the scalar flux. (We refer the reader to [12] for an explicit definition of this correction term.) A CMFD iteration then consists of

(i) an axial sweep [Eq. (3.28a)],
(ii) calculation of $\phi_k^{\ell+1/2}$ [Eq. (3.28b)],
(iii) the CMFD calculation of $\delta\phi_k^{(\ell+1/2)}$ (see [12]), and
(iv) the relaxed update equation

$$\phi_k^{(\ell+1)}(x,y) = \theta\Big(\phi_k^{(\ell+1/2)}(x,y) + \delta\phi_k^{(\ell+1/2)}\Big) + (1-\theta)\phi_k^{(\ell)}(x,y)\ . \tag{3.35}$$

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

$$\omega_1(\lambda_r,\zeta) = 1 - \frac{1-\zeta I_0(\lambda_r)}{1-\zeta\left(1+\frac{\lambda_r^2}{3}\right)^{-1}}\ , \tag{3.36a}$$

$$\zeta = c - \frac{\Lambda_z^2}{3}\ , \tag{3.36b}$$

$$\Lambda_z = \frac{\sin(\lambda_z\Sigma_t\Delta_z/2)}{\Sigma_t\Delta_z/2}\ . \tag{3.36c}$$

We wish to determine the maximum and minimum values of $\omega_1(\lambda_r,\zeta)$, for $0 \le \lambda_r < \infty$ and $0 \le \Lambda_z \le (\Sigma_t\Delta_z/2)^{-1}$, in order to use these in Eq. (3.33b) to obtain the optimum relaxation factor. Unfortunately, extremum values of ω cannot be determined explicitly, given the form of Eqs. (3.36). Evaluating this function numerically, these extrema can be seen in Figure 3.4. It is seen that the extremum of the growth factor decrease monotonically with the parameter ζ, with the maximum value (occurring for $\zeta = c = 1$) being $\omega \approx 0.2247$ (this is the standard result from traditional CMFD).

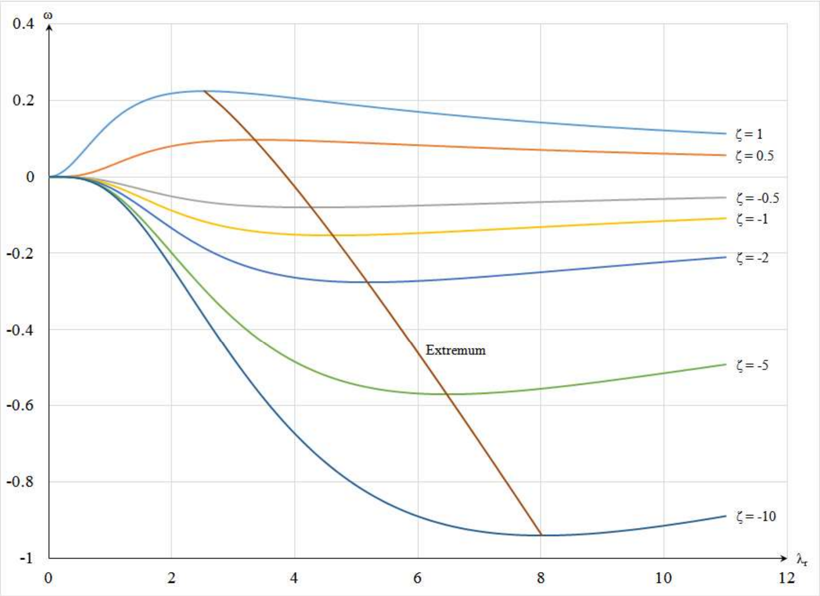


Figure 3.4: Growth Factor ω for Various Radial Frequencies λ_r for a Sample of Parametrized Values ζ

This figure 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:

$$\begin{gathered} \lambda_{ext} \approx S + M(F-\zeta)^P \ , \\ S = 1.93801895412889 \ , \quad M = 1.88037759461481 \ , \\ F = 1.07821249297909 \ , \quad P = 0.4879758371396750 \ . \end{gathered} \tag{3.37}$$

Now that λ_{ext} is “known", ω_{min} can be calculated from Eqs. (3.36), and thus θ_{opt} can be calculated from Eq. (3.33b).

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

$$\Sigma_t \Delta_z \gtrapprox \frac{2}{\sqrt{3(c + 10.947)}} \ . \tag{3.38}$$

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

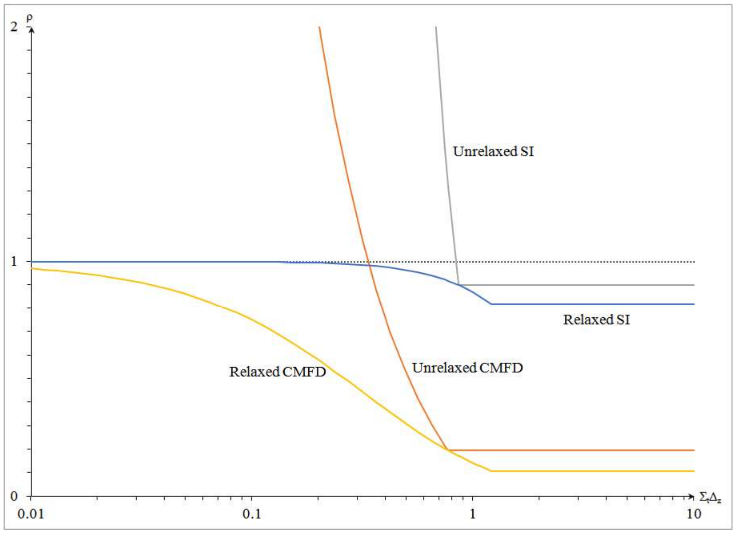


Figure 3.5: Spectral Radius ρ Comparison of SI and CMFD vs Optical Thickness $\Sigma_t \Delta_z$ with Scattering Ratio $c = 0.9$

Figure 3.5 illustrates that the region of conditional stability for the unrelaxed CMFD method is much larger than that of unrelaxed source iteration. This improvement comes from the improved communication between the planes — at each iterate, the planes receive information from the whole system, rather than just neighboring planes. The figure also shows that (i) the optimally relaxed SI scheme is unconditionally stable, (ii) the optimally relaxed CMFD method is unconditionally stable, and (iii) the relaxed CMFD method is significantly more efficient that the relaxed SI method. When the axial slice thicknesses tend to 0, the optimally relaxed SI and CMFD methods both become increasingly slowly converging. However, the CMFD method always converges much faster than SI, and for practical problems, the axial slice thicknesses are not small enough to be a practical concern.

The above analysis applies specifically to the 2D/1D Eqs. (3.27), 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. We have found that when these nodal solvers are used instead of the finite-difference method described above, the resulting discrete methods are again stabilized, using the relaxation parameters defined above.

The situation becomes more complicated with the nodal diffusion solvers, because sometimes axial “sweeps" are performed, which effectively couple each axial slice to more than just its nearest neighbors.

These “one-node" or “two-node" “sweeps" are performed to increase the stability and improve the convergence of the method, and they typically accomplish this. Presently, these “sweeps" are combined with the under-relaxation step described above. It is not clear whether the relaxation parameters should be changed, and if so, how, due to the utilization of the “one-node" and “two-node" sweeps. However, the important point to make is that the iterations in MPACT are stable, and the stabilization mechanism (under-relaxation) is theoretically optimal for the simplest axial solver. Whether these parameters are optimal for more complicated iteration schemes containing axial sweeps is an open question.

Chapter 4

Depletion Algorithm

In the simulation of depletion, MPACT can call the ORIGEN code [7], which is included in the SCALE package [17]. However, MPACT has its own internal depletion model, which is based closely on ORIGEN, with a reduced isotope library and number of isotopes. The following discussion pertains to the internal depletion model in MPACT. We refer the reader to the ORIGEN manual for a more detailed description of the basic equations.

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:

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^{N} \ell_{ij}\lambda_j X_j(t) + \bar{\phi}\sum_{k=1}^{N} f_{ik}\sigma_k X_k(t) - (\lambda_i + \sigma_i\bar{\phi})X_i(t)\ , \quad 1 \le i \le N\ , \tag{4.1}$$

where

$X_i(t) =$ density of nuclide i at time t ,

$\lambda_i =$ radioactive disintegration constant for nuclide i ,

$\sigma_i =$ 1-group neutron absorption cross section of nuclide i ,

$\bar{\phi} =$ 1-group neutron scalar flux ,

$\ell_{ij} =$ fraction of radioactive disintegrations by nuclide j that lead to the formation of nuclide i ,

$f_{ik} =$ the fraction of neutron reactions by nuclide k that lead to the formation of nuclide i .

If the nuclide concentrations are expressed as a vector

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

then Eqs. (4.1) can be written in matrix form as:

$$\frac{d\mathbf{X}(t)}{dt} = \boldsymbol{A} \cdot \mathbf{X}(t) \ , \tag{4.2}$$

where $\boldsymbol{A}$ is an $N \times N$ matrix constructed from characteristic neutron reaction rates and fractions and radioactive decay rates and fractions.

In principle, the matrix exponential method can be used to obtain the solution of Eq. (4.2) as:

$$\mathbf{X}(t) = \exp(\boldsymbol{A}t)\mathbf{X}(0) \ , \tag{4.3}$$

where the vector $\mathbf{X}(0)$ represents the known particle number densities at the beginning of the time step. Obtaining $\mathbf{X}(t)$ then becomes a matter of calculating $\exp(\boldsymbol{A}t)$, which is done through a Taylor series expansion:

$$\exp(\boldsymbol{A}t) = I + \boldsymbol{A}t + \frac{(\boldsymbol{A}t)^2}{2!} + \cdots = \sum_{n=0}^{\infty} \frac{(\boldsymbol{A}t)^n}{n!} \ . \tag{4.4}$$

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. (4.3). 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.

Since the full problem cannot be solved efficiently by the matrix exponential method, the matrix is divided into two parts; one for long-lived nuclides, and the other for short-lived nuclides. Then the matrix exponential in Eq. (4.2) can be accurately computed for just the long-lived nuclides. The criterion for the matrix separation is based on the “removal" half-life of a nuclide. Since any concentration of a nuclide essentially becomes zero after 10 half-lives, the “long-lived" nuclides are defined to be the nuclides for which the irradiation time interval is less than 10 times their removal half-life, i.e.

$$\Delta t \leq 10 t_{r,1/2} = 10 \left(\frac{\ln 2}{\lambda_i + \sigma_i \bar{\phi}} \right) \ . \tag{4.5}$$

To insure that the error of truncating the series expansion to a finite number of terms does not yield incorrect results, the number of terms used is determined dynamically based on the matrix norm defined by:

$$||\boldsymbol{A}|| = \min \left[\max_{1 \leq j \leq N} \left(\sum_{i=1}^{N} |a_{ij}| \right) , \max_{1 \leq i \leq N} \left(\sum_{j=1}^{N} |a_{ij}| \right) \right] \ . \tag{4.6}$$

The nuclide number densities for short-lived nuclide chains beginning with a long-lived precursor are calculated using the following iterative method. The short-lived daughter is assumed to be in secular equilibrium with its parent at the end of any time interval. The concentration of the parent is obtained from the exponential matrix method, and the concentration of the daughter is calculated by assuming its time rate of change is zero:

$$\frac{dX_i}{dt} \approx 0 = \sum_{j=1}^{N} a_{ij} X_j \ . \tag{4.7}$$

Eq. (4.7) can be solved easily using a Gauss-Seidel iteration. The coefficients in Eq. (4.7) 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. (4.7) and using assumed or previously-calculated values for the unknown concentrations to estimate the concentration for the next iteration:

$$X_i^{k+1} = -\frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} X_j^k \ . \tag{4.8}$$

This iterative procedure has been found to converge rapidly since, for these short-lived isotopes, cyclic chains are not usually encountered, and the procedure reduces to a direct solution.

For the full depletion chain, only the non-zero elements of the transition matrix are stored in two MPACT arrays for the diagonal and off diagonal elements. These coefficients are based on those found in Eq. (4.1) when it is written explicitly for each nuclide X_i. However, since only long-lived nuclides are considered in the matrix exponential method, a reduced transition matrix needs to be formulated.

A generalized treatment of the full transition matrix to produce the reduced transition matrix is achieved by searching through the decay chain and forming a queue of all short-lived precursors for an isotope. The queue is terminated when the farthest removed precursor is no longer short-lived. The solution of the Bateman equation is 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 may be written in the form:

$$X_i(t) = X_i(0) e^{-d_i t} + \sum_{k=1}^{i-1} X_k(0) \left[\sum_{j=k}^{i-1} \left(\frac{e^{-d_j t} - e^{-d_i t}}{d_i - d_j} a_{j+1,j} \prod_{n=k, n \neq j}^{i-1} \frac{a_{n+1,n}}{d_n - d_j} \right) \right] \ . \tag{4.9}$$

The notation $a_{i,j}$ for the first-order rate constant is the same as described before, and $d_i = -a_{i,i}$. In the present application, Eq. (4.9) is recast in the form

$$X_i(t) = X_i(0) e^{-d_i t} + \sum_{k=1}^{i-1} X_k(0) \prod_{n=k}^{i-1} \frac{a_{n+1,n}}{d_n} \left[\sum_{j=k}^{i-1} \left(d_j \frac{e^{-d_j t} - e^{-d_i t}}{d_i - d_j} \prod_{n=k, n \neq j}^{i-1} \frac{d_n}{d_n - d_j} \right) \right] \ , \tag{4.10}$$

by multiplication and division of

$$\prod_{n=k}^{i-1} d_n \, .$$

The first product in Eq. (4.10) is the fraction of atoms of isotope k that follow a particular sequence of decays and captures. If this product becomes less than 10^{-6}, contributions from nuclide k and its precursors to the concentration of nuclide i are neglected. This procedure is unnecessary for evaluating the outer summation because all the terms in this sum are known to be positive. In order to avoid a division by zero when two removal constants are approximately equal ($d_i \approx d_j$), the bracketed term in Eq. (4.10) is replaced by:

$$\sum_{j=k}^{i-1} \left(d_j \frac{e^{-d_j t} - e^{-d_i t}}{d_i - d_j} \prod_{n=k,n \neq j}^{i-1} \frac{d_n}{d_n - d_j} \right) \approx \sum_{j=k}^{i-1} \left(d_j t e^{-d_j t} \prod_{n=k,n \neq j}^{i-1} \frac{d_n}{d_n - d_j} \right) \, . \tag{4.11}$$

An analogous expression is derived for the case when $d_n \approx d_j$. 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 then can be thought of as the coefficient of $X_k(0)$, where the product over n is over the queue of short-lived nuclides. The exponential matrix method and the Bateman solutions complement each other in this approach. The exponential matrix method is quite accurate when the transition coefficients are small, but it has problems when including large rate constants; conversely, the Bateman solution has some numerical difficulties for extremely small rate constants, but is stable and accurate for large rate constants. The point depletion algorithm is the same as in ORIGEN and is summarized in Figure 4.1.

In MPACT, the flat flux regions of the fuel pin are azimuthally and radially dependent as shown on the left hand side of Figure 4.2. Currently, the depletable region of the pin is only radially dependent, as shown on the right hand side of Figure 4.2. This has been shown to be adequate unless there are strong local asymmetries. The capability to treat azimuthally-varying depletable regions is a simple modification, but will considerably increase memory requirements.

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 done by dividing the depletion problem into series of time steps, and periodically performing transport calculations. However, because the time dependence of the flux has non-linear 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, leading to a longer computation time. Therefore, to reduce computation time and allow for longer burnup step sizes, MPACT adopts two commonly used tech-

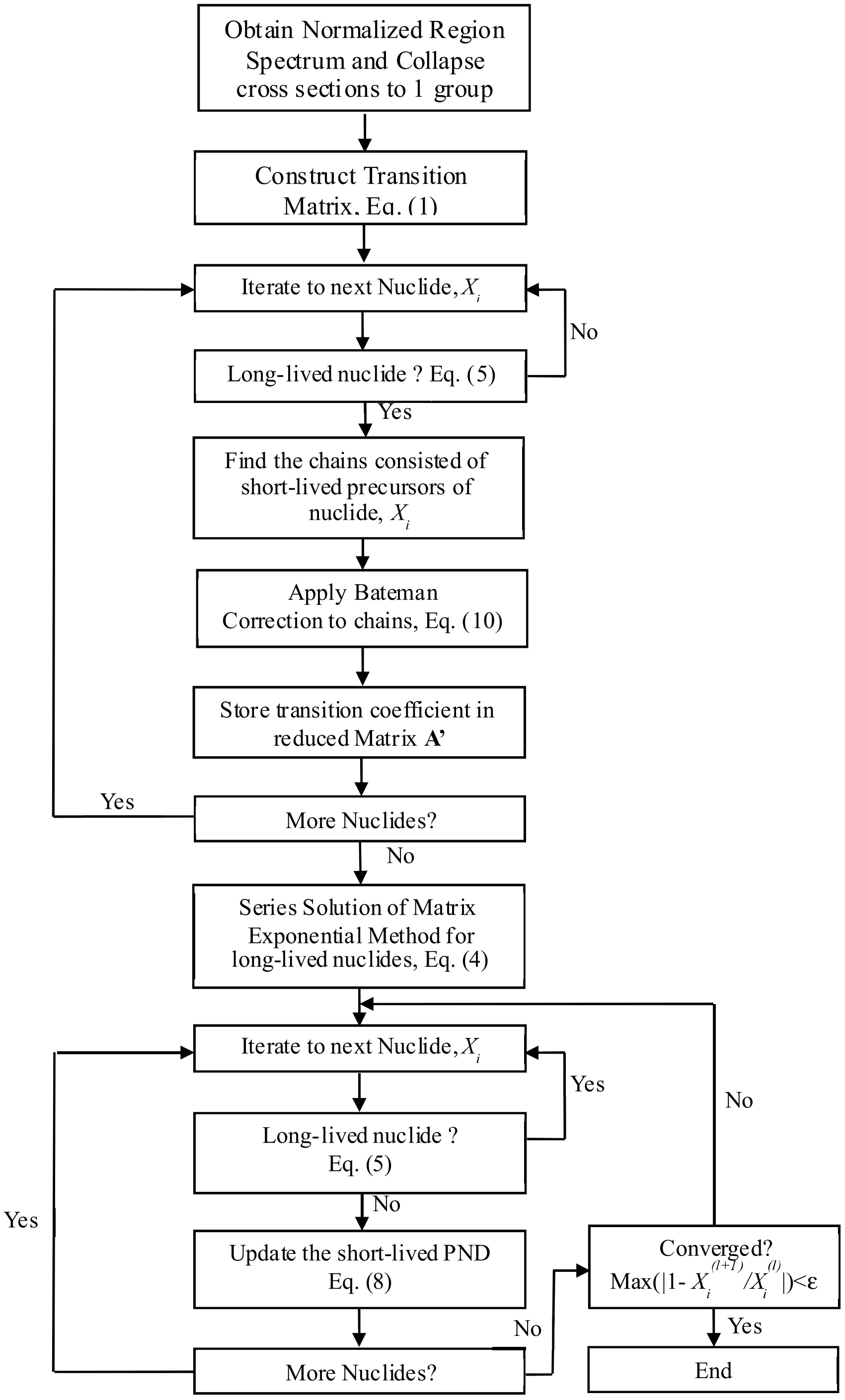
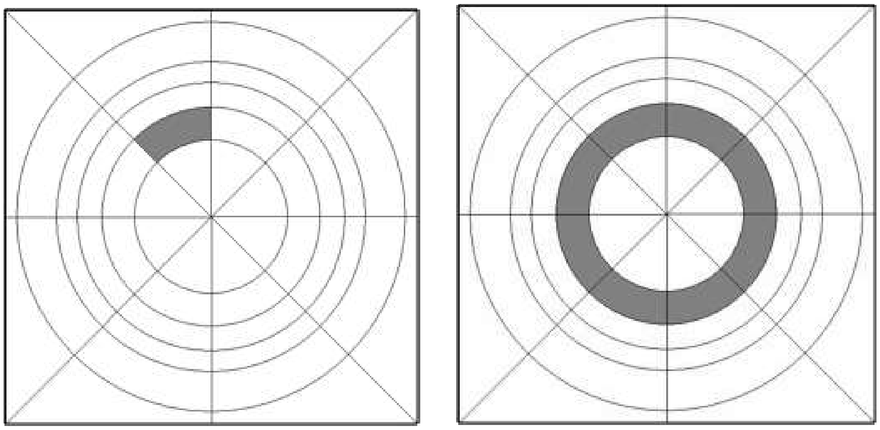


Figure 4.1: Point Depletion Algorithm



Uniform Cross Section Region and Flat Flux Region in the Fuel Pin (Grey)

Radially-Dependent Depletable Region in the Fuel Pin (Grey)

Figure 4.2: Depletion Zones in MPACT Pin Cell

niques: the predictor-corrector and sub-step methods. The predictor-corrector method works by computing a predicted nuclide concentration for a given time step, and then a corrected nuclide concentration. The basic predictor-corrector approach is:

$$X_{t2} = \frac{X^P_{t2}(\phi_{t1}, \sigma_{t1}) + X^C_{t2}(\phi^P_{t2}, \sigma^P_{t2})}{2} \ . \tag{4.12}$$

The predictor step includes the typical depletion calculation to obtain the particle number densities, $N_{t2}(\phi_{t1}, \sigma_{t1})$, at burnup t_2 by using the 1-group flux and cross section at the time of burnup t_1. At this point the new predicted 1-group flux (ϕ^p_{t2}) and cross section (σ^p_{t2}) are obtained through a transport calculation using the predicted concentration, $N^P_{t2}(\phi_{t1}, \sigma_{t1})$. 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, $N^C_{t2}(\phi^P_{t2}, \sigma^P_{t2})$, are obtained. The final particle number densities for t_2 are then taken to be the arithmetic mean of the predicted and corrected concentrations. Once N_{t2} is obtained, then a transport calculation is performed to obtain the steady state flux distribution at t_2, (ϕ_{t2}).

The sub-step method is applied to perform multiple depletion calculations between transport calculations. Since the depletion calculation typically takes less time than the transport calculation, this will often save computational time. What this amounts to mathematically, is that the normalization factor, f, of the flux becomes time-dependent, but the eigenvector representing the flux distribution is still assumed to be constant between transport calculations. For M sub-steps the m^{th} flux, representing the flux at time

$t_1 + m\Delta t/M$, used by the depletion calculations is:

$$\phi_m = \phi_{t1} f_{m-1} \ , \tag{4.13}$$

where P_{t1} is the total power at t_1, ϕ_{t1}^{j} and $\kappa\sigma_{t1}^{f,i,j}$ are the eigenvector for region j and the energy per fission multiplied by the microscopic fission cross section of region j and nuclide i at t_1, respectively, and $N_{m-1}^{i,j}$ is the nuclide concentration of sub-step $m-1$. The sub-step method allows for even coarser burnup steps without a loss in accuracy. An overview of the depletion algorithm in MPACT is depicted in Figure 4.3.

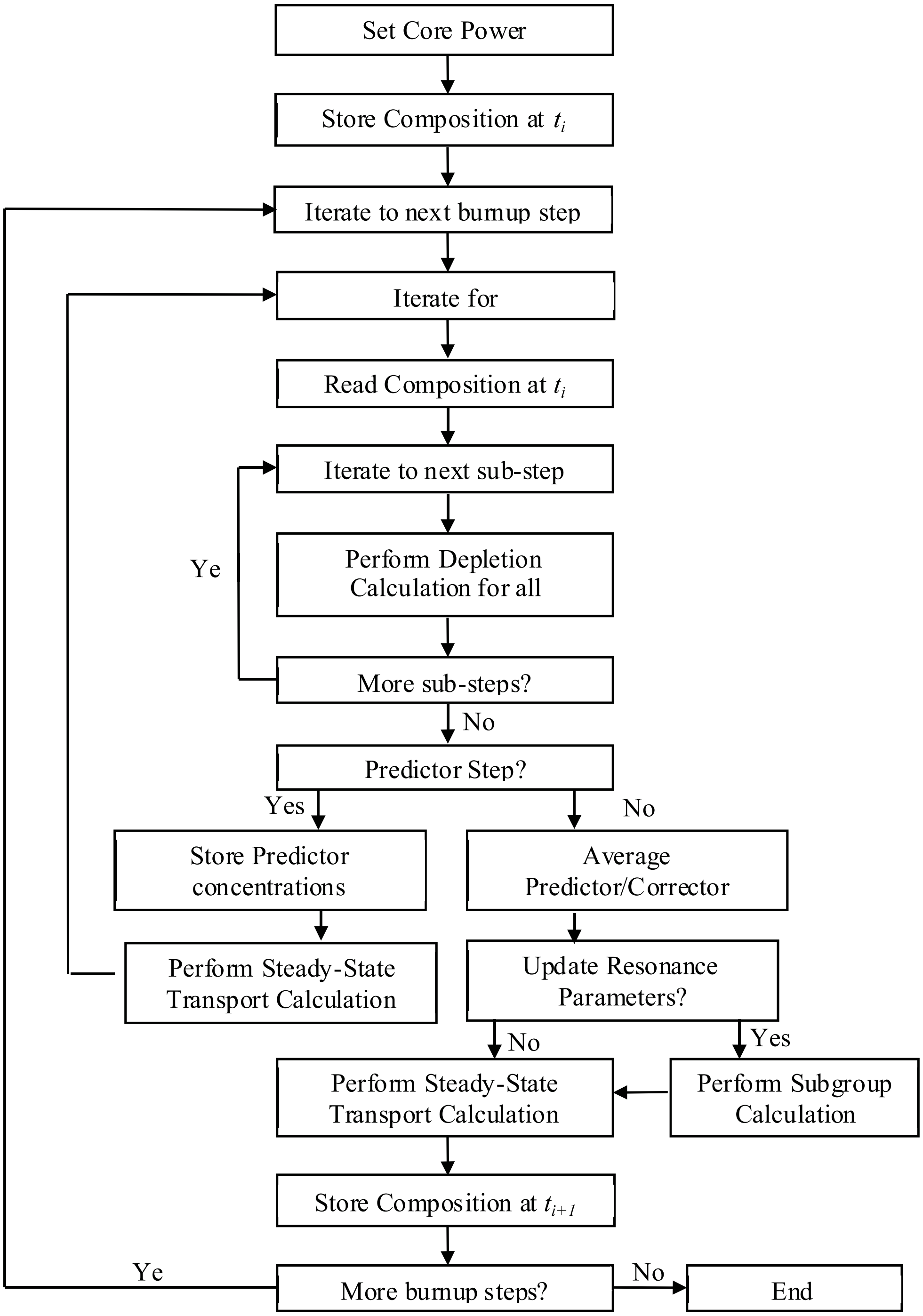


Figure 4.3: MPACT Depletion Algorithm

Chapter 5

Resonance Treatment

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

$$\sigma_{x,g}\left(\mathbf{r}\right)=\frac{\int_{\Delta u_g}\sigma_x\left(\mathbf{r},u\right)\phi\left(\mathbf{r},u\right)du}{\int_{\Delta u_g}\phi\left(\mathbf{r},u\right)du}\ . \tag{5.1}$$

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

In general, there are two ways to perform the resonance self-shielding calculation. The way guaranteeing the accuracy in the energy domain is solving the exact slowing-down equations for the problem of interest. Almost no approximations are made on the energy-dependent cross sections, except neglecting the direct fission and up-scattering contribution on the resolved resonance range. The ultra-fine group or continuous-energy (CE) cross sections are needed to resolve the resonance behavior of neutron interacting with isotopes. Because of the limited computational resources, exact slowing-down codes such as CENTRM [20] and RMET21 [14] are usually designed for 1-D cylindrical geometry calculations.

The basic configuration to be used in the actual core calculation is a square pin cell, which is converted into a circular cell through the Wigner-Seitz approximation. Due to the computational expense of direct calculation for 2-D problems, the exact method limited in 1-D cylindrical cells usually lacks of consideration of the inter-pin spatial self-shielding effects in the reactor geometry. Another approach utilizes pre-computed Resonance Integral (RI) tables, which are established by the exact slowing-down solution of a variety of background cross sections. Based on the equivalence theory [18], different methods can be derived in order to determine the equivalent cross sections for consideration of spatial self-shielding effects. The Bondarenko background cross section method [3] is the conventional method incorporating Dancoff factors, accounting for the spatial self-shielding. The Subgroup method [5] is another RI table based method, where the RI tables are converted to a set of subgroup levels and weights so that the equivalence cross sections are subgroup-level dependent. Recently, another promising RI table based method, the Embedded Self-Shielding Method (ESSM) was proposed [21] by Oak Ridge National Lab. Compared to the conventional Bondarenko method, in which the Dancoff factors are approximated or evaluated 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.

The MPACT code has the capability of performing the subgroup method and ESSM for the resonance self-shielding calculations, based on the multigroup libraries with subgroup parameters, such as the HELIOS library [6]. As the code is elaborately designed for easy extension of new libraries by implementing a small set of interfacing functions associated with the library, it is very straightforward to add new multigroup libraries to perform the resonance and transport calculations. Specifically, we incorporated the 56-G Multigroup library generated by Oak Ridge National Lab into MPACT for the initial verifications. The library is processed from ENDF-BVII.0 using sequences of SCALE-6.1 system [4], as well as optimization code for generating subgroup parameters.

The presentation of the resonance self-shielding treatment begins with the derivation of the neutron spectra to be used in the resonance integral calculations, which are the fundamentals formulating the RI table based methods. The Subgroup method and ESSM are respectively discussed in the second subsection. Other features of resonance treatment in MPACT are discussed in the last subsection.

5.1 The Resonance Self-Shielding Treatment

The rigorous approach to obtain the spectrum in Eq. (5.1) is solving the slowing down equation relative to a problem of interest

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\sigma_{t,i}\left(\mathbf{r},u\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)\\ =\frac{1}{4\pi}\sum_i\int_{u+\epsilon_i}^{u}\sigma_{s,i}\left(\mathbf{r},u'\right)\phi\left(\mathbf{r},u'\right)\frac{\exp^{u'-u}}{1-\alpha_i}du'\ ,\end{aligned} \tag{5.2}$$

where i is summed over all isotopes of the material, and ϵ_i is the maximum lethargy gain of isotope i per scattering. Three major 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) The up-scattering is neglected; (3) The direct fission source is neglected. In order to decouple the lethargy dependence in the scattering source from lethargy $u-\epsilon_i$ to u, the Intermediate Resonance (IR) approximation [8] is employed to achieve

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\sigma_{t,i}\left(\mathbf{r},u\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)\\ =\frac{1}{4\pi}\left(\sum_i\lambda_i\sigma_{p,g}\left(\mathbf{r}\right)+\sum_i\left(1-\lambda_i\right)\sigma_{s,i}\left(\mathbf{r},u\right)\phi\left(\mathbf{r},u\right)\right)\ .\end{aligned} \tag{5.3}$$

By neglecting $\lambda_i\sigma_{RS,i}\left(u\right)$ and assuming the isotropic fluxes in the second term of the right hand side, a much simpler equation is formed in which the source term has no flux dependence:

$$\boldsymbol{\Omega}\cdot\nabla\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)+\sum_i\left(\sigma_{a,i}\left(\mathbf{r},u\right)+\lambda_i\sigma_{p,i}\left(\mathbf{r}\right)\right)\varphi\left(\mathbf{r},u,\boldsymbol{\Omega}\right)=\frac{1}{4\pi}\sum_i\lambda_i\sigma_{p,g}\left(\mathbf{r}\right)\ . \tag{5.4}$$

For a homogenous material, the flux of Eq. (5.4) can be written as

$$\phi_{\text{hom}}\left(u\right)=\frac{\sum\limits_i\lambda_i\sigma_{p,i}}{\sum\limits_i\sigma_{a,i}\left(u\right)+\sum\limits_i\lambda_i\sigma_{p,i}}\ . \tag{5.5}$$

The equivalence theory correlates the flux of a homogeneous material with a heterogeneous material by introducing the equivalence cross section

$$\phi_{\text{het}}\left(u\right)=\frac{\sum\limits_i\lambda_i\sigma_{p,i}+\Sigma_e}{\sum\limits_i\sigma_{a,i}\left(u\right)+\sum\limits_i\lambda_i\sigma_{p,i}+\Sigma_e}=\frac{\Sigma_b}{\sum\limits_i\sigma_{a,i}\left(u\right)+\Sigma_b}\ . \tag{5.6}$$

By introducing Eq. (5.6) into Eq. (5.1), the effective cross section is only through the background cross section Σ_b, so a table of effective cross sections (or RI) can be built through various background levels. 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 by the fluxes from the slowing-down solution.

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

5.2 The Subgroup Method and ESSM

The subgroup method transforms the integration variable from energy to absorption cross section. Eq. (5.6) gives important information that the flux depression is mainly due to the absorption cross sections. Although the absorption cross sections are a strong function of energy (or lethargy), it is more efficient to perform the integration of Eq. (5.1) through absorption cross section rather than neutron energy [6].

$$\sigma_{x,g} = \frac{\int_{\Delta u_g} \sigma_x(u)\,\phi(u)\,du}{\int_{\Delta u_g} \phi(u)\,du} = \frac{\int_{\Delta u_g} f(u)\,du}{\int_{\Delta u_g} \phi(u)\,du} = \frac{\int_{\Delta u_g} f(\sigma)\,\frac{du}{d\sigma}d\sigma}{\int_{\Delta u_g} \phi(\sigma)\,\frac{du}{d\sigma}d\sigma} \quad \left(f(u) = \sigma_x(u)\,\phi(u)\right)\ . \tag{5.7}$$

The integrals of Eq. (5.7) can be cast into a quadrature form represented by the subgroup levels and weights

$$\sigma_{x,g} \approxeq \frac{\sum\limits_n \sigma_{x,g,n}\phi_{g,n}w_{x,g,n}}{\sum\limits_n \phi_{g,n}w_{x,g,n}}\ . \tag{5.8}$$

The subgroup levels and weights are determined by searching the desired fit for a set of pre-computed RI tables parameterized by background cross sections. In order to obtain the same set of subgroup levels and weights for the numerator and denominator of Eq. (5.8), the summation of the weights are forced to unity by including zero-level parameters $w_{x,g,0}$ and $\sigma_{x,g,0}$ [6].

Subgroup calculations are performed for each absorption levels by solving the FSP as Eq. (5.4), and the resulting fluxes can be used in Eq. (5.8) to estimate the effective cross sections. However, it is preferable to represent the level dependent flux in Eq. (5.8) by introducing Eq. (5.5),

$$\sigma_{x,g} \approxeq \frac{\sum\limits_{n} \sigma_{x,g,n} \frac{\sigma_{b,g,n}}{\sigma_{a,g,n}+\sigma_{b,g,n}} w_{x,g,n}}{\sum\limits_{n} \frac{\sigma_{b,g,n}}{\sigma_{a,g,n}+\sigma_{b,g,n}} w_{x,g,n}} \quad . \tag{5.9}$$

An assumption is made in this equation that there is no resonance interference among isotopes within the broad energy group g, so the flux depression is represented by the concerning resonant isotope itself. Practically, Eq. (5.9) is used for evaluation of the effective cross sections instead of directly using $\phi_{g,n}$ of the fixed source transport solution. This detour option is chosen because the dependence of $\sigma_{b,g,n}$ on $\sigma_{a,g,n}$ is much weaker than the dependence of $\phi_{g,n}$ on $\sigma_{a,g,n}$. The number of $\sigma_{b,g,n}$ capable of describing this dependence is therefore smaller than the number of subgroup levels used in the quadrature representing the effective cross section. Thus, fewer fixed source calculations are required for the resonance calculations. Similar with the subgroup treatment of HELIOS code, MPACT uses four absorption levels for the FSP calculation, and then the resulting tables of $\Sigma_{e,n}\left(\sigma_{a,n}\right)$ are interpolated to seven levels for the evaluation of effective cross sections. The category concept is applied to perform less fixed source transport calculations.

ESSM also solves the FSP to determine the equivalence cross sections, but the quadrature representation is replaced by iterations between fixed source solver and self-shielded parameters. Figure 5.1 depicts the ESSM procedures. The effective absorption is a monotonous function of background cross section, as shown by the solid blue line. Considering a problem with a real effective cross section $\sigma_{a,\text{real}}$ the ESSM is looking for, the method has to start with an initial guess of $\sigma_{b,0}$ associated with an effective $\sigma_{a,0}$ interpolated from the RI tables. Relative to $\sigma_{a,\text{real}}$, the smaller $\sigma_{a,0}$ introduced into Eq. (5.4) for solving the FSP should result in a lager background cross section $\sigma_{b,1}$ relative to $\sigma_{b,\text{real}}$. 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 bears quadrature errors. ESSM is a thoroughly embedded method covering everything within the transport calculations. 2 The computational time of subgroup method is pre-determined by the number of subgroup levels used in the fixed source calculations, while the computational time of ESSM is cross section-dependent, relative to the average number of iteration.

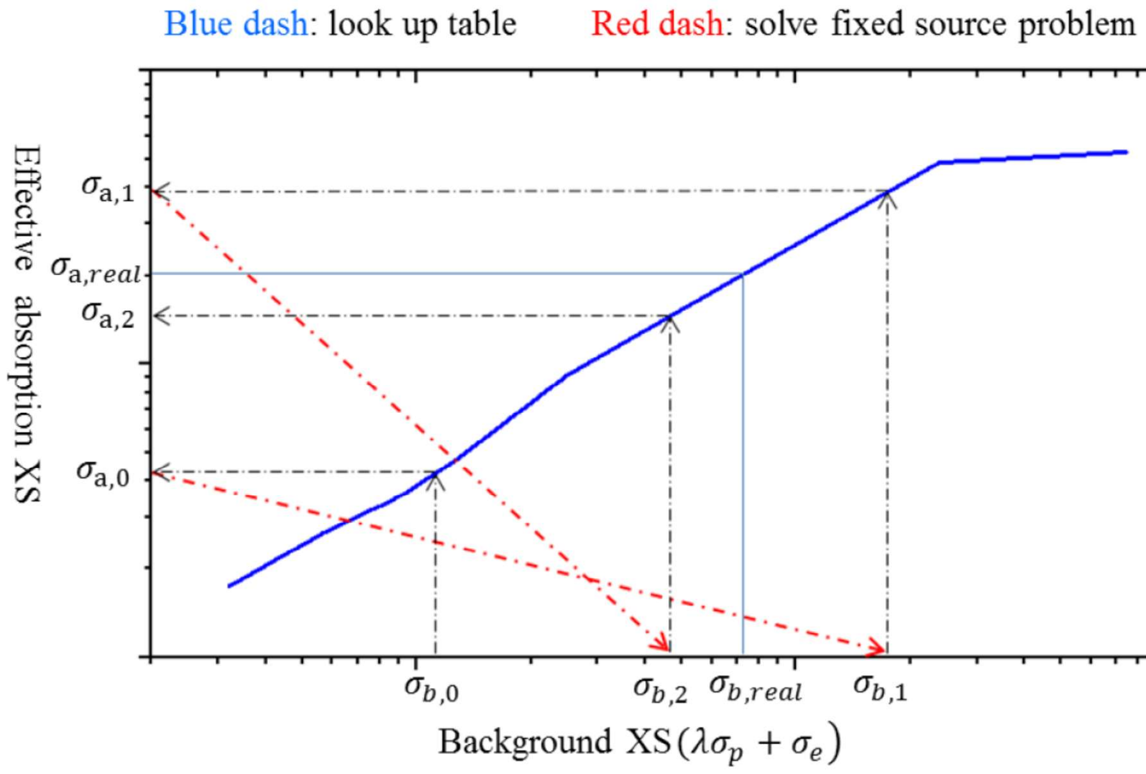


Figure 5.1: Conceptual illustration of ESSM procedures

5.3 Other Features of the Resonance Treatment in MPACT

All the RI table based methods such as subgroup 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 usually being considered afterward on the Multi-group level, e.g., by Bondarenko iteration described in WIMS code [2]. Ref. [15] describes our research on improving the treatment of resonance interference by incorporating the 0-D slowing-down solution. This feature will be added into MPACT once MPACT has a verified slowing-down solver as well as a point-wise cross section library, but for now, the conventional Bondarenko iteration method is used to roughly estimate the interference effect.

Taking the advantage of the availability of the resonance scattering RI tables in the Multigroup library, resonance scattering effects are considered explicitly by post correction in MPACT. 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 firstly converted back to the effective absorption of the representative isotope of the category, such that the equivalence cross section can be interpolated using the table of $\Sigma_{e,n}\left(\sigma_{a,n}\right)$. Then the background cross section associated with shielded absorption can be simply determined. For ESSM, 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 P0 scattering cross section of each resonance group can be calculated. Compared to the unshielded scattering cross section provided by the library, a ratio is defined as

$$f_{\text{iso},g} = \frac{\sigma^{\text{shielded}}_{s,\text{iso},g}}{\sigma^{\text{unshielded}}_{s,\text{iso},g}} \,. \tag{5.10}$$

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

$$\sigma^{\text{shielded}}_{s_0,\text{iso},g\to g'} = \begin{cases} f_{\text{iso},g}\,\sigma^{\text{unshielded}}_{s_0,\text{iso},g\to g'} & g' \neq g \\ f_{\text{iso},g}\left(\sigma^{\text{unshielded}}_{s_0,\text{iso},g\to g'} - \sum\limits_{g''}\sigma^{\text{unshielded}}_{s_1,\text{iso},g\to g''}\right) & g' = g \end{cases} \,. \tag{5.11}$$

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

Chapter 6

Simplified Thermal Hydraulic Model

6.1 Introduction

The simplified T/H solver in MPACT provides a basic mechanism to apply T/H 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 fuel temperature is solved on every pin in the model. This process is outlined next.

6.2 Fluid Flow Model

The flow model in MPACT provides the basic mass-energy balance of the fluid flow through the reactor. The first approximation in the flow model is that every assembly is a closed flow channel. The flow through 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

$$\dot{m}_{chan} = \left(\frac{A_{chan}}{\sum_{c \in channels} A_c} \right) \dot{m}_{core} \ . \tag{6.1}$$

Once the mass flow rate at the inlet of each channel is determined, the solver sequentially marches up the channel and determines the enthalpy at the outlet of each node in the neutronic solution

$$h_{out} = h_{in} + \frac{P}{\dot{m}_{chan}} \ , \tag{6.2}$$

where h_{in} and h_{out} is the flow enthalpy at the inlet and outlet of a node respectively and P is the nodal power. 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 inside the node to determine the average enthalpy inside the node of interest:

$$\overline{h} = \frac{h_{out} + h_{in}}{2} \; . \tag{6.3}$$

Once the average enthalpy is determined, equations of state are used to determine the coolant temperature and density:

$$\overline{T_{cool}} = T\left(\overline{h}, P_{sys}\right) \; , \tag{6.4a}$$

$$\overline{\rho_{cool}} = \rho\left(\overline{h}, P_{sys}\right) \; , \tag{6.4b}$$

where P_{sys} is the system pressure.

The equations of state for the simplified T/H model are all 1D cubic polynomials as a function of enthalpy or temperature. The pressure is at $2250psia$, the nominal operating pressure for most PWRs. The temperature ranges from $550K$ to $650K$, which also encompasses the operating range of a standard PWR.

Once the coolant properties are determined for each channel, the fuel temperature model (described next) is applied to each pin in the core.

6.3 Fuel Temperature Model

The fuel temperature model solves the 1D heat conduction equation in every pin in the model. The first step is to determine the cladding surface temperature. Once the clad surface temperature is determined, heat is conducted through the cladding, across the gap through the use of a gap conductivity, and finally through the fuel itself.

6.3.1 Transfer from Fluid to Clad

Obtaining the clad surface temperature is achieved through the use of the Dittus-Boelter heat transfer coefficient, which is modified to account for a regular square array of pins [19]:

$$T_{clad,surf} = T_{mod} + \frac{P}{A_{surf} h_{mDB}} \ , \tag{6.5a}$$

$$h_{mDB} = C_0 \frac{k}{D_h} \mathrm{Re}^{0.4} \mathrm{Pr}^{0.2} \ , \tag{6.5b}$$

$$C_0 = 0.042 \frac{p_{pin}}{2 r_{clad}} - 0.024 \ , \tag{6.5c}$$

where P 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. Several properties of the fluid are needed for this calculation, and they are calculated using the same cubic polynomial as the temperature and density discussed in the previous section.

6.3.2 Gap Conductance

Once the outer clad surface temperature is obtained, the conduction through each region can be calculated. This is the topic of the next section. One remaining region that needs special treatment is the fuel-clad gap, a small region filled with gas, which provides a relatively large thermal resistance. The dynamics of the fuel-clad gap are difficult to model, so instead of solving the conduction through this region, the solver treats this as a conductance in a similar way to the transfer of heat from the fluid to the clad surface temperature:

$$T_{fuel,surf} = T_{clad,in} + \frac{P}{A_{surf} h_{gap}} \ , \tag{6.6}$$

where $T_{fuel,surf}$ is the fuel surface temperature, $T_{clad,in}$ is the inner clad temperature, P is the local power, A_{surf} is the surface area of heat transfer, and h_{gap} is the gap conductance.

6.3.3 Radial Heat Transfer Equation

The flow of heat through a medium is well understood and is modelled using the heat conduction equation;

$$-\boldsymbol{\nabla} \cdot k\left(T\right) \boldsymbol{\nabla} T\left(\boldsymbol{x}\right) = \dot{q}\left(\boldsymbol{x}\right) \ , \tag{6.7}$$

where k is the thermal conductivity, T is the temperature, and $\dot{q}$ is the heat generation rate. This is simplified into a 1D cylindrical equation equation for the solution here;

$$-\frac{1}{r}\frac{d}{dr}rk\left(T\right)\frac{d}{dr}T\left(r\right)=\dot{q}\left(r\right)\ . \tag{6.8}$$

In order to discretize the 1D heat conduction equation, finite volumes are defined between 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:

$$T\left(r\right)=-\frac{\dot{q}}{4k}r^2+C_0\ln\left(r\right)+C_1\ , \tag{6.9a}$$

$$T\left(r_{i+1}\right)=T_{out}\ , \tag{6.9b}$$

$$q_i'\left(r_i\right)=-k_i\frac{dT}{dr}\ . \tag{6.9c}$$

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

$$T\left(r\right)=\frac{\dot{q}_i}{4k_i}\left(r_{i+1}^2-r^2\right)+\left(\frac{\dot{q}_i r_i^2}{2k_i}-\frac{q_i' r_i}{k_i}\right)\ln\left(\frac{r}{r_{i+1}}\right)+T\left(r_{i+1}\right)\ , \qquad r_i<r<r_{i+1}\ . \tag{6.10}$$

Using Eq. (6.10), the temperature on the inside of the volume, r_i, can be determined along with the volume average temperature of the volume:

$$\overline{T_i}=\frac{1}{\pi\left(r_{i+1}^2-r_i^2\right)}\int_{r_i}^{r_{i+1}}2\pi rT\left(r\right)dr\ . \tag{6.11}$$

With the equation relating the temperature at r_i and the average temperature $\overline{T_i}$ to the temperature at r_{i+1}, an iterative scheme can be devised starting at the outside of the clad and moving inward. The equation for the cladding is simplified because there is no heat generation;

$$T\left(r\right)=\frac{q_i' r_i}{k_i}\ln\left(\frac{r_{i+1}}{r}\right)+T\left(r_i\right)\quad r_i<r<r_{i+1}\ . \tag{6.12}$$

Since the thermal conductivity is a function of temperature, each volume is iterated until the average temperature stops changing. Once the temperature in the clad is obtained, the gap conductance model is used to obtain the fuel surface temperature. Then the same procedure is used to solve for the average temperature in each ring of the fuel. Since the fuel generates heat, the linear heat rate at r_i changes at each ring. The iteration procedure continues into the the fuel until the innermost region is reached. The major difference about the center core of fuel is that $r_i=0$ and $q'\left(0\right)=0$ because of symmetry of the 1D equations.

6.3.4 Thermal Properties

The thermal conductivity of the clad and fuel are both modeled using low order polynomials. The thermal conductivity of the clad is modeled using:

$$k_{clad}\left(T\right) = 7.51 + 2.09 \cdot 10^{-2}T - 1.45 \cdot 10^{-5}T^2 + 7.67 \cdot 10^{-9}T^3 \quad \frac{W}{m-K}\ . \tag{6.13}$$

The thermal conductivity of the fuel does not take into account several components; burnup, porosity, initial Plutonium loading, or Gadolinium loading, but instead is only a function of temperature:

$$k_{fuel}\left(T\right) = 1.05 + \frac{2150}{T - 73.15} \quad \frac{W}{m-K}\ . \tag{6.14}$$

6.4 Discussion

The simplified T/H solver described above is to be used as a preliminary scoping capability, to provide leading order effects of T/H feedback. However, the capabilities of this solver are limited. It is recommended that a more advanced T/H capability, such as CTF [16], be used for production calculations.

Bibliography

[1] J.R. Askew. A characteristics formulation of the neutron transport equation in complicated geometries. *UKAEA Report*, AEEW-M-1108, 1971. 4

[2] J.R. Askew, F.J. Fayers, and P. B. Kemshell. A general description of the lattice code wims. *J. British Nucl. Energy Soc.*, 5:564–585, 1966. 52

[3] I.I. Bondarenko et al. *Group Constants for Nuclear Reactor Calculations*. Consultants Bureau, New York, 1964. 47

[4] S.M. Bowman. SCALE 6: Comprehensive nuclear safety analysis code system. *Nucl. Technol.*, 174:126–148, 2011. 47

[5] D.E. Cullen. Application of the probability table method to multigroup calculations of neutron transport. *Nucl. Sci. Eng.*, 55:387, 1974. 47

[6] R.J.J. Stammler et. al. HELIOS methods. Technical report, Studsvik Scanpower, 2003. 47, 49

[7] 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. 38

[8] R. Goldstein and E.R. Cohen. Theory of resonance absorption of neutrons. *Nucl. Sci. Eng.*, 13:132–140, 1962. 48

[9] M.J. Halsall. CACTUS, a characteristics solution to the neutron transport equations in complicated geometries. *UKAEA Report*, AEEW-R-1291, 1980. 4

[10] 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. 31

[11] 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. 31, 32

[12] B.W. Kelly and E.W. Larsen. A consistent 2d/1d approximation to the 3d neutron transport equation. *Nucl. Eng. Design*, submitted. 31, 32, 34

[13] 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. 34

[14] F. Leszczynski. Neutron resonance treatment with details in space and energy for pin cells and rod clusters. *Ann. Nucl. Energy*, 14:589–601, 1987. 46

[15] 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. 52

[16] R.K. Salko and M.N. Avramova. (ctf) theory manual. Technical report, The Pennsylvania State University. 57

[17] 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. 38

[18] R.J.J. Stammler and M.J. Abbate. *Methods of Steady-state Reactor Physics in Nuclear Design*. Academic Press, London, 1983. 47

[19] L.S. Tong and J. Weisman. *Thermal Analysis of Pressurized Water Reactors (2nd edition)*. American Nuclear Society, Le Grange Park, Illinois, 1979. 55

[20] M.L. Williams and M. Asgari. Computation of continuous-energy neutron spectra with discrete ordinates transport theory. *Nucl. Sci. Eng.*, 121:173–201, 1974. 46

[21] M.L. Williams and K.S. Kim. The embedded self-shielding method. In *Proc. PHYSOR 2012, Knoxville, TN*, April 15-20, 2012. 47