Atomic Energy of Canada Limited
Toronto, April 20, 2004

DRAGON THEORY FOR 3-D CANDU PROBLEMS

G. MARLEAU

Institut de génie nucléaire
Département de génie physique
École Polytechnique de Montréal
January 2004

Contents

List of Figures

Preface

This report contains a brief description of some theoretical models used in the code DRAGON.[1–8] The exact topics we selected are those that are required for a good understanding of the various options provided in DRAGON for the typical three dimensional transport calculations performed at Atomic Energy of Canada Limited (AECL). Thus, we did not consider in great details the models used in DRAGON to analyze microscopic cross section libraries such as the self-shielding of resonant isotopes since AECL uses the side-step method to extract from their own lattice code the macroscopic cross sections required by DRAGON. For the same reason, the isotopic depletion model used in DRAGON will not be discussed in the current report. On the other hand, a rather general discussion of the transport equation can be found in this report. The main reason for the presence of this discussion is to illustrate the series of successive approximations considered in order to reduce the transport equation to a form that can be solved numerically.

This report is divided into 5 chapters. The first of these chapters mainly deals with the derivation of the collision probability equation that will be solved in the code DRAGON. It also contains a discussion of the general contents of the macroscopic cross section database required in DRAGON. Some information on the generation of this database using DRAGON or other codes is also provided.

In the second chapter we present some of the techniques used in DRAGON to compute the multigroup collision probabilities required to solve the transport equation for the flux. Starting with the definition of the collision probabilities, we first discuss how these can be cast in a form that is more manageable for numerical treatment when three-dimensional geometries are considered. Then, we discuss the quadrature technique that are programmed in DRAGON to perform numerically the four dimensional integral involved in the evaluation of these collision probabilities. This includes a brief discussion of the tracking procedure used in DRAGON for such geometries and a discussion on the use of this tracking information for the explicit integration of the collision probability. We also discuss the normalization schemes that can be used to improve precision of the tracks and collision probabilities generated in DRAGON for a given quadrature type.

The third chapter describes the technique used in DRAGON to solve the linear system that results from the collision probability discretization of the multigroup transport equation for the fluxes and eigenvalue. We also discuss the problem of neutron leakage and analyze the two different homogeneous leakage models that can be used to compute the buckling eigenvalue and the associated homogeneous diffusion coefficient.

Chapter 4 deals with the generation within DRAGON of the information required for finite

reactor codes. A discussion of the condensation and homogenization procedures is provided. Final comments on the use of DRAGON to solve the transport equation are provided in the last chapter.

Chapter 1

Introduction to the Collision Probability Method

The transport equation was formulated more than a century ago by Boltzmann to describe the kinetic behavior of gases. Its application to neutrons is much more recent and dates to the first study of nuclear chain reactors in the 1940s. It is based on the following assumptions:[9–12]

1. Only the mean value of the neutron density distribution is considered;
2. Neutrons are considered as point particles;
3. Neutrons travel in a straight line between collision points;
4. Neutron-neutron interactions are neglected;
5. Collisions between neutrons and nuclides are instantaneous;
6. The strength of the interaction between neutrons and nuclides is known and specified by the cross sections.

Typically the number of collisions per second between a neutron gas of density $n(\vec{r},\vec{v},t)$ traveling at a speed $\vec{v}=v\vec{\Omega}$ and a gas of nuclides of type I at rest with a density $N_I(\vec{r},t)$ in a region of volume V will be given by:

$$R_I(\vec{r},v\vec{\Omega},t)=\int_V vn(\vec{r},v\vec{\Omega},t)N_I(\vec{r},t)\sigma_I(v\vec{\Omega})d^3r\ , \tag{1.1}$$

where $\sigma_I(v\vec{\Omega})$ is the total microscopic cross section for a reaction between the neutron and nuclide I and R is known as the reaction rate. In general we will work with the neutron flux $\Phi(\vec{r},E,\vec{\Omega},t)$ defined as:

$$\Phi(\vec{r},E,\vec{\Omega},t)=vn(\vec{r},v\vec{\Omega},t)\ , \tag{1.2}$$

where $v(E)=\sqrt{2E/m}$ and m is the neutron mass.

In the case where more than one type of nuclide is present in V, the total reaction rate is given by:

$$R(\vec{r},E,\vec{\Omega},t)=\sum_I R_I(\vec{r},E,\vec{\Omega},t)=\int_V \Phi(\vec{r},E,\vec{\Omega},t)\Sigma(\vec{r},E,\vec{\Omega},t)d^3r\ , \tag{1.3}$$

where $\Sigma(\vec{r}, E, \vec{\Omega}, t)$ is the total macroscopic cross section associated with a mixture

$$\Sigma(\vec{r}, E, \vec{\Omega}, t) = \sum_I N_I(\vec{r}, t)\sigma_I(E, \vec{\Omega}) \ . \tag{1.4}$$

The total cross section can be subdivided into various types of reactions including neutron capture ($\Sigma_c(\vec{r}, E, \vec{\Omega}, t)$), neutron scattering ($\Sigma_s(\vec{r}, E, \vec{\Omega}, t)$) and nuclear fission ($\Sigma_f(\vec{r}, E, \vec{\Omega}, t)$).

1.1 The Multigroup Static Transport Equation

The static transport equation is simply a balance equation that states that the neutron production rate $\mathcal{C}$ at a specific position in phase space is equal to the rate $\mathcal{L}$ at which they disappear .

$$\mathcal{L}(\vec{r}, E, \vec{\Omega}) = \mathcal{C}(\vec{r}, E, \vec{\Omega}) \ . \tag{1.5}$$

1.1.1 The Neutron Balance Equation

The contribution to $\mathcal{L}$ is divided into two parts:

- Loss by collision $\mathcal{L}_c$:

$$\mathcal{L}_c = \Sigma(\vec{r}, E)\Phi(\vec{r}, E, \vec{\Omega}) \ , \tag{1.6}$$

  where $\Sigma(\vec{r}, E)$ is the total macroscopic cross section. Some of these collisions lead to the creation of neutrons via scattering and fission which are implicitly included in the total cross section. However, for simplicity, each nuclear collision is considered as an instantaneous two-step process where the neutron is first absorbed by a nuclide to form a new intermediate nuclide that then decays instantaneously by emitting various types of particles including neutrons. The neutrons generated are then considered in the creation process.

- Loss by streaming $\mathcal{L}_s$:

$$\mathcal{L}_s = \vec{\Omega} \cdot \vec{\nabla}\Phi(\vec{r}, E, \vec{\Omega}) \ . \tag{1.7}$$

  This term represents the net number of neutrons leaving a region of space due to the fact that they travel in direction $\vec{\Omega}$ with a kinetic energy E.

In the absence of external sources, the neutron production rate will be divided into two parts, namely:

- Creation by scattering $\mathcal{C}_s$:

$$\mathcal{C}_s = \int dE' d^2\Omega \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}) \Phi(\vec{r}, E', \vec{\Omega}') \; , \tag{1.8}$$

  with $\Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega})$ the cross section for scattering of a neutron with initial energy and direction $(E', \vec{\Omega}\,')$ to energy and direction $(E, \vec{\Omega})$.

- Creation by fission $\mathcal{C}_f$.

$$\mathcal{C}_f = \chi(\vec{r}, E) \int dE' d^2\Omega \nu \Sigma_f(\vec{r}, E') \Phi(\vec{r}, E', \vec{\Omega}') \; , \tag{1.9}$$

  where $\Sigma_f(\vec{r}, E')$ is the fission cross section, ν is the average number of neutrons produced by fission and $\chi(\vec{r}, E)$ is the energy distribution of these neutrons.

We can therefore write the neutron transport equation in the following form:

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \Phi(\vec{r}, E, \vec{\Omega}) + & \Sigma(\vec{r}, E) \Phi(\vec{r}, E, \vec{\Omega}) \\ & = \int dE' \int d^2\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}) \Phi(\vec{r}, E', \vec{\Omega}') \\ & + \frac{1}{k} \chi(\vec{r}, E) \int d^2\Omega' \int dE' \nu \Sigma_f(\vec{r}, E') \Phi(\vec{r}, E', \vec{\Omega}') \; , \end{aligned} \tag{1.10}$$

where the multiplication constant k is the eigenvalue of the problem and gives an indication of how far from equilibrium the system is. In the case of an infinite cell one generally uses k_∞ instead of k. One may also assume that the departure of our system from equilibrium is due to additional leakage. In that case, the neutron flux distribution inside the system can be factorized into a fine structure distribution and a global distribution:

$$\Phi(\vec{r}, E, \vec{\Omega}) \approx \Psi(\vec{r}, E, \vec{\Omega}) \exp(i\vec{B} \cdot \vec{r}) \; , \tag{1.11}$$

Using this factorization, along with some additional approximations (see Section 3.3 for a description of some of these approximations) one can transform Eq. (1.10) to:

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \Psi(\vec{r}, E, \vec{\Omega}) + & \left[\Sigma(\vec{r}, E) + D(\vec{r}, E) B^2 \right] \Psi(\vec{r}, E, \vec{\Omega}) \\ & = \int dE' \int d^2\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}) \Psi(\vec{r}, E', \vec{\Omega}') \\ & + \frac{1}{k_{\mathit{eff}}} \chi(\vec{r}, E) \int d^2\Omega' \int dE' \nu \Sigma_f(\vec{r}, E') \Psi(\vec{r}, E', \vec{\Omega}') \; , \end{aligned} \tag{1.12}$$

where $D(\vec{r}, E)$ is the leakage (or diffusion) coefficient, B^2 is known as the buckling, and k_{eff}, the effective multiplication constant, is generally set to 1.

In order to ensure the closure of the above equation, boundary conditions are required at the external surface S surrounding the system. Here, we will consider albedo boundary conditions of the form

$$\Phi(\vec{r}_S, E, \vec{\Omega} - 2(\vec{N}_S \cdot \vec{\Omega})) = \beta(\vec{r}_S, E)\Phi(\vec{r}_S, E, \vec{\Omega}) \ , \tag{1.13}$$

where the albedo $\beta(\vec{r}_S, E)$ represents the reflection coefficient at the surface S and $\vec{\Omega} - 2(\vec{N}_S \cdot \vec{\Omega})$ is the final direction in which the neutron travels after a reflection (see Figure 1.1). Note that

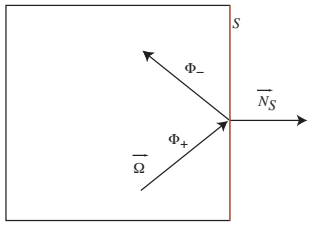

Figure 1.1: Neutron reflection at a surface

vacuum boundary conditions correspond to the special case where the albedo vanishes. One can also use periodic boundary conditions of the form:

$$\Phi(\vec{r}\,'_S, E, -\vec{\Omega}) = T(\vec{r}\,'_S, \vec{r}_S, E)\Phi(\vec{r}_S, E, \vec{\Omega}) \ , \tag{1.14}$$

where $T(\vec{r}\,'_S, \vec{r}_S, E)$ is the transmission coefficient for a neutron exiting the surface S at $\vec{r}_S$ and re-entering at a periodic point $\vec{r}\,'_S$ (see Figure 1.2).

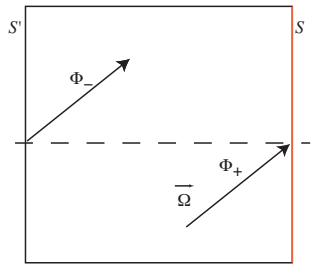

Figure 1.2: Neutron transmission in a periodic lattice

Since it is impossible to solve the transport equation analytically in the general case we will now look at one approximation that can be used to substantially reduce the complexity of the transport problem.

1.1.2 The Multigroup Approximation

Here we deal with the energy dependence of the neutron flux by subdividing the neutron energy range $0 \leq E \leq E_0$ into G non-uniform sub-intervals called groups. These groups will be classified in such a way that a neutron with energy E will be associated with group g if:

$$E_g \leq E \leq E_{g-1} \ ,$$

with

$$0 = E_G < E_{G-1} < \ldots < E_1 < E_0 \ .$$

In general the choice of the energy limits of the groups is left to the creator of the microscopic cross section library. The number of energy groups considered can be as small as 1 or as large as 10^5 depending on the problem one wants to study. For lattice and supercell calculations such as those performed in DRAGON, the common practice is to use libraries with at most a few hundred energy groups.

The multigroup approximation consists in integrating the transport equation over each energy group. Defining the flux in group g as:

$$\Phi^g(\vec{r},\vec{\Omega}) = \int_{E_g}^{E_{g-1}} \Phi(\vec{r},E,\vec{\Omega})dE \ , \tag{1.15}$$

we can then define the following group averaged macroscopic properties:

$$\Sigma^g(\vec{r}) = \frac{1}{\Phi^g(\vec{r},\vec{\Omega})} \int_{E_g}^{E_{g-1}} dE\Sigma(\vec{r},E)\Phi(\vec{r},E,\vec{\Omega}) \ , \tag{1.16}$$

$$\nu\Sigma_f^g(\vec{r}) = \frac{1}{\Phi^g(\vec{r},\vec{\Omega})} \int_{E_g}^{E_{g-1}} dE\nu\Sigma_f(\vec{r},E)\Phi(\vec{r},E,\vec{\Omega}) \ , \tag{1.17}$$

$$\chi^g(\vec{r}) = \int_{E_g}^{E_{g-1}} dE\chi(\vec{r},E) \ , \tag{1.18}$$

$$\begin{aligned}\Sigma_s^{h\rightarrow g}(\vec{r},\vec{\Omega}'\rightarrow\vec{\Omega}) =& \frac{1}{\Phi^h(\vec{r},\vec{\Omega})} \int_{E_g}^{E_{g-1}} dE \int_{E_h}^{E_{h-1}} dE' \\ & \Sigma_s(\vec{r},E'\rightarrow E,\vec{\Omega}'\rightarrow\vec{\Omega})\Phi(\vec{r},E',\vec{\Omega}) \ .\end{aligned} \tag{1.19}$$

In theory, each of the group average cross sections above should depend on $\vec{\Omega}$ since the flux used in the multigroup condensation also depends on this variable. In practice, we will assume that the number of groups considered is sufficiently large that the dependence on $\vec{\Omega}$ can be neglected in all cases except for $\Sigma_s^{g,h}(\vec{r},\vec{\Omega}'\rightarrow\vec{\Omega})$ which will remain dependent on $\vec{\Omega}$. A second concern is that in order to perform the above condensation, the flux distribution, which is a solution of the transport equation, must be known. Since this condensation takes place outside the lattice code, one may not be concerned by this problem except for the energy range where a large number of

resonances can be found for some isotopes. In that case, an additional self-shielding procedure is required to correct the group condensed cross section.[11]

Using these relations we can write Eq. (1.10) in the following form:

$$\begin{aligned}\vec{\Omega}\cdot\vec{\nabla}\Phi^g(\vec{r},\vec{\Omega})+&\Sigma^g(\vec{r})\Phi^g(\vec{r},\vec{\Omega})\\ &=\sum_{h=1}^{G}\int d^2\Omega'\Sigma_s^{h\to g}(\vec{r},\vec{\Omega}'\to\vec{\Omega})\Phi^h(\vec{r},\vec{\Omega}')\\ &+\frac{1}{k}\chi^g(\vec{r})\sum_{h=1}^{G}\int d^2\Omega'\nu\Sigma_f^h(\vec{r})\Phi^h(\vec{r},\vec{\Omega}')\ ,\end{aligned}\tag{1.20}$$

which is known as the static integro-differential form of the multigroup transport equation. This equation can be cast in the form:

$$\left[\vec{\Omega}\cdot\vec{\nabla}+\Sigma^g(\vec{r})\right]\Phi^g(\vec{r},\vec{\Omega})=Q^g(\vec{r},\vec{\Omega})=Q_s^g(\vec{r},\vec{\Omega})+\frac{1}{k}Q_f^g(\vec{r},\vec{\Omega})\ ,\tag{1.21}$$

where the source term $Q^g(\vec{r},\vec{\Omega})$ is composed of a scattering contribution $Q_s^g(\vec{r},\vec{\Omega})$:

$$Q_s^g(\vec{r},\vec{\Omega})=\sum_{h=1}^{G}\int d^2\Omega'\Sigma_s^{h\to g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega})\Phi^h(\vec{r},\vec{\Omega}')\ ,\tag{1.22}$$

and a fission term $Q_f^g(\vec{r},\vec{\Omega})$

$$Q_f^g(\vec{r},\vec{\Omega})=\chi^g\sum_{h=1}^{G}\nu\Sigma_f^h(\vec{r})\int d^2\Omega'\Phi^h(\vec{r},\vec{\Omega}')\ .\tag{1.23}$$

1.2 The Collision Probability Technique

The collision probability technique is based on the integral form of the transport equation. Here, in order to simplify the notation we will assume that the source is isotropic, namely:

$$\int_{4\pi}d^2\Omega Q^g(\vec{r},\vec{\Omega})=\frac{1}{4\pi}\int_{4\pi}d^2\Omega q^g(\vec{r})=q^g(\vec{r})\ ,$$

where $q(\vec{r})$ represents the scalar source at point $\vec{r}$. In this case the transport equation takes the form

$$\left[\vec{\Omega}\cdot\vec{\nabla}+\Sigma^g(\vec{r})\right]\Phi^g(\vec{r},\vec{\Omega})=\frac{1}{4\pi}q^g(\vec{r})\ .\tag{1.24}$$

1.2.1 The Integral Transport Equation

Now, we would like to evaluate the neutron flux at a point $\vec{r}$ due to neutrons created at any point $\vec{r}'$ surrounding it. Since the neutron flux $\Phi^g(\vec{r}', \vec{\Omega})$ at any point $\vec{r}'$ satisfies the transport equation, we can write the drift operator as $\vec{\Omega} \cdot \vec{\nabla}' = -d/dR$ for each point $\vec{r}' = \vec{r} - R\vec{\Omega}$ and the local transport equation becomes:[12]

$$\left[-\frac{d}{dR} + \Sigma^g(\vec{r} - R\vec{\Omega})\right] \Phi^g(\vec{r} - R\vec{\Omega}, \vec{\Omega}) = \frac{1}{4\pi} q^g(\vec{r} - R\vec{\Omega}) \ ,$$

which can be integrated to:[13]

$$\begin{aligned}\Phi^g(\vec{r}, \vec{\Omega}) =& e^{-\tau^g(R_S)} \Phi^g_-(\vec{r}'_S, \vec{\Omega}) \Theta(\vec{r}, \vec{r}'_S, \vec{\Omega}) \\ &+ \frac{1}{4\pi} \int_0^R e^{-\tau^g(R')} q^g(\vec{r}') \Theta(\vec{r}, \vec{r}', \vec{\Omega}) dR' \ ,\end{aligned} \tag{1.25}$$

where $\Phi^g_-(\vec{r}'_S, \vec{\Omega})$ represents the incoming flux at a point $\vec{r}'_S$ on the boundary surface S surrounding V and

$$\tau^g(R) = \int_0^R \Sigma^g(\vec{r} - R'\vec{\Omega}) dR' \ , \tag{1.26}$$

$$\Theta(\vec{r}, \vec{r}', \vec{\Omega}) = \begin{cases} 1 & \text{if } \vec{r} = \vec{r}' + R'\vec{\Omega} \\ 0 & \text{otherwise} \end{cases} \ , \tag{1.27}$$

Similarly, one can obtain an equation for the outgoing angular flux $\phi_+(\vec{r}_S, \vec{\Omega}')$ at S by taking the limit $\vec{r} = \vec{r}_S$ in equation Eq. (1.25):

$$\begin{aligned}\Phi^g_+(\vec{r}_S, \vec{\Omega}) =& e^{-\tau^g(R_S)} \Phi^g_-(\vec{r}'_S, \vec{\Omega}) \Theta(\vec{r}_S, \vec{r}'_S, \vec{\Omega}) \\ &+ \int_0^R e^{-\tau^g(R')} q^g(\vec{r}') \Theta(\vec{r}_S, \vec{r}', \vec{\Omega}) dR' \ .\end{aligned} \tag{1.28}$$

The boundary conditions that will be used in this case are identical to those used for the integro-differential transport equation namely

$$\Phi^g_-(\vec{r}_S, \vec{\Omega} - 2(\vec{N}_S \cdot \vec{\Omega})) = \beta^g(\vec{r}_S) \Phi^g_+(\vec{r}_S, \vec{\Omega}) \ , \tag{1.29}$$

and

$$\Phi^g_-(\vec{r}'_S, -\vec{\Omega}) = T^g(\vec{r}'_S, \vec{r}_S) \Phi^g_+(\vec{r}_S, \vec{\Omega}) \ . \tag{1.30}$$

1.2.2 The Collision Propability Approximations

The collision probability (CP) technique is one of the most widely used method for solving the integral form of the transport equation.[11–19] It consists in dividing the domain into N_V regions of volume V_j where one assumes that the cross sections and the sources are constant:

$$\begin{aligned}\Sigma^g(\vec{r}) =& \Sigma_j^g \qquad \text{for } \vec{r} \in V_j \ , \\ q^g(\vec{r}) =& q_j^g \qquad \text{for } \vec{r} \in V_j \ .\end{aligned}$$

We will also subdivide the external boundary S into N_S surfaces of area S_α. The angular flux on these surfaces is then approximated using a limited series expansion in terms of half-range spherical harmonics $\psi^\nu(\vec{\Omega}, \vec{N}_\pm)$. In DRAGON, this expansion is limited to the first order term for 3-D geometries, namely:[13,20]

$$\Phi_\pm^g(\vec{r}_S, \vec{\Omega}) = \frac{1}{4\pi}\phi_\pm^g(\vec{r}_S) \ . \tag{1.31}$$

We will also define the average scalar flux inside each region as:

$$\phi_i^g = \frac{1}{V_i}\int_{V_i} d^3r \int d^2\Omega \Phi^g(\vec{r}, \vec{\Omega}) \ , \tag{1.32}$$

while the average angular flux on each surface will be written as:

$$\phi_{\pm,\alpha}^g = \frac{4}{S_\alpha}\int_{S_\alpha} d^2r \int_{(\vec{\Omega}\cdot\vec{N}_\pm)>0} d^2\Omega(\vec{\Omega}\cdot\vec{N}_\pm)\Phi_\pm^g(\vec{r}_S, \vec{\Omega}) \ . \tag{1.33}$$

We can now integrate Eq. (1.25) over each volume V_i and direction $\vec{\Omega}$ to obtain the following relation for the average scalar flux in region i:

$$\begin{aligned}V_i\phi_i^g =& \sum_{\alpha=1}^{N_S}\int_{V_i}\int_{S_\alpha} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_-)\phi_{-,\alpha}^g\Theta(\vec{r}, \vec{r}'_S, \vec{\Omega})d^3rd^2r' \\ &+ \sum_{j=1}^{N_V}\int_{V_i}\int_{V_j} \frac{e^{-\tau^g(R)}}{4\pi R^2}q_j^g\Theta(\vec{r}, \vec{r}', \vec{\Omega})d^3r'd^3r \ ,\end{aligned} \tag{1.34}$$

where we assumed that $\phi_-^g(\vec{r}_S)$ is constant and equal to $\phi_{-,\alpha}^g$ on each surface S_α.

We can also obtain an equation for the outgoing angular flux on the surface α in the form:

$$\begin{aligned}\frac{S_\alpha}{4}\phi_{+,\alpha}^g =& \sum_{\beta=1}^{N_S}\int_{S_\alpha}\int_{S_\beta} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_+)(\vec{\Omega}\cdot\vec{N}_-)\phi_{-,\beta}^g\Theta(\vec{r}_S, \vec{r}'_S, \vec{\Omega})d^2rd^2r' \\ &+ \sum_{j=1}^{N_V}\int_{S_\alpha}\int_{V_j} \frac{e^{-\tau(R)}}{4\pi R^2}(\vec{\Omega}\cdot\vec{N}_+)q_j^g\Theta(\vec{r}_S, \vec{r}', \vec{\Omega})d^3r'd^2r \ .\end{aligned} \tag{1.35}$$

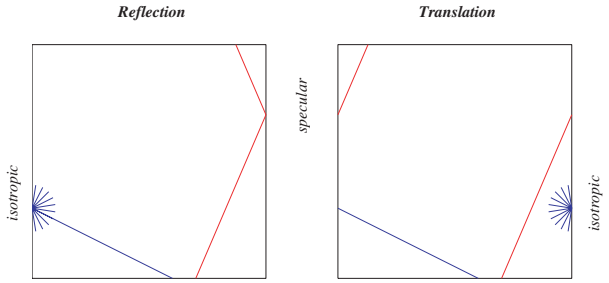

Figure 1.3: Surface flux behaviour for specular and isotropic boundary conditions

A final comment concerns the discretization of the boundary conditions described in Eq. (1.29). Using the approximations described above, we may write Eqs. (1.29) and (1.30) as

$$\phi^g_{-,\beta} = A^g_{\alpha\beta}\phi^g_{+,\alpha} \ , \tag{1.36}$$

where $A^g_{\alpha\beta}$ is the group dependent albedo/transmission matrix. One can find in Figure 1.3 a graphical description of the behaviour of the reflected or transmitted surface flux when specular (exact) and isotropic (white or approximate) boundary conditions are considered.

Based on Eqs. (1.34) and (1.35) we can now define four types of probabilities:

$$\tilde{p}^g_{ij} = V_i p^g_{ij} = \int_{V_i}\int_{V_j} \frac{e^{-\tau^g(R)}}{4\pi R^2}\Theta_i\Theta_j d^3r' d^3r \ , \tag{1.37}$$

$$\tilde{p}^g_{i\alpha} = V_i p^g_{i\alpha} = \int_{V_i}\int_{S_\alpha} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_-)\Theta_i\Theta_\alpha d^3r' d^2r \ , \tag{1.38}$$

$$\tilde{p}^g_{\alpha i} = \frac{S_\alpha}{4} p^g_{\alpha i} = \int_{S_\alpha}\int_{V_i} \frac{e^{-\tau^g(R)}}{4\pi R^2}(\vec{\Omega}\cdot\vec{N}_+)\Theta_\alpha\Theta_i d^2r' d^3r \ , \tag{1.39}$$

$$\tilde{p}^g_{\alpha\beta} = \frac{S_\alpha}{4} p^g_{\alpha\beta} = \int_{S_\alpha}\int_{S_\beta} \frac{e^{-\tau^g(R_S)}}{4\pi R_S^2}(\vec{\Omega}\cdot\vec{N}_-)(\vec{\Omega}\cdot\vec{N}_+)\Theta_\alpha\Theta_\beta d^2r d^2r' \ , \tag{1.40}$$

where

$$\Theta_i = \begin{cases} 1 & \text{if } \vec{r} \text{ is in } V_i \\ 0 & \text{otherwise} \end{cases} \ ,$$

$$\Theta_\alpha = \begin{cases} 1 & \text{if } \vec{r}\,' \text{ is on } S_\alpha \\ 0 & \text{otherwise} \end{cases} \ .$$

The transport equation then becomes:

$$\phi_i^g = \sum_{\alpha=1}^{N_S} p_{i\alpha}^g \phi_{-,\alpha}^g + \sum_{j=1}^{N_V} p_{ij}^g q_j^g \ , \tag{1.41}$$

$$\phi_{+,\alpha}^g = \sum_{\beta=1}^{N_S} p_{\alpha\beta}^g \phi_{-,\beta}^g + \sum_{j=1}^{N_V} p_{\alpha j}^g q_j^g \ , \tag{1.42}$$

or in matrix notation:

$$\vec{\phi}^g = \mathbf{P}_{vs}^g \vec{J}_-^g + \mathbf{P}_{vv}^g \vec{q}^g \ , \tag{1.43}$$

$$\vec{J}_+^g = \mathbf{P}_{ss}^g \vec{J}_-^g + \mathbf{P}_{sv}^g \vec{q}^g \ . \tag{1.44}$$

where $\vec{\phi}^g$, $\vec{J}_\pm^g$ and $\vec{q}^g$ are vectors containing ϕ_i^g, ϕ_α^g and q_i^g respectively. Similarly, the matrices $\mathbf{P}_{vv}^g$, $\mathbf{P}_{vs}^g$, $\mathbf{P}_{sv}^g$ and $\mathbf{P}_{ss}^g$ contain the elements p_{ij}^g, $p_{i\alpha}^g$, $p_{\alpha i}^g$ and $p_{\alpha\beta}^g$ respectively.

1.2.3 Symmetry and Conservation Relations

Now using the symmetry of the optical path length, we can directly derive the following symmetry relations:

$$V_i p_{ij}^g = \tilde{p}_{ij}^g = \tilde{p}_{ji}^g = V_j p_{ji}^g \ , \tag{1.45}$$

$$V_i p_{i\alpha}^g = \tilde{p}_{i\alpha}^g = \tilde{p}_{\alpha i}^g = \frac{S_\alpha}{4} p_{\alpha i}^g \ , \tag{1.46}$$

$$\frac{S_\alpha}{4} p_{\alpha\beta}^g = \tilde{p}_{\alpha\beta}^g = \tilde{p}_{\beta\alpha}^g = \frac{S_\beta}{4} p_{\beta\alpha}^g \ . \tag{1.47}$$

These collision probabilities also satisfy the following classical conservation properties:

$$\sum_{\alpha=1}^{N_\alpha} p_{i\alpha}^g + \sum_{j=1}^{N_j} p_{ij}^g \Sigma_j^g = 1 \ , \tag{1.48}$$

$$\sum_{\beta=1}^{N_\beta} p_{\alpha\beta}^g + \sum_{i=1}^{N_i} p_{\alpha i}^g \Sigma_i = 1 \ . \tag{1.49}$$

1.3 Boundary Conditions

Before considering the boundary conditions let us look back at the assumptions we used in deriving the collision probability equations. First we assumed that the source inside each region is uniform. A second assumption is that the angular flux on outer surfaces can be represented by a single term in the series expansions for the current in spherical harmonics. This means that

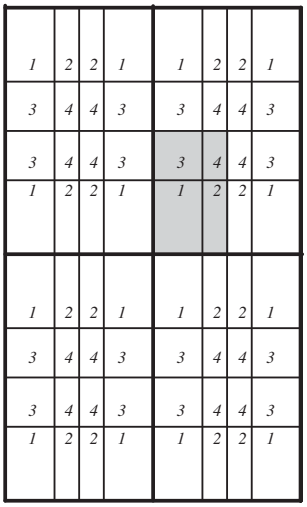

Figure 1.4: Example of geometry unfolding for mirror reflection

a cosine distribution is assumed for the angular flux on these surfaces. Note that this approximation is exact for the case where the incoming angular flux on an external surface vanishes. The angular flux distribution at the interface between two regions is generally not considered explicitly in collision probability calculations this being equivalent to using an infinite spherical harmonics series expansion for the angular fluxes on at region interfaces.

As a result, two different options can be considered when treating the boundary conditions in DRAGON. First, one can try to eliminate as many as the outer surfaces as possible (surface without vacuum boundary conditions) using the remaining boundary conditions. One can find in Figure 1.4 an example of a two dimensional geometry with total reflection boundary conditions that has been unfolded 4 times along both direction. In theory, this unfolding can be extended to infinity and the collision probability matrix for the resulting infinite system computed. In practice, DRAGON cannot compute the collision probability matrix for infinite systems in three dimensions. A similar process can also be used for a system with periodic boundary conditions as illustrated in Figure 1.5. In both cases, all the external surfaces become interfaces and no

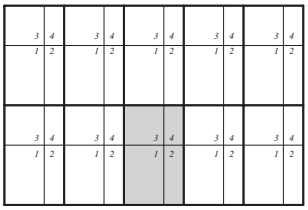

Figure 1.5: Example of geometry unfolding for periodic boundary conditions

angular approximation for the surface flux is required.

The second option is to use explicitly Eq. (1.36) to approximate the angular flux distribution on some surfaces (see Figure 1.3). Since the boundary conditions are in the form of a relation between the outgoing angular flux $\phi^g_{+,\beta}$ on a surface S_β and the incoming angular flux $\phi^g_{-,\alpha}$ on a different surface S_α (see Eq. (1.36)):

$$\vec{J}^g_- = \mathbf{A}^g \vec{J}^g_+ \ . \tag{1.50}$$

This relation will be used to eliminate the angular flux from the collision probability equations.

After multiplying Eq. (1.44) by $\mathbf{A}^g$ one obtains:

$$\vec{J}^g_- = \mathbf{A}^g \mathbf{P}^g_{ss} \vec{J}^g_- + \mathbf{A}^g \mathbf{P}^g_{sv} \vec{q}^g \ ,$$

which, after inversion yields:

$$\vec{J}^g_- = (\mathbf{I} - \mathbf{A}^g \mathbf{P}^g_{ss})^{-1} \mathbf{A}^g \mathbf{P}^g_{sv} \vec{q}^g \ ,$$

where $\mathbf{I}$ is the identity matrix. After substitution in Eq. (1.43) one obtains

$$\vec{\phi}^g = \left(\mathbf{P}^g_{vv} + \mathbf{P}^g_{vs} (\mathbf{I} - \mathbf{A}^g \mathbf{P}^g_{ss})^{-1} \mathbf{A}^g \mathbf{P}^g_{sv} \right) \vec{q}^g = \mathbf{P}^g_{c,vv} \vec{q}^g \ , \tag{1.51}$$

where

$$\mathbf{P}^g_{c,vv} = \left(\mathbf{P}^g_{vv} + \mathbf{P}^g_{vs} ((\mathbf{A}^g)^{-1} - \mathbf{P}^g_{ss})^{-1} \mathbf{P}^g_{sv} \right) \ , \tag{1.52}$$

is known as the complete collision probability matrix.

1.4 Cross Sections Considerations

Two types of multigroup cross-section database can be read by DRAGON. These are the database for the mixture macroscopic cross-section and the database for isotope microscopic cross-section that contains itself a macroscopic cross-section database. In fact, most of the modules in DRAGON only deal with the macroscopic cross-section (stand-alone or imbedded in a microscopic) database.

Thus, in the first part of this section we will concentrate our efforts on describing the general contents of the macroscopic database required for 3–D transport calculations. In the second part we will present the various formats of microscopic cross-section libraries that can be processed by DRAGON and briefly discuss the resonance self-shielding model used in this code.

1.4.1 Macroscopic Cross-Section Requirements

A macroscopic cross section database (or MACROLIB) should contain minimally the following cross sections:

- The multigroup total cross section associated with each mixture Σ^g_m;

- The isotropic component of the multigroup scattering cross section associated with each mixture $\Sigma^{h\to g}_{m,s,0}$ defined as:

$$\Sigma^{h\to g}_{m,s,0} = \int_{4\pi} d^2\Omega^2 \Sigma^{h\to g}_{m,s}(\vec{\Omega}\,' \to \vec{\Omega}) P_0(\vec{\Omega}\,' \cdot \vec{\Omega}) = 2\pi \int_{-1}^{1} d\mu \Sigma^{h\to g}_{m,s}(\mu) P_0(\mu) \ ,$$

  where $P_l(\mu)$ are Legendre polynomials that satisfy the following orthogonality conditions:

$$\int_{-1}^{1} d\mu P_l(\mu) P_m(\mu) = \frac{2\delta_{l,m}}{(2l+1)} \ ;$$

- The product of the average neutron emitted per fission with the multigroup fission cross section associated with each mixture $\nu\Sigma^g_{m,f}$;

- The multigroup fission spectrum associated with each mixture χ^g_m.

In DRAGON, the linearly anisotropic contribution to the scattering cross section $\Sigma^{h\to g}_{m,s,1}$:

$$\Sigma^{h\to g}_{m,s,1} = \int_{4\pi} d^2\Omega^2 \Sigma^{h\to g}_{m,s}(\vec{\Omega}\,' \to \vec{\Omega}) P_1(\vec{\Omega}\,' \cdot \vec{\Omega}) = 2\pi \int_{-1}^{1} d\mu \Sigma^{h\to g}_{m,s}(\mu) P_1(\mu) \ ,$$

is generally not required. In fact for DRAGON, like most collision probability based transport codes, the effect of scattering anisotropy is taken into account approximately using the transport correction. All the transport calculations are performed using transport corrected total ($\tilde{\Sigma}^g_m$) and scattering ($\tilde{\Sigma}^{h\to g}_{m,s,0}$) cross sections:

$$\begin{aligned} \tilde{\Sigma}^g_m &= \Sigma^g_m - \Sigma^g_{m,\text{tc}} \ , \\ \tilde{\Sigma}^{h\to g}_{m,s,0} &= \Sigma^{h\to g}_{m,s,0} - \delta^{gh}\Sigma^g_{m,\text{tc}} \ . \end{aligned}$$

if the multigroup transport correction associated with each mixture $\Sigma^g_{m,\text{tc}}$ is also provided on the MACROLIB. Note that B_1 leakage calculations are special in that they require the presence of the linearly anisotropic contribution to the scattering cross section on the database (see Section 3.3).

Finally, some library also provide cross section for multi-neutron production reactions such as

$$\mathrm{X}^Z_A + \mathrm{n}^0_1 \to \mathrm{X}^Z_{A-1} + 2\mathrm{n}^0_1 + \gamma \ .$$

DRAGON does not treat explicitly the macroscopic cross sections (here $\Sigma^g_{(\text{n,2n})}$) associated with such reactions and assumes that the scattering cross section is already corrected to take into account this effect:

$$\tilde{\Sigma}^{h\to g}_{m,s,0} = \Sigma^{h\to g}_{m,s,0} + 2\delta^{gh}\Sigma^g_{m,(\text{n,2n})} \ .$$

where the factor 2 indicates that two neutrons are produced by this reaction. Note that the DRAGON module that processes the microscopic cross section database will automatically include the contribution from multi-neutron production reactions in the macroscopic scattering cross section.

The method for creating the macroscopic cross section database required for general transport calculations in DRAGON is not unique. For example, the `MAC:` module can generate such a data base using the multigroup mixture cross section provided directly in the input file or a GOXS format file compatible with the TRANSX transport code.[21] Alternatively, and possibly, the most frequently used method for generating the MACROLIB is via the `LIB:` module with a microscopic cross section library (see Section 1.4.2 for more information on this subject). Finally, the module that performs group condensation and region homogenization (`EDI:`) can also generate a MACROLIB that can be used directly in DRAGON for transport calculations.

Because the contents of the MACROLIB is well documented and public, alternative methods of generating such cross section databases can also be designed. For example, a WIMS-AECL based MACROLIB can be created using the information available on `TAPE16`.[22] In fact, this is the basis of the side-step method that was used at AECL for the validation of DRAGON. Similarly, calculations recently performed by the Canadian Nuclear Safety Commission used a MACROLIB derived from HELIOS output files.[23]

1.4.2 Microscopic Libraries Treatment

As we stated above, DRAGON can also process microscopic cross-section libraries to generate MACROLIB to be used further along the calculation chain. Various library formats are supported by the `LIB:` module of DRAGON including the following three standard formats. The WIMS–AECL microscopic cross-section library in its direct access format.[22] Temperature and dilution interpolation are performed in this case using the same options as those used in the WIMS–AECL code. The main difference between WIMS and DRAGON that arises in the treatment of the library is at the level of the resonance self-shielding calculation (assuming identical dilutions) since WIMS–AECL performs its self-shielding calculations using the absorption resonance integral while in DRAGON the total resonance integral is considered. The burnup information available on the WIMS format library can also be read and processed by DRAGON.

MATXS sequential format libraries[24,25] generally used with the TRANSX transport code can also be processed.[21] In this case the temperature and dilution interpolation are identical to those used in TRANSX. Assuming that the dilution and temperature of each isotope in a given mixture are identical in DRAGON and TRANSX, both codes will lead to the same set of macroscopic cross section for this mixture.

One can also use libraries in the WIMS–D4 format.[26,27] These libraries contain the same type of information as that available on the WIMS–AECL format libraries, including identical temperature and dilution interpolation and a similar decay chain. One of the main advantages of using the WIMS–D4 format is the fact that the format is distributed freely.

Note that in DRAGON, isotopes from a MATXS file can be mixed with isotopes from WIMS-D4 or WIMS–AECL files to generate a mixture. The only restriction imposed on the libraries is that their multigroup structure must be identical to avoid undue mixing of microscopic cross sections.

For libraries that contain dilution dependent resonance integral tables for resonant isotopes, DRAGON can perform self-shielding calculations for the cross sections. The self-shielding procedure implemented is a generalization to arbitrary geometries of the Stamm'ler method.[11,28] This method uses equivalence relations to associate the self-shielded cross-sections in an heterogeneous media with the self-shielded cross section in an homogeneous media. This equivalence relation relies on the assumption that a rational expansion for the fuel to fuel collision probability $p_{II}^g(N_I\sigma_I^{*g})$ of the form

$$p_{II}^g(N_I\sigma_I^{*g}) = \sum_{n=1}^{3} \frac{\alpha_n^g}{N_I\left(\sigma_I^{*g} + \sigma_{en}^g\right)} \ , \tag{1.53}$$

is valid. Here σ_I^{*g} is the total microscopic cross section for the resonant isotope I ($*$ denotes self-shielded cross sections) and N_I is the number density of isotope I.

After the microscopic dilution cross section σ_{en}^g and the coefficients in the expansion α_n^g have been computed, one can evaluate the averaged microscopic dilution cross section

$$\tilde{\sigma}_{en}^g = \left(\sum_{n=1}^{3} \alpha_n^g \sqrt{\sigma_{en}^g}\right)^2 \ , \tag{1.54}$$

which will be used for interpolation in resonance integral tables and for the computation of the resonant fine structure function.

The above technique requires an iterative scheme since σ_I^{*g} is needed to compute $\tilde{\sigma}_{en}^g$. However, a few iterations are generally sufficient to converge the self-shielding process. Once the self-shielded microscopic cross sections for resonant isotopes are known, they can be used to update the microscopic and macroscopic cross sections.

A last comment concerns what is required before the self-shielding process is started. Since the fuel-to-fuel collision probabilities will be computed, the geometry must be specified and tracked beforehand.

Chapter 2

3–D Collision Probability Calculations

The evaluation of the different types of probabilities defined in Eqs. (1.37) to (1.40) all involve an integration over $\vec{r}\,'$ for the initial volume or surface and an integration over $\vec{r}$ for the final volume or surface (see Figure 2.1). Here, for the integral over $\vec{r}$, we will consider spherical

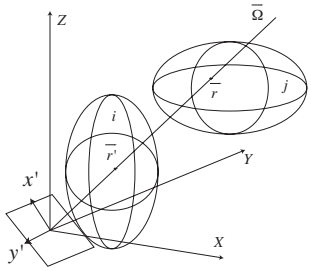

Figure 2.1: General 3–D geometry for collision probability integration

coordinates:

$$\int_{V_j} d^3r\Theta_j = \int_{4\pi} d^2\Omega \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} R^2 dR\Theta_j \ ,$$

$$\int_{S_\beta} (\vec{\Omega}\cdot\vec{N}_-)d^2r\Theta_\beta = \int_{4\pi} d^2\Omega R_S^2\Theta_\beta \ ,$$

where $\vec{\Omega}$ is a solid angle that represents the neutron direction of travel. The integral over $\vec{r}\,'$ (initial volume or surface) will be represented by

$$\int_{V_i} \Theta_i d^3r' = \int dx' \int dy' \int dR'\Theta_i \ ,$$

$$\int_{S_\alpha} \Theta_\alpha(\vec{\Omega}\cdot\vec{N}_+)d^2r' = \int dx' \int dy'\Theta_\alpha \ ,$$

where R' is the distance traveled by the neutron in the region i on a line parallel to the direction $\vec{\Omega}$ while x' and y' define a plane normal to this direction.

Accordingly we can write the various probabilities as:

$$\tilde{p}^g_{ij} = \int_{4\pi} \frac{d^2\Omega}{4\pi} \int dx' \int dy' \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' \int_{R_{j-\frac{1}{2}}}^{R_{j+\frac{1}{2}}} dR\, e^{-\tau^g(R',R)} \Theta_i \Theta_j \ , \tag{2.1}$$

$$\tilde{p}^g_{i\alpha} = \int_{4\pi} \frac{d^2\Omega}{4\pi} \int dx' \int dy' \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' e^{-\tau^g(R',R_S)} \Theta_i \Theta_\alpha \ , \tag{2.2}$$

$$\tilde{p}^g_{\alpha j} = \int_{4\pi} \frac{d^2\Omega}{4\pi} \int dx' \int dy' \int_{R_{j-\frac{1}{2}}}^{R_{j+\frac{1}{2}}} dR e^{-\tau^g(R'_S,R)} \Theta_\alpha \Theta_j \ , \tag{2.3}$$

$$\tilde{p}^g_{\alpha\beta} = \int_{4\pi} \frac{d^2\Omega}{4\pi} \int dx' \int dy' e^{-\tau^g(R'_S,R_S)} \Theta_\alpha \Theta_\beta \ . \tag{2.4}$$

2.1 Collision Probabilities in 3-D

Here, we will first analyze $\tilde{p}^g_{ij}$. According to the notation given in Figure 2.2 we can write:

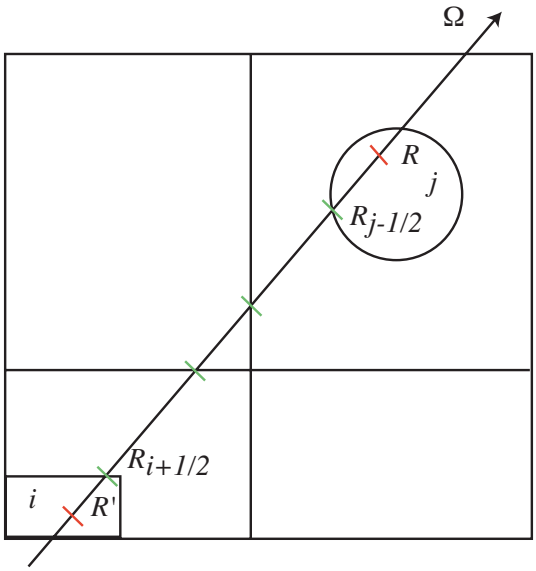

Figure 2.2: Notation for optical path evaluation

$$\tau^g(R', R) = \begin{cases} (R_{i+\frac{1}{2}} - R')\Sigma^g_i + \sum_{k=i+1}^{j-1} \Delta R_k \Sigma^g_k + (R - R_{j-\frac{1}{2}})\Sigma^g_j & \text{for } i < j \\ (R - R')\Sigma^g_i & \text{for } i = j \end{cases} \ , \tag{2.5}$$

where ΔR_k is given by,

$$\Delta R_k = R_{k+\frac{1}{2}} - R_{k-\frac{1}{2}} \ .$$

In the case where $i < j$ and $\Sigma_i^g \neq 0$ and $\Sigma_j^g \neq 0$ we are left with the following expression for $\tilde{p}_{ij}^g$ after the integration over R and R' have been performed:

$$\begin{aligned}\tilde{p}_{ij}^g =&\frac{1}{4\pi\Sigma_i^g\Sigma_j^g}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i\Theta_j \\ &\left[F^g_{i-\frac{1}{2},j+\frac{1}{2}} - F^g_{i-\frac{1}{2},j-\frac{1}{2}} - F^g_{i+\frac{1}{2},j+\frac{1}{2}}(1) + F^g_{i+\frac{1}{2},j-\frac{1}{2}}\right] ,\end{aligned}$$

where F is defined as

$$F^g_{i\pm\frac{1}{2},j\pm\frac{1}{2}} = \exp\left[-\tau^g_{i\pm\frac{1}{2},j\pm\frac{1}{2}}\right] , \tag{2.6}$$

and

$$\begin{aligned}\tau^g_{i\pm\frac{1}{2},j\pm\frac{1}{2}} =&\Sigma_i^g(R_{i+\frac{1}{2}} - R_{i\pm\frac{1}{2}}) + \sum_{k=i+1}^{j-1}\Sigma_k^g(R_{k+\frac{1}{2}} - R_{k-\frac{1}{2}}) + \Sigma_j^g(R_{j\pm\frac{1}{2}} - R_{j-\frac{1}{2}}) \\ =&\Sigma_i^g(R_{i+\frac{1}{2}} - R_{i\pm\frac{1}{2}}) + \tau_{i+\frac{1}{2},j-\frac{1}{2}} + \Sigma_j^g(R_{j\pm\frac{1}{2}} - R_{j-\frac{1}{2}}) .\end{aligned} \tag{2.7}$$

Factoring the term in $\tau_{i+\frac{1}{2},j-\frac{1}{2}}$ from the above equation we obtain:

$$\begin{aligned}\tilde{p}_{ij}^g =&\frac{1}{4\pi\Sigma_i^g\Sigma_j^g}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i\Theta_j \\ &\times\left[1-\exp\left(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}\right)\right]\exp\left(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}}\right)\left[1-\exp\left(-\tau^g_{j-\frac{1}{2},j+\frac{1}{2}}\right)\right] .\end{aligned} \tag{2.8}$$

Three other cases may be considered, namely $\Sigma_i^g = 0$ which leads to:

$$\begin{aligned}\tilde{p}_{ij}^g =&\frac{1}{4\pi\Sigma_j^g}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i\Theta_j \\ &\times\Delta R_i \exp(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}})\left[1-\exp(-\tau^g_{j-\frac{1}{2},j+\frac{1}{2}})\right] ,\end{aligned} \tag{2.9}$$

since the exponential is now independent of R', $\Sigma_j^g = 0$ which gives:

$$\begin{aligned}\tilde{p}_{ij}^g =&\frac{1}{4\pi\Sigma_i^g}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i\Theta_j \\ &\times\Delta R_j \exp(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}})\left[1-\exp(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}})\right] ,\end{aligned} \tag{2.10}$$

since the exponential is now independent of R. Finally when both Σ_i^g and Σ_j^g vanish, the exponential function becomes independent on both R and R' and we obtain

$$\tilde{p}_{ij}^g =\frac{1}{4\pi}\int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i\Theta_j\Delta R_i\Delta R_j \exp(-\tau^g_{i+\frac{1}{2},j-\frac{1}{2}}) . \tag{2.11}$$

In the case where $i = j$, the expression for $\tilde{p}_{ii}$ can be written as

$$\tilde{p}_{ii}^{g} = \frac{1}{4\pi} \int_{0}^{4\pi} d^2\Omega \int dx' \int dy' \int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' \times \left[\int_{R_{i-\frac{1}{2}}}^{R'} dR \Theta_i \Theta_i \exp(-\Sigma_i^g (R' - R)) + \int_{R'}^{R_{i+\frac{1}{2}}} dR \Theta_i \Theta_i \exp(-\Sigma_i^g (R - R')) \right] .$$

The integrals over R and R' then yields:

$$\int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' \int_{R_{i-\frac{1}{2}}}^{R'} dR \exp(-\Sigma_i^g (R' - R)) = \frac{\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}}{\left(\Sigma_i^g\right)^2} - \frac{1}{\left(\Sigma_i^g\right)^2} \left[1 - \exp(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}) \right] ,$$

$$\int_{R_{i-\frac{1}{2}}}^{R_{i+\frac{1}{2}}} dR' \int_{R'}^{R_{i-\frac{1}{2}}} dR \exp(-\Sigma_i^g (R - R')) = \frac{\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}}{\left(\Sigma_i^g\right)^2} - \frac{1}{\left(\Sigma_i^g\right)^2} \left[1 - \exp(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}) \right] ,$$

and we obtain:

$$\tilde{p}_{ii}^{g} = \frac{1}{2\pi \left(\Sigma_i^g\right)^2} \int_{0}^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_i \times \left[\tau^g_{i-\frac{1}{2},i+\frac{1}{2}} - \left(1 - \exp(-\tau^g_{i-\frac{1}{2},i+\frac{1}{2}}) \right) \right] . \tag{2.12}$$

In the case where $\Sigma_i^g = 0$ the above equation reduces to:

$$\tilde{p}_{ii}^{g} = \frac{1}{4\pi} \int_{0}^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_i \left(\Delta R_i \right)^2 . \tag{2.13}$$

Next, let us consider the leakage probability $\tilde{p}_{i\alpha}^{g}$. Using the notation of Figure 2.2 we can write:

$$\tau^g(R', R_\alpha) = (R_{i+\frac{1}{2}} - R')\Sigma_i^g + \sum_{k=i+1}^{N_V} \Delta R_k \Sigma_k^g , \tag{2.14}$$

for the optical path associated with $\tilde{p}_{i\alpha}^{g}$. Using the notation for the surface integral presented before, one obtains after integration over R':

$$\tilde{p}_{i\alpha}^{g} = \frac{1}{4\pi\Sigma_i^g} \int_{0}^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_\alpha \exp(-\tau^g_{i+\frac{1}{2},\alpha}) \left[1 - \exp(-\tau^g_{j-\frac{1}{2},j+\frac{1}{2}}) \right] . \tag{2.15}$$

In the case where $\Sigma_i^g = 0$ the expression for $\tau^g(R)$ becomes independent of R' and we obtain

$$\tilde{p}_{i\alpha}^{g} = \frac{1}{4\pi} \int_{0}^{4\pi} d^2\Omega \int dx' \int dy' \Theta_i \Theta_\alpha \Delta R_i \exp(-\tau^g_{i+\frac{1}{2},\alpha}) . \tag{2.16}$$

Finally, for the transmission probability we will have:

$$\tau^g(R'_\alpha, R_\beta) = \sum_{k=1}^{N_V} \Delta R_k \Sigma_k^g = \tau_{\alpha,\beta}^g \,, \tag{2.17}$$

for the optical path associated with $\tilde{p}_{\alpha\beta}^g$. We can then write

$$\tilde{p}_{\alpha\beta}^g = \frac{1}{4\pi} \int_0^{4\pi} d^2\Omega \int dx' \int dy' \Theta_\alpha \Theta_\beta \exp(-\tau_{\alpha,\beta}^g) \,. \tag{2.18}$$

2.2 Numerical Quadrature and Tracking

The code DRAGON can be used to evaluate numerically the collision probability matrix associated with various types of geometries.[6,7] Because of the complexity of the calculation procedure, this work is generally divided between two different modules. The first module, called the tracking module, consist in selecting the specific numerical quadrature scheme that will be used to analyze a given geometry and to generate the integration points and weights required for a specific problem. In the second module, called the assembly module, the summation process associated with the numerical quadrature is performed and the multigroup collision probabilities are avaluated. Here, we will present the specific numerical quadrature technique used for 3–D calculations. The discussion on how the assembly module uses this information to generate the collision probability matrices will be presented in the next section.

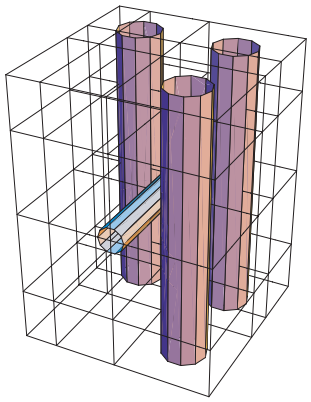

Figure 2.3: Simplified CANDU adjuster rod geometry

Here, we will consider the case where a 3–D Cartesian region is subdivided into a number $N_x N_y N_z$ of Cartesian sub-regions which may contain embedded annular regions over which the cross sections are assumed uniform (see Figure 2.3). The size of each of these regions does not need to be identical, however, they must be arranged in such a way as to form a regular mesh in both the X, Y and Z direction. To each Cartesian region will be associated 6 surfaces denoted respectively by $x_\pm$, $y_\pm$ and $z_\pm$ as described in Figure 2.4.

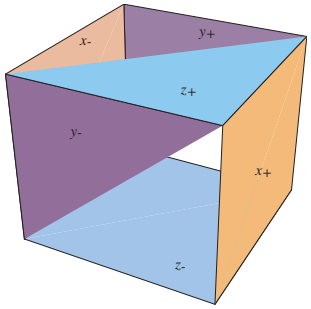

Figure 2.4: Surface notation for simple 3–D geometry

2.2.1 Numerical Quadrature

As one can see in Eq. (2.8), (2.15) and (2.18) the evaluation of $\tilde{p}^g_{ij}$, $\tilde{p}^g_{i\alpha}$ and $\tilde{p}^g_{\alpha\beta}$ involves a 4-dimensional integration that will be performed numerically within the code DRAGON. For the integration over the solid angle

$$\vec{\Omega} = (\cos\varphi \sin\theta, \sin\varphi \sin\theta, \cos\theta) \ ,$$

we will consider only the upper half-sphere, namely

$$0 \leq \varphi \leq 2\pi \qquad \text{and} \qquad 0 \leq \theta \leq \pi/2 \ .$$

This is because the contributions to the integral arising from $\vec{\Omega}$ directed towards the lower half-sphere will be associated with the probability $\tilde{p}^g_{ji}$ which is symmetric to $\tilde{p}^g_{ij}$. Here the integral over the solid angle $d^2\Omega$ is discretized using the equal weight EQ_N quadrature technique developed for S_n method.[29,30] For a quadrature of order N_Ω, a total of $4N_\Omega(N_\Omega+2)/8$ points will be selected in the upper half-sphere:

$$\frac{1}{4\pi}\int_0^{4\pi} d^2\Omega F(\vec{\Omega}) = W_\Omega \sum_i^{N_\Omega(N_\Omega+2)/8} \left[F(\vec{\Omega}_{1,i}) + F(\vec{\Omega}_{2,i}) + F(\vec{\Omega}_{3,i}) + F(\vec{\Omega}_{4,i})\right] \ , \qquad (2.19)$$

where the weight W_Ω is given by:

$$W_\Omega = \frac{1}{N_\Omega(N_\Omega+2)/2} \ .$$

and direction $\vec{\Omega}_{k,i}$ is written in terms of its director cosines:

$$\begin{aligned}
\vec{\Omega}_{1,i} &= (\cos(\Omega_{x,m}), \cos(\Omega_{y,n}), \cos(\Omega_z)) = (u^x_m, u^y_n, u^z) \ , \\
\vec{\Omega}_{2,i} &= (-\cos(\Omega_{x,m}), \cos(\Omega_{y,n}), \cos(\Omega_z)) = (-u^x_m, u^y_n, u^z) \ , \\
\vec{\Omega}_{3,i} &= (\cos(\Omega_{x,m}), -\cos(\Omega_{y,n}), \cos(\Omega_z)) = (u^x_m, -u^y_n, u^z) \ , \\
\vec{\Omega}_{4,i} &= (-\cos(\Omega_{x,m}), -\cos(\Omega_{y,n}), \cos(\Omega_z)) = (-u^x_m, -u^y_n, u^z) \ ,
\end{aligned}$$

such that

$$u^z = \sqrt{1 - (u_m^x)^2 + (u_n^y)^2} \, ,$$

and $\Omega_{x,i}$ and $\Omega_{y,i}$ are located in the first quadrant, namely:

$$0 < \varphi < \pi/2 \qquad \text{and} \qquad 0 < \theta < \pi/2 \, .$$

The second step consists in discretizing the dx' and dy' integral over a plane normal to the direction $\vec{\Omega}$ (see Figure 2.5). The integration limits for these integrals will be specified in such

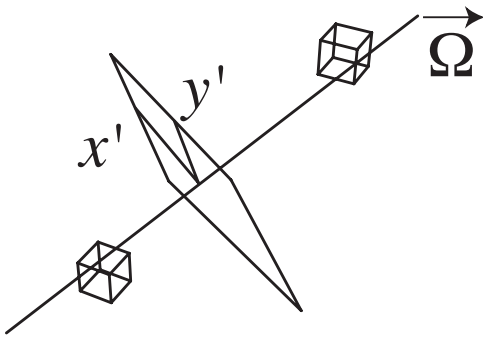

Figure 2.5: Tracking in a plane normal to direction $\vec{\Omega}$

a way that they are independent of the specific integration direction. Accordingly, after locating the center of the cell (x_c, y_c, z_c), we will compute h_+, the radius of the smallest sphere centered at (x_c, y_c, z_c) surrounding the cell. Then the integration limits will be given by:

$$-h_+ \leq x' \leq h_+ \qquad \text{and} \qquad -h_+ \leq y' \leq h_+ \, .$$

Selecting a tracking density of ρ_p will be equivalent to selecting N_p^2 spatial points such that:

$$N_p = (2\sqrt{\rho_p} h_+) + 1 \, . \tag{2.20}$$

As a result the effective spacing between the tracks will be given by:

$$\delta = \frac{2h_+}{N_p} \, , \tag{2.21}$$

and the quadrature points associated with the tracking line l_{mn} will be given by:

$$u_m^x = \left(\frac{2m-1}{2}\right)\delta \,, \tag{2.22}$$

$$u_n^y = \left(\frac{2n-1}{2}\right)\delta \,, \tag{2.23}$$

such that

$$\int dx' \int dy' F(x',y') = W_p \sum_{m=1}^{N_p}\sum_{n=1}^{N_p} F(u_m^x, u_n^y) \,, \tag{2.24}$$

and the weight W_p is given by:

$$W_p = \delta^2 \,.$$

A final point concerns the specific location of the x' and y' axis. The (x',y') plane is defined arbitrarily with respect to the direction $\vec{\Omega}$. A rotation of the plane around the z' axis defined by $\vec{\Omega}$ will yield a new plane which can also be used for the $dx'dy'$ integration. In DRAGON, we chose to ensure that the integral over a cubic cell is invariant under a rotation of the integration plane. Thus, for each direction $\vec{\Omega}$ three different integration planes are selected and tracked successively. As a result the weight W_p associated with each track direction is reduced by a factor of 3. These planes are selected in such a way that the x' axis lies successively in the (x,y), (z,x) and (y,z) planes (see Figure 2.6). Defining $\vec{\Omega}'_x$, $\vec{\Omega}'_y$ and $\vec{\Omega}'_z$ to be the unit vectors

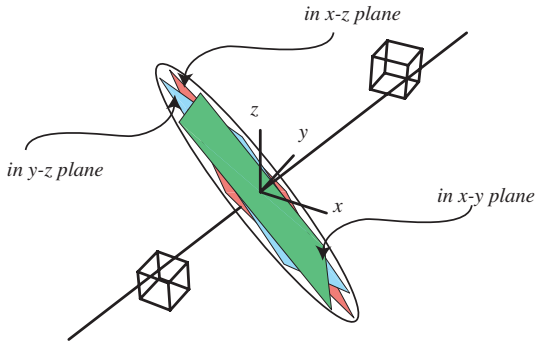

Figure 2.6: Normal plane selection for (x',y') integration

defining the direction of the axis x', y' and z' respectively, we can then write:

$$\vec{\Omega}'_x = \left(-\frac{u_n^y}{\sqrt{1-(u_m^x)^2}} \, , \frac{u_m^x}{\sqrt{1-(u^z)^2}} \, , 0 \right) \, ,$$
$$\vec{\Omega}'_y = \left(\frac{u_m^x u^z}{\sqrt{1-(u^z)^2}} \, , \frac{u_n^y u^z}{\sqrt{1-(u^z)^2}} \, , -\sqrt{1-(u^z)^2} \right) \, ,$$
$$\vec{\Omega}'_z = (u_m^x, u_n^y, u^z) \, ,$$

in the case where the x' axis is in the (x, y) plane. In addition we will use

$$\vec{\Omega}'_x = \left(-\frac{u^z}{\sqrt{1-(u_n^y)^2}} \, , 0 \, , \frac{u_m^x}{\sqrt{1-(u_n^y)^2}} \right) \, ,$$
$$\vec{\Omega}'_y = \left(\frac{u_m^x u_n^y}{\sqrt{1-(u_n^y)^2}} \, , -\sqrt{1-(u_n^y)^2} \, , \frac{u^z u_n^y}{\sqrt{1-(u_n^y)^2}} \right) \, ,$$
$$\vec{\Omega}'_z = (u_m^x, u_n^y, u^z) \, ,$$

or

$$\vec{\Omega}'_x = \left(0 \, , -\frac{u^z}{\sqrt{1-(u_m^x)^2}} \, , \frac{u_n^y}{\sqrt{1-(u_m^x)^2}} \right) \, ,$$
$$\vec{\Omega}'_y = \left(-\sqrt{1-(u_m^x)^2} \, , \frac{u_n^y u_m^x}{\sqrt{1-(u_m^x)^2}} \, , \frac{u^z u_m^x}{\sqrt{1-(u_m^x)^2}} \right) \, ,$$
$$\vec{\Omega}'_z = (u_m^x, u_n^y, u^z) \, ,$$

for the case where the x' axis is in the (z, x) or (y, z) planes respectively.

2.2.2 Tracking

In DRAGON, the 3-D tracking is generally performed using the `EXCELT:` module and proceeds as follows. For each tracking direction $\vec{\Omega}_{k,j}$, three sets of parallel lines $l_{m,n}$ are first generated. These are then followed as they travel through the cell. One then identifies the successive external surfaces and regions crossed by these lines and evaluate the distance $\tilde{l}_{i,m,n}$ crossed by the neutron as it travels through a region (see Figure 2.7). This information can then be saved on a temporary tracking file. There is also the possibility in DRAGON to use these lines to integrate directly the collision probabilities thereby avoiding the use of the tracking file (using the module `EXCELL:` that combines the `EXCELT:` and `ASM:` module).

The tracking module of DRAGON also recognizes some of the cell symmetries defined in the geometry module and acts accordingly. When a cell possesses a mirror reflection with

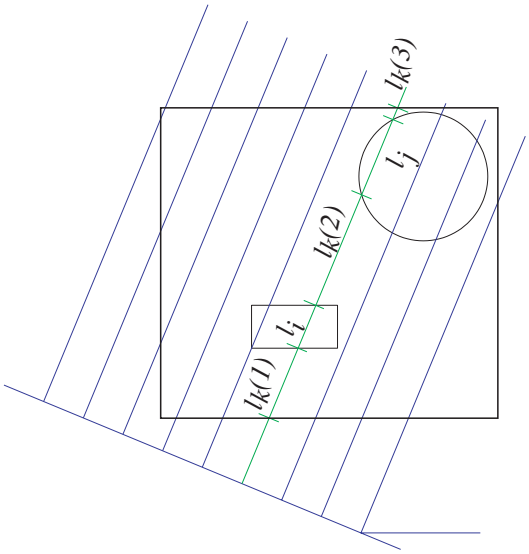

Figure 2.7: 2–D projection of 3–D tracking lines

respect to a plane normal to the x axis, this cell is first unfolded (see Figure 2.8). Then, because of the symmetry of the angular quadrature, only the directions $\vec{\Omega}_{1,i}$ and $\vec{\Omega}_{3,i}$ are tracked. For a cell with a mirror reflection with respect to a plane normal to the y axis the same technique is used but the directions $\vec{\Omega}_{1,i}$ and $\vec{\Omega}_{2,i}$ are now considered. Finally, for a cell with a mirror reflection with respect to a plane normal to the z axis the cell is still unfolded, however, this has no effect on the number of tracking lines to be considered.

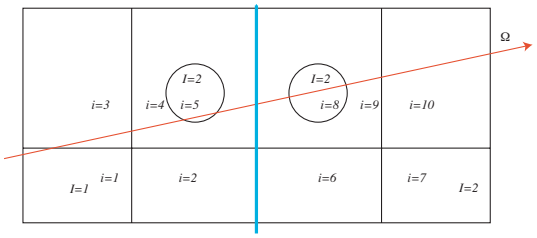

Figure 2.8: Unfolding a symmetric cell in DRAGON

Note that during the tracking procedure DRAGON initially considers the regions in the reflected and original cell to be independent. The final identification of the region number associated with various spatial location will only be performed during the creation of the final tracking file where the track segments are re-normalized, if required, and where successive track segments crossing the same region will be combined. The optional re-normalization of the segment lengths $\tilde{l}_{i,m,n}$ is based on the fact that the surface integral associated with each

tracking direction $\vec{\Omega}_{k,j}$ should satisfy:

$$V_i = \int dx' \int dy' l_i = \frac{W_p}{3} \sum_{m=1}^{N_p} \sum_{n=1}^{N_p} l_{i,m,n} \ ,$$

for each region i. Since this is not necessarily the case for our set of segment lengths $\tilde{l}_{i,m,n}$, they will be re-normalized to $l_{i,m,n}$ using

$$l_{i,m,n} = \left(\frac{V_i}{\tilde{V}_i} \right) \tilde{l}_{i,m,n} \ ,$$

where

$$\tilde{V}_i = \frac{W_p}{3} \sum_{m=1}^{N_p} \sum_{n=1}^{N_p} \tilde{l}_{i,m,n} \ .$$

2.3 Collision Probability Integration

Here we will discuss mainly the collision probability integration module `ASM:` which is associated with the `EXCELT:` tracking module of DRAGON. The procedure we will describe is if fact identical to that used in the `EXCELL:` module (direct collision probability integration without using a tracking file). Both of these collision probability integration modules are programmed in such a way that they can treat explicitly voided regions.

The DRAGON integration routine used in this case is `PIJI3D` (`QIJI3D` for the `EXCELL:` module). The first step in this routine consists in scanning the integration line and computing the contributions to $(\Sigma_i^g)^2 \tilde{p}_{ii}^g$ (see Eq. (2.12)),

$$\frac{1}{2}(\Sigma_i^g)^2 \tilde{p}_{ii}^g = \sum_n W_n \sum_{m \in i} \left(\tau_{i,n,m}^g - \kappa_{i,n,m}^g \right) \ , \tag{2.25}$$

where $W_n = W_\Omega W_p/3$ and

$$\kappa_{i,n,m}^g = \left(1 - \exp\left[-\tau_{i,n,m}^g \right] \right) \ , \tag{2.26}$$

$$\tau_{i,n,m}^g = \Sigma_i^g l_{i,m,n} \qquad \text{with } m \in i \ . \tag{2.27}$$

Note the presence of the factor 1/2 in the above relation. This factor is justified by the fact that we are tracking the line in only one direction thereby taking into account only the contributions associated with $R' < R$. The contribution from $R' > R$, which are identical to those with $R' < R$, will be added later when the collision probability symmetrization process is considered

This is then followed in `PIJI3D` by a second scan of the integration line, to compute the contributions to $\tilde{p}_{ij}^g$ corresponding to the remaining regions in the cell. In this case, a two levels

sweep is considered, namely, for an initial line segment m, the line is analyzed for each segment $m' = m+1, M$ and we compute compute:

$$\Sigma_i^g \Sigma_j^g \tilde{p}_{ij}^g = \sum_n W_n \sum_{m\in i} \sum_{m'\in j} \kappa_{i,n,m}^g \kappa_{n,m+1,m'-1}^g \kappa_{j,n,m'}^g \ , \tag{2.28}$$

using

$$\kappa_{n,m,m'}^g = \prod_{l=m}^{m'} \exp\left[-\tau_{i,n,l}^g\right] \ , \tag{2.29}$$

where $\kappa_{i,n,m}^g$ and $\tau_{i,n,l}^g$ are defined in Eqs. (2.26) and (2.27) respectively.

The contributions to the leakage and escape probabilities are also computed using similar relations:

$$\Sigma_i^g \tilde{p}_{\alpha i}^g = \sum_n W_n \sum_{m\in i} \kappa_{n,1,m-1}^g \kappa_{i,n,m}^g \ , \tag{2.30}$$

$$\Sigma_i^g \tilde{p}_{i\beta}^g = \sum_n W_n \sum_{m\in i} \kappa_{i,n,m}^g \kappa_{n,m+1,M}^g \ , \tag{2.31}$$

$$\tilde{p}_{\alpha\beta}^g = \sum_n W_n \kappa_{n,1,M}^g \ , \tag{2.32}$$

where α and β are respectively the initial and/or final surfaces associated with track n.

In `PIJI3D`, the case where one or more region is voided is treated independently. The general procedure is similar to that above with the following differences. For the evaluation of $\tilde{p}_{ii}^g$, where i is a voided region, instead of using Eq. (2.25), DRAGON uses the following relation:

$$\frac{1}{2}\tilde{p}_{ii}^g = \sum_n \frac{W^n}{2} \sum_{m\in i} \left(\kappa_{i,n,m,v}\right)^2 \ , \tag{2.33}$$

where

$$\kappa_{i,n,m,v} = l_{i,m,n} \qquad \text{with } m \in i \ . \tag{2.34}$$

For the remaining probabilities Eq. (2.28) or Eqs. (2.30) to (2.32) are still used, however for a voided region k the terms of the form $\kappa_{k,n,m}^g$ is now be replaced by $\kappa_{k,n,m,v}$.

This integration procedure is not complete since we have considered only the contributions with $m < m'$ (and $R' < R$ for $\tilde{p}_{ii}^g$). As a result the collision probabilities will contain only half the possible contributions that should have been included. Since the probabilities $\tilde{p}_{ij}^g$ should be symmetric, then all the contributions added to $\tilde{p}_{ij}^g$ for $i \neq j$ should also have been included in $\tilde{p}_{ji}^g$. Accordingly, DRAGON uses:

$$\tilde{p}_{ij}^g = \tilde{p}_{ij}^g + \tilde{p}_{ji}^g \ ,$$

for $i \leq j$. This procedure ensures that the collision probabilites are symmetric and takes care of the factor of 1/2 introduced in Eq. (2.25). The task of completing the collision probability matrices for the isotropic 3–D integration module is also performed in the routine `PIJI3D`.

The routine we described above is called directly by the routine `EXCELP` and reads explicitly the tracking file. In the case where the tracks are not stored in a file but are kept in memory, the routine `QIJI3D` is called via the routine `EXCELL`.

Note that the explicit form of the collision probability computed in these routines depends on the presence or absence of voided regions. In fact the collision and leakage probabilities associated with non-voided region will be restored to their explicit from ($\tilde{p}_{ij}^g$ instead of $\Sigma_i^g \Sigma_j^g \tilde{p}_{ij}^g$) directly in the routine `EXCELP` or `EXCELL`.

2.4 Neutron Conservation and Collision Probability Normalization

Once the transmission, leakage and collision probabilities have been evaluated independently one can verify if the neutron conservation relations described in Eqs. (1.48) and (1.49) are satisfied numerically. In fact one generally obtains:

$$R_j^g = \tilde{P}_{0j}^g - \sum_{\alpha=1}^{N_\alpha} \tilde{P}_{\alpha j}^g - \sum_{i=1}^{N_i} \tilde{P}_{ij}^g \ , \tag{2.35}$$

$$R_\beta^g = \tilde{P}_{0\beta}^g - \sum_{\alpha=1}^{N_\alpha} \tilde{P}_{\alpha\beta}^g - \sum_{i=1}^{N_i} \tilde{P}_{i\beta}^g \ , \tag{2.36}$$

where R_i^g and R_α^g represent the numerical errors on the conservation law for regions and surfaces respectively and we have used

$$\begin{aligned}
\tilde{P}_{ij}^g &= \tilde{P}_{ji}^g = \Sigma_j^g \Sigma_i^g V_i p_{ij}^g \ , \\
\tilde{P}_{\alpha j}^g &= \tilde{P}_{j\alpha}^g = \frac{S_\alpha}{4} \Sigma_i^g p_{\alpha j}^g \ , \\
\tilde{P}_{i\beta}^g &= \tilde{P}_{\beta i}^g = \Sigma_i^g V_i p_{i\beta}^g \ , \\
\tilde{P}_{\alpha\beta}^g &= \tilde{P}_{\beta\alpha}^g = \frac{S_\alpha}{4} p_{\alpha\beta}^g \ , \\
\tilde{P}_{0j}^g &= \Sigma_j^g V_j \ , \\
\tilde{P}_{0\beta} &= \frac{S_\beta}{4} \ .
\end{aligned}$$

These conservation laws can be restored in DRAGON using different collision probabilities normalization schemes that we will now describe.

2.4.1 Diagonal Normalization

The simplest normalization scheme that can be used to restore the conservation laws above while preserving the symmetry relations consist in updating the diagonal entries of the colli-

sion and leakage matrices.[31,32] Using this scheme, one can redefine the diagonal elements of collision probability matrix ($\tilde{P}^g_{D,ii}$ and $\tilde{P}^g_{D,\alpha\alpha}$) as follows:

$$\tilde{P}^g_{D,ii} = \tilde{P}^g_{ii} - R^g_i \ , \tag{2.37}$$

$$\tilde{P}^g_{D,\alpha\alpha} = \tilde{P}^g_{\alpha\alpha} - R^g_\alpha \ . \tag{2.38}$$

Large errors in the conservation equations will result in large changes of the diagonal elements of the collision probability matrix. For small values of p^g_{ii} this scheme may result in non-physical negative probabilities in the final collision probability matrix. Moreover, this scheme cannot be applied to problems involving voided zones.

2.4.2 Gelbard Normalization

Gelbard suggested a different normalization scheme, which is based on a correction using the collision probabilities in the homogeneous limit.[32,33] This scheme has been generalized to

$$\tilde{P}^g_{G,ij} = \tilde{P}^g_{ij} - \frac{1}{\tilde{\Sigma}^g}\left(\tilde{P}^g_{0i}R^g_j + \tilde{P}^g_{0j}R^g_i - \tilde{P}^g_{0i}\tilde{P}^g_{0j}\tilde{R}^g_v\right) \ , \tag{2.39}$$

$$\tilde{P}^g_{G,\alpha\alpha} = \tilde{P}^g_{\alpha\alpha} - \frac{1}{\tilde{S}}(\tilde{P}^g_{0\beta}R^g_\alpha + \tilde{P}^g_{0\alpha}R^g_\beta - \tilde{P}^g_{0\alpha}\tilde{P}^g_{0\beta}\tilde{R}^g_s) \ , \tag{2.40}$$

where

$$\tilde{\Sigma}^g = \sum_i \tilde{P}^g_{0i} \ ,$$

$$\tilde{S} = \sum_\alpha \tilde{P}_{0\alpha} \ ,$$

$$\tilde{R}^g_v = \frac{1}{\tilde{\Sigma}^g}\sum_i R^g_i \ ,$$

$$\tilde{R}^g_s = \frac{1}{\tilde{S}}\sum_\alpha R^g_\alpha \ .$$

This additive scheme can be applied even in voided zones. It will redistribute the corrections to the conservation laws over the complete collision probability matrix. Corrections on the diagonal entries are now weaker than for the previous scheme, but this technique may still produce negative probabilities.

2.4.3 Multiplicative Normalization

Another way to reestablish conservation laws is to define weighting factors w^g_i and w^g_α that will be applied to the collision probability matrix in a multiplicative manner as follows:[32]

$$\tilde{P}^g_{N,ij} = w^g_i w^g_j \tilde{P}^g_{ij} \ , \tag{2.41}$$

$$\tilde{P}^g_{N,\alpha\beta} = w^g_\alpha w^g_\beta \tilde{P}^g_{\alpha\beta} \ . \tag{2.42}$$

This results in a quadratic system that we can solve for the weights w_i^g and w_α^g. The main advantages of this multiplicative normalization is to preserve the null probabilities as well as the relative size of the collision probability matrix entries (since we have $w_i^g \approx 1$) and the overall positivity of the matrix. However, solving the resulting set of quadratic equations for the weight is generally a CPU intensive task.

2.4.4 HELIOS Type Normalization

Finally, one can also consider a simplified multiplicative normalization scheme which does not involve the solution of a system of non-linear equations.[23] Instead of using Eq. (2.41), we will use:

$$\tilde{P}_{H,ij}^g = (w_i^g + w_j^g)\tilde{P}_{ij}^g \,, \tag{2.43}$$

$$\tilde{P}_{H,\alpha\alpha}^g = (w_\alpha^g + w_\beta^g)\tilde{P}_{\alpha\alpha}^g \,. \tag{2.44}$$

The conservation laws will be restored if:

$$w_b^g \left(\tilde{P}_{bb}^g + \sum_a \tilde{P}_{ab}^g \right) = \tilde{P}_{0b}^g + w_b^g \tilde{P}_{bb}^g - \sum_a w_a^g \tilde{P}_{ab}^g = \tilde{P}_{0b}^g - \sum_{a \neq b} w_a^g \tilde{P}_{ab}^g \,, \tag{2.45}$$

where the generic indices a and b run over the regions and surfaces. This system is solved in DRAGON for w_b^g using an iterative process, namely, assuming that $w_a^g(l)$ at iteration l is known for $a \neq b$ we can write:

$$w_b^g(l+1) = \frac{\tilde{P}_{0b}^g - \sum_{a \neq b} w_a^g(l) \tilde{P}_{ab}^g}{\tilde{P}_{bb}^g + \sum_a \tilde{P}_{ab}^g} \,.$$

For $l = 1$, the weights $w_a^g(l) = 0.5$ are all identical, corresponding to the case where the collision probabilities are already normalized. The solution for $w_a^g(l+1)$ is assumed converged when:

$$\max \left(\frac{w_a^g(l+1) - w_a^g(l)}{w_a^g(l+1)} \right) \leq \epsilon \,.$$

This is the default normalization scheme used in DRAGON.

2.5 Boundary Conditions

The integration procedure described above is used to evaluate the collision probabilities associated with a cell isolated in space. In order to take into account the boundary conditions, DRAGON then follows explicitly the technique described in Section 1.3 and computes a

complete collision probability matrix that takes into account the boundary conditions using the relation:

$$\mathbf{P}^g_{c,vv} = \mathbf{P}^g_{vv} + \mathbf{P}^g_{vs}\mathbf{P}^g_{c,ss}\mathbf{P}^g_{sv} \ ,$$

where

$$\mathbf{P}^g_{c,ss} = \left(\mathbf{I} - \mathbf{A}^g\mathbf{P}^g_{ss}\right)^{-1}\mathbf{A}^g = \left((\mathbf{A}^g)^{-1} - \mathbf{P}^g_{ss}\right)^{-1} \ , \tag{2.46}$$

and $\mathbf{A}^g$ is the multigroup reflection/transmission matrix.

The matrix $\mathbf{P}^g_{c,ss}$ is generally computed using the last form of the above equation. However, in the cases where vacuum boundary conditions are applied on different surfaces S_α, a slightly modified expression must be considered. The need for this modification becomes evident if one realizes that for such problems the matrix $\mathbf{A}^g$ is singular and its inverse does not exists. These modifications can be illustrated in the following way.

Let us assume that vacuum boundary conditions are applied to the first n surfaces while reflection or periodic boundary conditions are considered for the remaining surfaces. The matrix $\mathbf{A}^g$ and $\mathbf{P}^g_{ss}$ can then be written in the form:

$$\mathbf{A}^g = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^g_{22} \end{bmatrix} \ ,$$
$$\mathbf{P}^g_{ss} = \begin{bmatrix} \mathbf{P}^g_{11} & \mathbf{P}^g_{12} \\ \mathbf{P}^g_{21} & \mathbf{P}^g_{22} \end{bmatrix} \ .$$

We can therefore write:

$$\left(\mathbf{I} - \mathbf{A}^g\mathbf{P}^g_{ss}\right)^{-1}\mathbf{A}^g = \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & (\mathbf{A}^g_{22})^{-1} - \mathbf{P}^g_{22} \end{bmatrix}\right)^{-1}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \ . \tag{2.47}$$

Accordingly, in the case where vacuum boundary are considered, we will replace the relation for $\mathbf{P}^g_{c,ss}$ described in Eq. (2.46) by the form described in Eq. (2.47).

Chapter 3

Solving the Collision Probability Equations

The multigroup transport equations to be solved in a cell containing N regions form of a linear system (see Eq. (1.51)):[34]

$$\vec{\phi}^g = \mathbf{P}^g_{c,vv}(\vec{q}^g_s + \frac{1}{k}\vec{q}^g_f) \ , \tag{3.1}$$

where $\vec{\phi}^g$ is a vector made up of the fluxes $\phi^g_i, i = 1, N$. Here $\mathbf{P}^g_{c,vv}$ is the multigroup collision probability matrix that includes boundary conditions. The source has been divided as usual into two parts, namely a contribution from scattering $\vec{q}^g_s$:

$$\vec{q}^g_s = \sum_{h=1}^{G} \Sigma^{h\to g}_s \vec{\phi}^h = \sum_{h>g} \Sigma^{h\to g}_s \vec{\phi}^h + \sum_{h<g} \Sigma^{h\to g}_s \vec{\phi}^h + \Sigma^{g\to g}_s \vec{\phi}^g \ , \tag{3.2}$$

and a contribution from fission $\vec{q}^g_f$:

$$\vec{q}^g_f = \chi^g \sum_{h=1}^{G} \nu\Sigma^h_f \vec{\phi}^h \ , \tag{3.3}$$

where $\Sigma^{h\to g}_s$, χ^g and $\nu\Sigma^h_f$ are diagonal matrices (with respect to the regional index i).

Instead of using the complete collision probability matrix $\mathbf{P}^g_{c,vv}$, DRAGON modifies the above equation is such a way as to include in the left hand side of Eq. (3.1) the term involving within-group scattering $\Sigma^{g\to g}_s$. The transport equation then takes the form:

$$\vec{\phi} = \mathbf{W}(\Sigma_{d,s}\vec{\phi} + \Sigma_{u,s}\vec{\phi} + \frac{\chi}{k}\nu\Sigma_f\vec{\phi}) \ , \tag{3.4}$$

where $\mathbf{W}$ is a group-wise block diagonal matrix containing the following submatrices:

$$\mathbf{W}^g = (\mathbf{I} - \mathbf{P}^g_{c,vv}\Sigma^{g\to g}_s)^{-1}\mathbf{P}^g_{c,vv} \ , \tag{3.5}$$

and $\vec{\phi}$ is now a $N \times G$ dimensional vector. The up and down scattering matrices $\Sigma_{u,s}$ and $\Sigma_{d,s}$ are off-diagonal lower and upper triangular group matrices which are themselves diagonal with

respect to i, the region:

$$\Sigma_{u,s} = \begin{cases} \Sigma_s^{h\to g} & \text{for } h > g \\ 0 & \text{otherwise} \end{cases} , \tag{3.6}$$

$$\Sigma_{d,s} = \begin{cases} \Sigma_s^{h\to g} & \text{for } h < g \\ 0 & \text{otherwise} \end{cases} . \tag{3.7}$$

Note that the above equations can be solved directly in a group-by-group fashion if one assumes that a fixed rather than a fission source is present in the cell and that $\Sigma_{u,s}$ vanishes identically. In all the other cases, a two levels iteration process will generally be considered. The outer iteration or power iteration will involve an iteration over the fission source and the determination of the multiplicative constant k. The inner iteration, called the multigroup iteration, will deal with the up-scattering sources.

3.1 The Power Iteration

Here we will consider the case where $\Sigma_{u,s}$ vanishes identically. The transport equation then takes the form:

$$\vec{\phi} = \mathbf{W}(\Sigma_{d,s}\vec{\phi} + \frac{\chi}{k}\nu\Sigma_f\vec{\phi}) . \tag{3.8}$$

Let us now assume as a zero order approximation ($l = 0$) that an approximate distribution for the flux inside the cell $\vec{\phi}(0)$ is known and that $k(0) = 1$. Then, one can solve the above equation for $\vec{\phi}(l)$ in the presence of a fixed source $\vec{q}^g(l-1)$ using

$$\vec{\phi}^g(l) = \mathbf{W}^g \left(\sum_{h<g} \Sigma_s^{h\to g}\vec{\phi}^g + \vec{q}_f^g(l-1) \right) ,$$

such that:

$$\vec{q}_f^g(l) = \frac{\chi^g}{Q(l)} \sum_{h=1}^{G} \nu\Sigma_f^h\vec{\phi}^h(l) ,$$

where

$$Q(l) = \sum_{g=1}^{G}\sum_{i=1}^{N} V_i\chi_i^g \sum_{h=1}^{G} \nu\Sigma_{f,i}^h\phi_i^h(l) .$$

After the solution $\vec{\phi}^g(l)$ for $l > 0$ and $g = 1, G$ has been evaluated, it is possible to compute $Q(l)$ using the above relation and obtain:

$$k(l) = Q(l) .$$

The iteration process is repeated until:

$$k(l) - k(l-1) < \epsilon_1 \ ,$$

and

$$\left| \frac{\vec{\phi}(l)}{k(l)} - \frac{\vec{\phi}(l-1)}{k(l-1)} \right| < \epsilon_2 \left| \frac{\vec{\phi}(l)}{k(l)} \right| \ ,$$

are both satisfied. In DRAGON, the parameters ϵ_1 and ϵ_2 can be defined independently. Any flux distribution can be selected to start the iteration process (a flat flux approximation in space and energy is used by default).

3.2 The Multigroup Iteration

Since the fission source problem can be replaced by a series of fixed source problems as indicated in the previous section, we will now concentrate on the solution of the transport equation in the presence of up-scattering, namely

$$\vec{\phi}^g = \mathbf{W}^g \left(\sum_{h<g} \Sigma_s^{h \to g} \vec{\phi}^h + \sum_{h>g} \Sigma_s^{h \to g} \vec{\phi}^h + \vec{q}^g \right) \ . \tag{3.9}$$

Because we will solve the above equation group by group staring with $g = 1$, the evident iteration procedure to use is the Gauss-Seidel strategy where one solves for $\vec{\phi}^g(l)$ using:

$$\vec{\phi}^g(l) = \mathbf{W}^g \left(\sum_{h<g} \Sigma_s^{h \to g} \vec{\phi}^h(l) + \sum_{h>g} \Sigma_s^{h \to g} \vec{\phi}^h(l-1) + \vec{q}^g \right) \ ,$$

and iterate until:

$$\left| \frac{\vec{\phi}^g(l) - \vec{\phi}^g(l-1)}{\vec{\phi}^g(l)} \right| < \epsilon_3 \ .$$

where in DRAGON the convergence parameter ϵ_3 can be defined by the user. One way to improve on this strategy is to use the flux rebalancing technique which we shall discuss next or a successive over-relaxation scheme with a computed relaxation factor which we will present later in this section.

3.2.1 Flux Rebalancing Technique

Starting from Eq. (3.1) the following neutron conservation relation can be derived

$$\sum_{i=1}^{N} \Sigma_i^g V_i \phi_i^g = \sum_{i=1}^{N} R_i^g V_i \left(\sum_{h=1}^{G} \Sigma_s^{h\to g} \phi_i^h + q_i^g \right) \ , \tag{3.10}$$

where

$$R_i^g = 1 - \sum_{i=1}^{N} \Sigma_j^g p_{c,ij}^g \ , \tag{3.11}$$

and the $p_{c,ij}^g$ are the components of the matrix $\mathbf{P}_{c,vv}^g$.

The solution to the flux equation at thermal iteration l does not necessarily satisfy the above equation since the up-scattering sources are proportional to the flux from a previous iteration. One way to remedy this problem is to define a modified flux:

$$\tilde{\phi}_i^g = \alpha^g \phi_i^g \ , \tag{3.12}$$

such that $\tilde{\phi}_i^g$ satisfies Eq. (3.10). As a result we obtain the following equation for α^g:

$$\sum_{h=1}^{G} M^{h\to g} \alpha^h = \tilde{q}^g \ ,$$

where

$$M^{h\to g} = \sum_{i=1}^{N} R_i^g V_i \left(\Sigma_i^h \delta_{gh} - \Sigma_s^{h\to g} \right) \phi_i^h \ ,$$

while

$$\tilde{q}^g = \sum_{i=1}^{N} R_i^g V_i q_i^g \ .$$

In DRAGON, this linear equation is solved for the rebalancing coefficients α^g at each thermal iteration.

3.2.2 Variational Acceleration

We can combine the Gauss–Seidel iteration scheme with an over-relaxation scheme where we will solve

$$\vec{\Gamma}(l) = \mathbf{W} \left(\Sigma_{d,s} \vec{\Gamma}(l) + \Sigma_{u,s} \vec{\phi}(l-1) + \vec{q} \right) \ ,$$

for $\vec{\Gamma}(l)$ and then define an improved flux distribution for the next iteration using:

$$\vec{\phi}(l) = \vec{\Gamma}(l) + \omega(l)\vec{\Delta}(l) \ ,$$

where

$$\vec{\Delta}(l) = \vec{\Gamma}(l) - \vec{\phi}(l-1) \ .$$

The main problem with such a technique generally resides in computing the over-relaxation parameter $\omega(l)$. In DRAGON it is evaluated using on a variational principle derived from the transport equation.

In fact, each transport equation can be derived from a functional using the Euler-Lagrange equations. In order to simplify the reconstruction of the transport functional $\mathcal{F}[\vec{\phi}]$, we will first symmetrize Eq. (3.4):

$$\mathbf{Z}^T\mathbf{Z}\vec{\phi} - \mathbf{Z}^T\mathbf{W}\vec{q} = 0 \ , \tag{3.13}$$

where we have used:

$$\mathbf{Z} = [\mathbf{I} - \mathbf{W}\left(\Sigma_{d,s} + \Sigma_{u,s}\right)] \ .$$

The required functional is of the form:

$$\mathcal{F}[\vec{\phi}] = \frac{1}{2}\vec{\phi}^T\mathbf{Z}^T\mathbf{Z}\vec{\phi} - \vec{\phi}^T\mathbf{Z}^T\mathbf{W}\vec{q} \, . \tag{3.14}$$

The variational over-relaxation parameter $\omega(l)$, will then be chosen in such a way that $\vec{\phi}^g(l)$ minimizes the above functional, that is :

$$\frac{\partial}{\partial\omega(l)}\mathcal{F}[\vec{\phi}(l)] = 0 \ .$$

Expanding $\vec{\phi}(l)$ in terms of $\vec{\Gamma}(l)$ and $\vec{\Delta}(l)$, the derivative can be easily evaluated to:

$$\vec{\Delta}^T\mathbf{Z}^T\mathbf{Z}[\vec{\Gamma} + \omega\vec{\Delta}] + [\vec{\Gamma} + \omega\vec{\Delta}]^T\mathbf{Z}^T\mathbf{Z}\vec{\Delta} - 2\vec{\Delta}^T\mathbf{Z}^T\mathbf{W}\vec{q} = 0 \ ,$$

where it is understood that ω, $\vec{\Delta}$ and $\vec{\Gamma}$ are functions of l, the iteration number. The above equation can be solved directly for ω:

$$\omega = \frac{2\vec{\Delta}^T\mathbf{Z}^T\mathbf{W}\vec{q} - \vec{\Delta}^T\mathbf{Z}^T\mathbf{Z}\vec{\Gamma} - \vec{\Gamma}^T\mathbf{Z}^T\mathbf{Z}\vec{\Delta}}{2\vec{\Delta}^T\mathbf{Z}^T\mathbf{Z}\vec{\Delta}} \ . \tag{3.15}$$

Since the matrix $\mathbf{Z}^T\mathbf{Z}$ is symmetric, we can directly write:

$$\vec{\Gamma}^T\mathbf{Z}^T\mathbf{Z}\vec{\Delta} = \vec{\Delta}^T\mathbf{Z}^T\mathbf{Z}\vec{\Gamma} \ ,$$

and we obtain

$$\omega = \frac{[\mathbf{Z}\vec{\Delta}]^T[\mathbf{W}\vec{q} - \mathbf{Z}\vec{\Gamma}]}{[\mathbf{Z}\vec{\Delta}]^T\mathbf{Z}\vec{\Delta}} .$$

We can further simplify the above equation using the definition of $\mathbf{Z}$ to the form:

$$\omega = -\frac{[\vec{\Delta} - \mathbf{W}\vec{S}_1]^T[\vec{\Gamma} - \mathbf{W}\vec{S}_2]}{[\vec{\Delta} - \mathbf{W}\vec{S}_1]^T[\vec{\Delta} - \mathbf{W}\vec{S}_1]} ,$$

where $\vec{S}_1$ and $\vec{S}_2$ are defined as:

$$\begin{aligned}\vec{S}_1 =&(\Sigma_{d,s} + \Sigma_{u,s})\vec{\Delta} ,\\ \vec{S}_2 =&(\vec{q} + \Sigma_{d,s}\vec{\Gamma} + \Sigma_{u,s}\vec{\Gamma}) .\end{aligned}$$

Therefore, obtaining ω is equivalent to solving the transport equation twice, once for a source $\vec{S}_1$ and once for a source $\vec{S}_2$. In DRAGON, the thermal iteration procedure generally alternates between free and accelerated iterations (by default 3 free followed by 3 accelerated iterations).

3.3 Leakage Models

Two different homogeneous leakage models can be used in DRAGON when solving the transport equation in three dimensional geometries namely the B_0 and B_1 models.[35,36] Both of these models are based on a factorization of the flux in a fine structure component $\Psi^g(\vec{r},\vec{\Omega})$ and a buckling dependent distribution similar to that described in Eq. (1.11):

$$\Phi^g(\vec{r},\vec{\Omega}) \approx \Psi^g(\vec{r},\vec{\Omega})\exp(i\vec{B}\cdot\vec{r}) . \tag{3.16}$$

Using this factorization in Eq. (1.10) one obtains:

$$\begin{aligned}\vec{\Omega}\cdot\vec{\nabla}\Psi^g(\vec{r},\vec{\Omega}) + &\left[\Sigma^g(\vec{r}) + i\vec{B}\cdot\vec{\Omega}\right]\Psi^g(\vec{r},\vec{\Omega})\\ &= \sum_h \int d^2\Omega' \Sigma_s^{h\to g}(\vec{r},\vec{\Omega}'\to\vec{\Omega})\Psi^h(\vec{r},\vec{\Omega}')\\ &+ \sum_h \chi^g(\vec{r}) \int d^2\Omega' \nu\Sigma_f^h(\vec{r})\Psi^h(\vec{r},\vec{\Omega}') ,\end{aligned} \tag{3.17}$$

where we have selected k=1. In an infinite homogeneous media the scalar flux and vector current (now independent of $\vec{r}$) are related to each other according to:

$$\vec{\Omega}\Psi^g(\vec{\Omega}) = \vec{J}^g(\vec{\Omega}) = -iD^g\vec{B}\Psi^g(\vec{\Omega}) , \tag{3.18}$$

where D^g is an homogeneous diffusion coefficient. Assuming that this relation can be extended to infinite heterogeneous systems, we can then write Eq. (3.17) in the form:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi^g(\vec{r},\vec{\Omega})+\left[\Sigma^g(\vec{r})+D^gB^2\right]\Psi^g(\vec{r},\vec{\Omega})=\sum_h\int d^2\Omega'\Sigma_s^{h\to g}(\vec{r},\vec{\Omega}'\to\vec{\Omega})\Psi^h(\vec{r},\vec{\Omega}')$$
$$+\sum_h\chi^g(\vec{r})\int d^2\Omega'\nu\Sigma_f^h(\vec{r})\Psi^h(\vec{r},\vec{\Omega}')\;, \tag{3.19}$$

where D^gB^2 is known as the homogeneous leakage cross section. The problem that consists in determining the homogeneous leakage cross section for this heterogeneous problem will now be considered.

Suppose that a solution to Eq. (3.19) is known for $B^2=0$. We can then use this solution to define an equivalent infinite homogeneous problem:

$$\Sigma^g\Psi^g(\vec{\Omega})+i\vec{B}\cdot\vec{J}^g(\vec{\Omega})=\sum_h\int d^2\Omega'\Sigma_s^{h\to g}(\vec{\Omega}'\to\vec{\Omega})\Psi^h(\vec{\Omega}')$$
$$+\sum_h\chi^g\int d^2\Omega'\nu\Sigma_f^h\Psi^h(\vec{\Omega}')\;, \tag{3.20}$$

where the total cross sections is homogenized according to:

$$\Sigma^g=\frac{\int d^3r\Psi^g(\vec{r},\vec{\Omega})\Sigma^g(\vec{r})}{\int d^3r\Psi^g(\vec{r},\vec{\Omega})}\;, \tag{3.21}$$

and similar relations are used for the fission and scattering cross sections. If we can solve Eq. (3.20) for D^g and B, they can then be inserted back in Eq. (3.19) to obtain an improved solution for the heterogeneous transport problem. We can repeat this iterative process until the solution to Eqs. (3.19) and (3.20) become consistent at which point a homogeneous leakage cross section has been found that is compatible with the heterogenous transport problem.

In the above algorithm, we assumed that a solution to Eq. (3.20) could be obtained. In fact, we also assume that the solution to the homogeneous problem will be faster than the solution to the full heterogeneous problem. This can be shown to be the case when a limited series expansion for the scattering cross section is considered. In fact using:

$$\Sigma_s^{h\to g}(\vec{\Omega}'\to\vec{\Omega})=\Sigma_{s,0}^{h\to g}+3\Sigma_{s,1}^{h\to g}\vec{\Omega}\cdot\vec{\Omega}'\;,$$

Eq. (3.20) can be expanded in Legendre polynomials where only the zeroth and first order contributions do not vanish (B_1 approximation). We can then write:

$$\Sigma^g\Psi^g(\vec{\Omega})+i\vec{B}\cdot\vec{J}^g(\vec{\Omega})=\sum_h\Sigma_{s,0}^{h\to g}\int d^2\Omega'\Psi^h(\vec{\Omega}')+3\sum_h\Sigma_{s,1}^{h\to g}\int d^2\Omega'\vec{\Omega}\cdot\vec{J}^h(\vec{\Omega}')$$
$$+\sum_h\chi^g\nu\Sigma_f^h\int d^2\Omega'\Psi^h(\vec{\Omega}')\;, \tag{3.22}$$

After some manipulations followed by integrations over $d^2\Omega$ and $d^2\Omega\vec{\Omega}$, one obtains:

$$\begin{aligned}
\psi^g =& \alpha^g \sum_h (\Sigma_{s,0}^{h\rightarrow g} + \chi^g \nu \Sigma_f^h)\psi^h + 3\beta^g \sum_h \Sigma_{s,1}^{h\rightarrow g} \frac{\vec{B}\cdot\vec{j}^h}{iB^2} \ , \\
\vec{j}^g =& \beta^g \sum_h \left[(\Sigma_{s,0}^{h\rightarrow g} + \chi^g \nu \Sigma_f^h) \frac{\vec{B}\psi^h}{iB^2} + 3\Sigma^g \Sigma_{s,1}^{h\rightarrow g} \frac{\vec{j}^h}{B^2} \right] \ ,
\end{aligned}$$

where we have defined:

$$\begin{aligned}
\psi^g =& \int d^2\Omega \Psi^g(\vec{\Omega}) \ , \\
\vec{j}^g =& \int d^2\Omega \vec{\Omega}\,' \Psi^g(\vec{\Omega}) \ , \\
\alpha^g =& \frac{1}{B} \arctan\left(\frac{B}{\Sigma^g}\right) \ , \\
\beta^g =& 1 - \Sigma^g \alpha^g \ .
\end{aligned}$$

Once the solution for ψ^g and $\vec{j}^g$ and the eigenvalue B for the homogeneous problem has been obtained, we can then compute the homogeneous diffusion coefficient using:

$$D^g = \frac{i\vec{B}\cdot\vec{j}^g}{B^2\psi^g} \ .$$

Finally, in the case where $\Sigma_{s,1}^{h\rightarrow g} = 0$ (B_0 approximation), D^g takes the form:

$$D^g = \frac{\beta^g}{\alpha^g} \ .$$

Chapter 4

Condensation and Homogenization Techniques

In general lattice cell calculations are performed on heterogeneous cells using multigroup macroscopic cross sections generated from a multigroup microscopic cross-section library. On the other hand, the information that is required by finite reactor codes consist of cell average fewgroup macroscopic cross sections. Accordingly, a lattice code should be able to provide as an output fewgroup cell average macroscopic cross sections. In this chapter we will describe how the condensation (reduction in the number of groups) and homogenization (generation of macroscopic cross sections averaged over many spatial regions) of the cell properties are performed in DRAGON.

Recall that the transport equation derived in Chapter 1 is based on a statistical neutron balance that relies on the evaluation of reaction rates. In fact, for a multigroup and multiregion problem the various reaction rates defined in Eq. (1.3) can be written as:

$$R_i^g = V_i \phi_i^g \Sigma_i^g \ , \tag{4.1}$$

$$R_{s,i}^{h\to g} = V_i \phi_i^h \Sigma_{s,i}^{h\to g} \ , \tag{4.2}$$

$$R_{f,i}^g = V_i \phi_i^g \nu\Sigma_{f,i}^g \ , \tag{4.3}$$

$$R_{p,i}^g = \chi_i^g V_i \sum_h \phi_i^h \nu\Sigma_{f,i}^h \ , \tag{4.4}$$

where R_i^g represents the number of collisions per second with a neutron of energy located in the group g taking place in region i. Similarly, $R_{s,i}^{h\to g}$ is the rate at which neutrons in group h are scattered to group g. Finally $R_{f,i}^g$ and $R_{p,i}^g$ represent the number of fission reaction per second initiated by a neutron in group g and the average number of neutrons emitted in group g after a fission.

Preservation of such reaction rates therefore seems a natural and physically valid criterion for selecting a condensation or homogenization procedure. For example, in many experiments one measures the sum over all energies of a typical reaction rate in a specific region in space. In the case where the multigroup flux distribution inside the cell is known, the total reaction rate is computed using:

$$R_i = \sum_g V_i \phi_i^g \Sigma_i^g \ . \tag{4.5}$$

On the other hand if one solves an equivalent one group multiregion problem one should obtain:

$$R_i = V_i \phi_i \Sigma_i \ . \tag{4.6}$$

Accordingly, ϕ_i and Σ_i should be defined in such a way that the reaction rates are be preserved in the group condensation process.

Similarly, one may decide to compare with an experiment the rate at which collisions will take place in a cell for a specific energy group g. In the case where a multigroup and multiregion solution of the transport equation has been evaluated, this collision rate is computed according to:

$$R^g = \sum_i V_i \phi_i^g \Sigma_i^g \ , \tag{4.7}$$

while for the equivalent one region case we should obtain:

$$R^g = V \phi^g \Sigma^g \ . \tag{4.8}$$

We will then require that ϕ^g and Σ^g are defined in such a way that the reaction rates are also preserved by the spatial homogenization procedure.

4.1 Condensation Technique

Here, for simplicity, we will derive the condensation relations using the multigroup integro-differential transport equation (see Eq. (1.20)):

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \Phi^g(\vec{r}, \vec{\Omega}) + \Sigma^g(\vec{r}) \Phi^g(\vec{r}, \vec{\Omega}) = & \sum_h \int d^2\Omega' \Sigma_s^{h \to g}(\vec{r}, \vec{\Omega}' \to \vec{\Omega}) \Phi^h(\vec{r}, \vec{\Omega}') \\ & + \frac{1}{k} \sum_h \int d^2\Omega' \chi^g(\vec{r}) \nu \Sigma_f^h(\vec{r}, t) \Phi^h(\vec{r}, \vec{\Omega}') \ . \end{aligned}$$

Summing the above over all the energy groups one obtains:

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \sum_g \Phi^g(\vec{r}, \vec{\Omega}) + \sum_g \Sigma^g(\vec{r}) \Phi^g(\vec{r}, \vec{\Omega}) = & \sum_g \sum_h \int d^2\Omega' \Sigma_s^{h \to g}(\vec{r}, \vec{\Omega}' \to \vec{\Omega}) \Phi^h(\vec{r}, \vec{\Omega}') \\ & + \sum_g \frac{1}{k} \sum_h \int d^2\Omega' \chi^g(\vec{r}) \nu \Sigma_f^h(\vec{r}) \Phi^h(\vec{r}, \vec{\Omega}') \ . \end{aligned}$$

On the other hand the one group version of the same equation is:

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \Phi(\vec{r}, \vec{\Omega}) + \Sigma(\vec{r}) \Phi(\vec{r}, \vec{\Omega}) = & \int d^2\Omega' \Sigma_s(\vec{r}, \vec{\Omega}' \to \vec{\Omega}) \Phi(\vec{r}, \vec{\Omega}') \\ & + \frac{1}{k} \int d^2\Omega' \chi(\vec{r}) \nu \Sigma_f(\vec{r}) \Phi(\vec{r}, \vec{\Omega}') \ . \end{aligned}$$

One can easily see that these two equations are equivalent provided we define

$$\begin{aligned}
\Phi(\vec{r},\vec{\Omega}) =& \sum_g \Phi^g(\vec{r},\vec{\Omega}) \ , \\
\Sigma(\vec{r}) =& \frac{1}{\Phi(\vec{r},\vec{\Omega})} \sum_g \Sigma^g(\vec{r})\Phi^g(\vec{r},\vec{\Omega}) \ , \\
\Sigma_s(\vec{r},\vec{\Omega}' \rightarrow \vec{\Omega}) =& \frac{1}{\Phi(\vec{r},\vec{\Omega}')} \sum_g \sum_h \Sigma_s^{h\rightarrow g}(\vec{r},\vec{\Omega}' \rightarrow \vec{\Omega})\Phi^h(\vec{r},\vec{\Omega}') \ , \\
\chi(\vec{r}) =& \sum_g \chi^g(\vec{r}) \ , \\
\nu\Sigma_f(\vec{r}) =& \frac{1}{\Phi(\vec{r},\vec{\Omega}')} \sum_h \nu\Sigma_f^h(\vec{r})\Phi^h(\vec{r},\vec{\Omega}') \ .
\end{aligned}$$

Namely, the cross section condensation can be performed using a multigroup flux weighting. In fact, in the case where a multiregion discretization of the problem has been consider, the condensation procedure will take the form:

$$\begin{aligned}
\phi_i^K =& \sum_{g\in G_K} \phi_i^g \ , \\
\Sigma_i^K =& \frac{1}{\phi_i^K} \sum_{g\in G_K} \Sigma_i^g \phi_i^g \ , \\
\Sigma_{s,i}^{LK} =& \frac{1}{\phi_i^L} \sum_{h\in G_L} \sum_{g\in G_K} \Sigma_{s,i}^{h\rightarrow g} \phi_i^h \ , \\
\chi_i^K =& \sum_{g\in G_K} \chi_i^g \ , \\
\nu\Sigma_{f,i}^K =& \frac{1}{\phi_i^K} \sum_{g\in G_K} \nu\Sigma_{f,i}^h \phi_i^g \ ,
\end{aligned}$$

where each macrogroup K includes the set of microgroups $g \in G_K$.

4.2 Full Cell Homogenization

In the case where the cell is heterogeneous and extends to infinity (no leakage), we can write the integral transport equation in the form (Eq. (1.41) with $\phi_{-,\alpha}^g = 0$):

$$\phi_i^g = \sum_j p_{ij}^g (\sum_h \Sigma_{s,j}^{h\rightarrow g} \phi_j^h + \frac{\chi_j^g}{k} \sum_h \nu\Sigma_{f,j}^h \phi_j^h) \ .$$

Multiplying the above equation by $\Sigma_i^g V_i$ and summing over all regions i yields:

$$\sum_i \Sigma_i^g V_i \phi_i^g = \sum_h \left(\sum_i \Sigma_{s,i}^{h\to g} V_i \phi_i^h + \frac{1}{k} \sum_i \chi_i^g \nu \Sigma_{f,i}^h V_i \phi_i^h \right) ,$$

using the collision probabilities conservation relation of Eq. (1.48). This homogenized transport equation can be compared to the transport equation in an infinite homogeneous cell:

$$\hat{\Sigma}^g V \hat{\phi}^g = \sum_h \left(\hat{\Sigma}_{s,i}^{h\to g} V \hat{\phi}^h + \frac{1}{k} \hat{\chi}^g \hat{\nu} \hat{\Sigma}_f^h V \hat{\phi}^h \right) .$$

One can readily see that the homogenized and homogeneous transport equations are identical if one assumes that:

$$\begin{aligned}
\hat{\phi}^g =& \frac{1}{V} \sum_i V_i \phi_i^g , \\
\hat{\Sigma}^g =& \frac{1}{V\hat{\phi}^g} \sum_i V_i \Sigma_i^g \phi_i^g , \\
\hat{\Sigma}_s^{h\to g} =& \frac{1}{V\hat{\phi}^h} \sum_i V_i \Sigma_{s,i}^{h\to g} \phi_i^h , \\
\hat{\nu}\hat{\Sigma}_f^g =& \frac{1}{V\hat{\phi}^g} \sum_i V_i \nu \Sigma_{f,i}^g \phi_i^g , \\
\hat{\chi}^g =& \frac{1}{V \sum_h \hat{\nu}\hat{\Sigma}_f^g \hat{\phi}^g} \sum_i \chi_i^g V_i \sum_h \nu \Sigma_{f,i}^g \phi_i^h .
\end{aligned}$$

As a result, a flux-volume homogenization technique is adequate in this case.

4.3 Partial Cell Homogenization and SPH Factors

In the case where the cell resulting from the homogenization process is also heterogeneous, namely, the N initial regions are combined into M regions, each of these final regions I being composed of M_I initial regions i, then the M region heterogeneous transport equation takes the from:

$$V_I \hat{\Sigma}_I^g \hat{\phi}_I^g = \sum_J \hat{p}_{JI}^g(\hat{\Sigma}^g)(\sum_h \hat{\Sigma}_{s,J}^{h\to g} \hat{\phi}_J^h + \frac{\hat{\chi}_J^g}{k} \sum_h \hat{\nu}\hat{\Sigma}_{f,J}^h \hat{\phi}_J^h) ,$$

where $\hat{P}_{JI}^g(\hat{\Sigma}^g)$ represents the fact that the collision probabilities are computed in this case using the homogenized cross sections $\hat{\Sigma}_I^g$. The homogenized transport equation on the other hand takes the form:

$$\sum_{i\in M_I} V_i \Sigma_i^g \phi_i^g = \sum_{i\in M_I} \sum_J \sum_{j\in M_J} p_{ji}^g(\Sigma^g)(\sum_h \Sigma_{s,j}^{h\to g} \phi_j^h + \frac{\chi_J^g}{k} \sum_h \nu \Sigma_{f,j}^h \phi_j^h) .$$

The main problem here is that in order for the above two equations to be equivalent we need:

$$\sum_{i\in M_I} V_i \Sigma_i^g \phi_i^g = V_I \Sigma_I^g \phi_I^g \ ,$$

and

$$\sum_J \hat{p}_{JI}^g(\hat{\Sigma}^g)(\sum_h \hat{\Sigma}_{s,J}^{h\to g}\hat{\phi}_J^h + \frac{\hat{\chi}_J^g}{k}\sum_h \hat{\nu}\hat{\Sigma}_{f,J}^h\hat{\phi}_J^h) =$$
$$\sum_{i\in M_I}\sum_J\sum_{j\in M_J} p_{ji}^g(\Sigma^g)(\sum_h \Sigma_{s,j}^{h\to g}\phi_j^h + \frac{\chi_J^g}{k}\sum_h \nu\Sigma_{f,j}^h\phi_j^h) \ ,$$

to be simultaneously true.

Because there is no simple relation between $\hat{p}_{JI}^g(\hat{\Sigma}^g)$ and $p_{ji}^g(\Sigma^g)$, the direct flux-volume homogenization method described in the previous section is no longer adequate. The alternative here is to use a non-linear process. Assuming that the reaction rate are still conserved, we could redefine the homogeneous flux $\hat{\phi}_I^g$ and cross sections $\hat{\Sigma}_I^g$ as follows:

$$\hat{\phi}_I^g = \frac{1}{\mu_I^g}\phi_I^g \ , \tag{4.9}$$

and

$$\hat{\Sigma}_I^g = \mu_I^g \Sigma_I^g \ , \tag{4.10}$$

where the factor μ_I^g are arbitrary and a flux-volume homogenization procedure similar to that described in the previous section was used for the homogenized flux ϕ_I^g and cross sections Σ_I^g. The homogenization factors defined above are known as the SPH factors and can be determined numerically using an iterative procedure.[37]

Finally note that the same technique can be applied to full cell homogenization in the case where non-reflective boundary conditions are considered (void boundary conditions for example).

4.4 Microscopic Cross Section Homogenization

In the above analysis we assumed that only the macroscopic cross sections were to be homogenized and condensed. However in some cases one may want to perform the homogenization process over the microscopic cross sections associated with a specific isotope. According to Eq. (1.4), the macroscopic cross section associated with a material is simply the sum over all isotopes of the isotopic macroscopic cross section Σ_I namely

$$\Sigma_i^g = \sum_I \Sigma_{I,i}^g \ , \tag{4.11}$$

where we have used

$$\Sigma^g_{I,i} = N_{I,i}\sigma^g_I \ , \tag{4.12}$$

with $N_{I,i}$, the concentration of isotope I in region i.

As a result, both the homogenization and condensation procedure described above remain valid for $\Sigma^g_{I,i}$. Moreover since the final concentration of isotope I in the cell will be given by:

$$N_I = \frac{1}{V}\sum_i N_{I,i}V_i \ ,$$

we can define the equivalent homogenized microscopic cross section as:

$$\hat{\sigma}^K_I = \frac{\mu^K_I}{N_I V \phi^K}\sum_{i\in M_I}\sum_{g\in G_K} N_{I,i}V_i\sigma^g_I\phi^g_i \ ,$$

where we may note that the microscopic cross sections now become dependent on the spatial position.

Conclusion

Let us start with some comments and warning on the use of the collision probability technique method related to the approximation we discussed in Chapter 1.

The first assumption we will consider is that the sources are assumed constant inside each region. Because these sources are proportional to the flux, this implies that the flux must also be assumed constant inside each region. One way to ensure that this is the case consists in selecting the spatial discretization in such a way that the average path length crossed by neutrons inside each region is much smaller that the neutron mean free path in the region. In general, this is difficult to achieve. The problem here is that for most problems this implies that the number of regions to be considered is so large that the memory requirements exceed what is available on most computer. In other cases, even if the size of the problem to analyze can still be managed, the dimensions of some regions become so small that it is nearly impossible to ensure that each region is crossed by a sufficient number of integrations lines. However, for a large number of problems the system to analyze is not too heterogeneous, the sources in the system do not vary to strongly from region to region and the collision probability technique can still be used with a high degree of reliability even if a relatively coarse mesh is considered.

The second assumption to consider is that the angular flux must be constant and isotropic on external surfaces. In that case the solution is to try to get rid of the external surfaces with non-zero re-entrant angular flux using the symmetries of the problem. If this is not possible, the alternative is to select a model where the region of interest is far from the external surfaces.

Finally, as one may have realized, this report only gives a brief description of the theory of DRAGON. We intentionally avoided discussing many topics including the theory related to solution of 2-D problems using the collision probability, the interface current and the characteristics methods as well as the technique used for isotopic depletion calculations and in the processing of microscopic cross section libraries.

References

[1] G. Marleau, R. Roy and A. Hébert, *DRAGON: A Collision Probability Transport Code for Cell and Supercell Calculations*, Report IGE–157, Institut de génie nucléaire, École Polytechnique de Montréal, Montréal, Québec (1994).

[2] G. Marleau, A. Hébert and R. Roy, "New Computational Methods Used in the Lattice Code DRAGON", *Topical Meeting on Advances in Reactor Physics*, pp 1.177–1.188, Charleston, South Carolina, March 8–11 1992;

[3] A. Hébert, G. Marleau and R. Roy, "Application of the Lattice Code DRAGON to CANDU Analysis", *Trans. Am. Nucl. Soc.*, **72**, 335 (1995);

[4] R. Roy, A. Hébert and G. Marleau, "A Transport Method for Treating Three-Dimensional Lattices of Heterogeneous Cells", *Nucl. Sci. Eng.*, **101**, 217–225 (1989).

[5] R. Roy, G. Marleau, J. Tajmouati and D. Rozon, "Modeling of CANDU Reactivity Control Devices with the Lattice Code DRAGON", *Ann. nucl. Energy*, **21**, 115–132 (1994).

[6] G. Marleau, A. Hébert and R. Roy, *A User's Guide for DRAGON*, Report IGE-174 Rev. 5, École Polytechnique de Montréal, Institut de Génie Nucléaire (2000).

[7] A. Hébert, G. Marleau and R. Roy, *A Description of the DRAGON Data Structures*, Report IGE-232 Rev. 3, École Polytechnique de Montréal, Institut de Génie Nucléaire (2000).

[8] G. Marleau, *DRAGON Theory Manual Part 1: Collision Probability Calculations*, Report IGE-236 Rev. 1, École Polytechnique de Montréal, Institut de Génie Nucléaire (2001).

[9] B. Davison, *Neutron Transport Theory*, Oxford University Press, London (1957).

[10] G. Bell and S. Glasstone, *Neutron Reactor Theory*, Robert E. Krieger Publishing Company, Malabar, Florida (1970).

[11] R.J.J. Stamm'ler and M.J. Abbate, *Methods of Steady-State Reactor Physics in Nuclear Design*, Academic Press, London (1983).

[12] E.E. Lewis and W.F. Miller, Jr., *Computational Methods of Neutron Transport*, John Wiley and Sons, New York (1984).

[13] R. Sanchez, "Approximate Solutions of the Two-Dimensional Integral Transport Equation by Collision Probability Methods", *Nucl. Sci. Eng*, **64**, 384-404 (1977).

[14] F.E. Driggers, *A Method for Calculating Neutron Absorption and Flux Spectra at Epithermal Energies*, Report AECL-1996, Atomic Energy of Canada Limited, Chalk River, Ontario (1964).

[15] A. Kavenoky, *Calcul et utilisation des probabilité de première collision pour des milieux hétérogènes à une dimension: Les programmes ALCOLL et CORTINA*, Note CEA-N-1077, Commissariat à l'énergie atomique, Saclay, France (1969).

[16] J.R. Askew, "Review of the Status of Collision Probability Methods", *Seminar on Numerical Reactor Calculations*, 185-196, International Atomic Energy Agency, Vienna, 1972.

[17] D. Emendorfer, "Physics Assumptions and Applications of Collision Probabilities Methods", *ANS Conference on Mathematical Models and Computational Techniques for Analysis of Nuclear Systems*, Ann Arbor, Michigan, CONF-730414-P2, U.S. Atomic Energy Commission (1974).

[18] A. Kavenoky, "Status of Integral Transport Theory", *ANS/ENS International Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems*, Munich, Germany, April 27-29, 1981.

[19] R. Sanchez and N.J. McCormick, "A Review of Neutron Transport Approximations", *Nucl. Sci. Eng*, **80**, 481-535 (1982).

[20] G. Marleau and A. Hébert, "Analysis of Cluster Geometries Using the DP1 Approximation of the $J_{\pm}$ Technique", *Nucl. Sci. Eng*, **111**, 257-270 (1992).

[21] R.E. Macfarlane, *TRANSX-CTR: A code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes for Fusion Systems Analysis*, Reoprt LA-9863-MS, Los Alamos Scientific Laboratory, New Mexico (1984).

[22] J. Griffiths, *WIMS–AECL Users Manual*, Report RC-1176/COG-94-52, Atomic Energy of Canada Limited, Chalk River, Ontario (1994); see also J.V. Donnelly, *WIMS–CRNL: A User's Manual for the Chalk River Version of WIMS*, Report AECL-8955, Atomic Energy of Canada Limited, Chalk River, Ontario (1986).

[23] E.A. Villarino, R.J.J. Stamm'ler, A.A. Ferri and J.J. Casal, "HELIOS: Angularly Dependent Collision Probabilities", *Nucl. Sci. Eng.*, **112**, 16–31 (1992).

[24] *MATXS7A - 69 Neutron Group Cross Section Library in MATXS*, DLC-117, RSIC Data Library Collection, Oak Ridge National Laboratory (1985).

[25] P. Vontobel and S. Pelloni, "New JEF/EFF Based MATXS-Formatted Nuclear Data Libraries", *Nucl. Sci. Eng.*, **101**, 298 (1989).

[26] J.R. Askew, F.J. Fayers and P.B. Kemshell, "A General Description of the Lattice Code WIMS", *J. Br. Energy Soc.*, **5**, 564 (1966).

[27] M.J. Halsall and C.J. Taubman, *The '1986' WIMS Nuclear Data Library*, Report AEEW-R 2133, United Kingdom Atomic Energy Authority, Winfrith (1986)

[28] A. Hébert and G. Marleau, "Generalization of the Stamm'ler Method for the Self-Shielding of Resonant Isotopes in Arbitrary Geometries", *Nucl. Sci. Eng.*, **108**, 230 (1991).

[29] B.J. Carlson, *Tables of Equal Weight Quadrature EQ_N Over the Unit Sphere*, LA-4734, Los Alamos Scientific Laboratory (1971).

[30] R. Roy, A. Hébert and G. Marleau, "A Transport Method for Treating 3D Lattices of Heterogeneous Cells", *International Topical Meeting on Advances in Reactor Physics, Mathematics and Computation*, pp 665-679, Paris, France, April 27–30, 1987.

[31] J.V. Donnelly, *WIMS–CRNL: A User's Manual for the Chalk River Version of WIMS*, Report AECL-8955, Atomic Energy of Canada Limited, Chalk River, Ontario (1986).

[32] R. Roy and G. Marleau, "Normalization Techniques for Collision Probability Matrices", *PHYSOR-90*, pp IX.40-IX.49, Marseille, France, April 23–27, 1990.

[33] E.M. Gelbard, "Refinements in the Computation of Collision Probabilities", *International Meeting on Advances in Nuclear Engineering Computational Methods*, Knoxville, Tennessee, April 9–11, 1989

[34] G. Marleau and A. Hébert, "Solving the Multigroup Transport Equation Using the Power Iteration Method", *1985 Simulation Symposium on Reactor Dynamics and Plant Control*, Kingston, Ontario, April 22–23, 1985.

[35] A. Hébert, *Neutronique*, École Polytechnique de Montréal, Institut de Génie Nucléaire (1983).

[36] I. Petrovic and P. Benoist, "B_N Theory: Advances and New Models for Neutron Leakage Calculations", *Advances in Nuclear Science and Technology*, **24**, (1996).

[37] A. Hébert, "A Consistent Technique for Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly", *Nucl. Sci. Eng.*, **113**, 227-238 (1993); see also A. Hébert and G. Mathonnière, "Development of a Third-Generation Superhomogénéisation Method for the Homogenization of a Pressurized Water Reactor Assembly", *Nucl. Sci. Eng.*, **115**, 129-141 (1993).