HIGHER-ORDER DGFEM TRANSPORT CALCULATIONS ON POLYTOPE MESHES FOR MASSIVELY-PARALLEL ARCHITECTURES

A Dissertation

by

MICHAEL WAYNE HACKEMACK

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jean Ragusa
Committee Members,	Marvin Adams
	Jim Morel
	Nancy Amato
	Troy Becker
Head of Department,	Yassin Hassan

August 2016

Major Subject: Nuclear Engineering

1. THE DGFEM FORMULATION OF THE MULTIGROUP S_N EQUATIONS

The movement of bulk materials and particles through some medium can be described with the statistical behavior of a non-equilibrium system. Boltzmann first devised these probabilistic field equations to characterize fluid flow via driving temperature gradients [1]. His work was later extended to model general fluid flow, heat conduction, hamiltonian mechanics, quantum theory, general relativity, and radiation transport, among others. The Boltzmann Equation can be written in the general form:

$$\frac{\partial u}{\partial t} = \left(\frac{\partial u}{\partial t}\right)_{force} + \left(\frac{\partial u}{\partial t}\right)_{advec} + \left(\frac{\partial u}{\partial t}\right)_{coll} \tag{1.1}$$

where $u(\vec{r}, \vec{p}, t)$ is the transport distribution function parameterized in terms of position, $\vec{r} = (x, y, z)$, momentum, $\vec{p} = (p_x, p_y, p_z)$, and time, t. In simplified terms, Eq. (1.1) can be interpreted that the time rate of the change of the distribution function, $\frac{\partial u}{\partial t}$, is equal to the sum of the change rates due to external forces, $\left(\frac{\partial u}{\partial t}\right)_{force}$, advection of the particles, $\left(\frac{\partial u}{\partial t}\right)_{advec}$, and particle-to-particle and particle-to-matter collisions, $\left(\frac{\partial u}{\partial t}\right)_{coll}$ [2].

For neutral particle transport, the following assumptions [3] about the behavior of the radiation particles can be utilized:

1. Particles may be considered as points;

2. Particles do not interact with other particles;

3. Particles interact with material target atoms in a binary manner;

4. Collisions between particles and material target atoms are instantaneous;

5. Particles do not experience any external force fields (*e.g.* gravity).

These assumptions lead to the first order form of the Boltzmann Transport Equation, which we simply call the transport equation for brevity. The remainder of the chapter is outlined as follows. Section 1.1 provides the general form of the neutron transport equation with some variants. Section 1.2 describes how we discretize the transport equation in energy with the multigroup methodology and Section 1.3 presents the angular discretization via collocation. Section 1.4 details which boundary conditions will be employed for our work. Section 1.5 will conclude our discretization procedures in the spatial domain. Section 1.6 will present the iterative procedures used to converge our solution space. We then present concluding remarks for the chapter in Section 1.7.

1.1 The Neutron Transport Equation

The time-dependent neutron angular flux, $\Psi(\vec{r}, E, \vec{\Omega}, t)$, at spatial position $\vec{r}$, with energy E moving in direction $\vec{\Omega}$ and at time t, is defined within an open, convex spatial domain $\mathcal{D}$, with boundary, $\partial\mathcal{D}$, by the general neutron transport equation:

$$\begin{aligned}\frac{\partial \Psi}{\partial t} + \vec{\Omega}\cdot\vec{\nabla}\Psi(\vec{r}, E, \vec{\Omega}, t) + \sigma_t(\vec{r}, E, t)\Psi(\vec{r}, E, \vec{\Omega}, t) &= Q_{ext}(\vec{r}, E, \vec{\Omega}, t) \\ &+ \frac{\chi(\vec{r}, E, t)}{4\pi}\int dE' \nu\sigma_f(\vec{r}, E', t)\int d\Omega' \Psi(\vec{r}, E', \vec{\Omega}', t) \\ &+ \int dE' \int d\Omega' \sigma_s(E' \rightarrow E, \Omega' \rightarrow \Omega)\Psi(\vec{r}, E', \vec{\Omega}')\end{aligned} \tag{1.2}$$

with the following, general boundary condition:

$$\Psi(\vec{r},E,\vec{\Omega},t) = \Psi^{inc}(\vec{r},E,\vec{\Omega},t) + \int dE' \int d\Omega' \gamma(\vec{r},E' \to E,\vec{\Omega}' \to \vec{\Omega},t)\Psi(\vec{r},E',\vec{\Omega}',t) \\ \text{for } \vec{r} \in \partial\mathcal{D}^- \left\{\partial\mathcal{D}, \vec{\Omega}' \cdot \vec{n} < 0\right\}. \tag{1.3}$$

In Eqs. (1.2) and (1.3), the physical properties of the system are defined as the following: $\sigma_t(\vec{r},E,t)$ is the total neutron cross section, $\chi(\vec{r},E,t)$ is the neutron fission spectrum, $\sigma_f(\vec{r},E',t)$ is the fission cross section, $\nu(\vec{r},E',t)$ is the average number of neutroncs emitted per fission, $\sigma_s(E' \to E,\Omega' \to \Omega,t)$ is the scattering cross section, $Q_{ext}(\vec{r},E,\vec{\Omega},t)$ is a distributed external source, $\Psi^{inc}(\vec{r},E,\vec{\Omega},t)$ is the incident boundary source, and $\gamma(\vec{r},E' \to E,\vec{\Omega}' \to \vec{\Omega},t)\Psi(\vec{r},E',\vec{\Omega}',t)$ is the boundary albedo.

We can simplify Eq. (1.2) to:

$$\frac{\partial\Psi}{\partial t} + \mathbf{L}\Psi = \mathbf{F}\Psi + \mathbf{S}\Psi + \mathbf{Q}, \tag{1.4}$$

by dropping the dependent variable parameters and using the following operators:

$$\begin{aligned} \mathbf{L}\Psi &= \vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{r},E,\vec{\Omega},t) + \sigma_t(\vec{r},E,t)\Psi(\vec{r},E,\vec{\Omega},t), \\ \mathbf{F}\Psi &= \frac{\chi(\vec{r},E,t)}{4\pi} \int dE' \nu\sigma_f(\vec{r},E',t) \int d\Omega' \Psi(\vec{r},E',\vec{\Omega}',t), \\ \mathbf{S}\Psi &= \int dE' \int d\Omega' \sigma_s(E' \to E,\Omega' \to \Omega,t)\Psi(\vec{r},E',\vec{\Omega}',t), \\ \mathbf{Q} &= Q_{ext}(\vec{r},E,\vec{\Omega},t), \end{aligned} \tag{1.5}$$

where **L** is the loss operator which includes total reaction and streaming, **F** is the fission operator, and **S** is the scattering operator. If we wish to analyze a transport problem at steady-state conditions, we simply drop the temporal derivative to form

$$\mathbf{L}\Psi = \mathbf{F}\Psi + \mathbf{S}\Psi + \mathbf{Q}, \tag{1.6}$$

and note that the operators of Eq. (1.5) no longer depend on time, t.

There is a special subset of transport problems that is routinely analyzed to determine the neutron behavior of a fissile system called the *k-eigenvalue problem*. In Eq. (1.2), $\nu(\vec{r}, E)$ acts as a multiplicative factor on the number of neutrons emitted per fission event. We replace this multiplicative factor in the following manner:

$$\nu(\vec{r}, E) \rightarrow \frac{\tilde{\nu}(\vec{r}, E)}{k}, \tag{1.7}$$

where we have introduced the eigenvalue, k. By also dropping the external source term, the steady-state neutron transport equation in Eq. (1.6) can be rewritten into

$$(\mathbf{L} - \mathbf{S})\,\tilde{\Psi} = \frac{1}{k}\mathbf{F}\tilde{\Psi}, \tag{1.8}$$

where $(k, \tilde{\Psi})$ forms an appropriate eigenvalue-eigenvector pair. Of most interest is the eigenpair corresponding to the eigenvalue of largest magnitude.

We can then gain knowledge of the behavior of the neutron population in the problem by taking the full phase-space integrals of the loss operator $\int\int\int\ \mathbf{L}\tilde{\Psi}\, dE\, d\Omega\, d\vec{r}$, the fission operator $\int\int\int\ \mathbf{F}\tilde{\Psi}\, dE\, d\Omega\, d\vec{r}$, and the scattering operator $\int\int\int\ \mathbf{S}\tilde{\Psi}\, dE\, d\Omega\, d\vec{r}$. With the appropriate eigenvector solution, $\tilde{\Psi}$, the k eigenvalue then has the meaning as the multiplicative value which balances Eq. (1.8) in an integral sense. This means that k also has a physical meaning as well. A value $k < 1$ is called subcritical and corresponds to a system whose neutron population decreases in time; a value $k = 1$ is called critical and corresponds to a system whose neutron population remains constant in time; and a value $k > 1$ is called supercritical and corresponds to a system whose neutron population increases in time [4].

1.2 Energy Discretization

We begin our discretization procedures by focusing on the angular flux's energy variable. An ubiquitous energy discretization procedure in the transport community is the multigroup method [5, 6]. The multigroup method is defined by splitting the angular flux solution into G number of distinct, contiguous, and non-overlapping energy intervals called groups. We begin by restricting the full energy domain, $[0, \infty)$, into a finite domain, $E \in [E_G, E_0]$. E_0 corresponds to some maximum energy value and E_G corresponds to some minimum energy value (typically 0). We have done this by defining $G + 1$ discrete energy values that are in a monotonically continuous reverse order: $E_G < E_{G-1} < \ldots < E_1 < E_0$.

From this distribution of energy values, we then say that a particular energy group, g, corresponds to the following energy interval:

$$\Delta E_g \in [E_g, E_{g-1}]. \tag{1.9}$$

Figure 1.1 provides a visual representation between the $G + 1$ discrete energy values and the G energy groups. While the order that we have prescribed may seem illogical (high-to-low) to those outside of the radiation physics community, it has been historically applied this way because radiation transport problems are iteratively solved from high energy to low energy. If the group structure is well chosen, then the transport solution can be more efficiently and easily obtained.

For the remainder of this energy discretization procedure, we will utilize the steady-state form of the transport equation in Eq. (1.6). The time-dependent and eigenvalue forms are analagous and would be derived identically. Taking the energy interval for group g as defined in Eq. (1.9), the energy-integrated angular flux of group g is

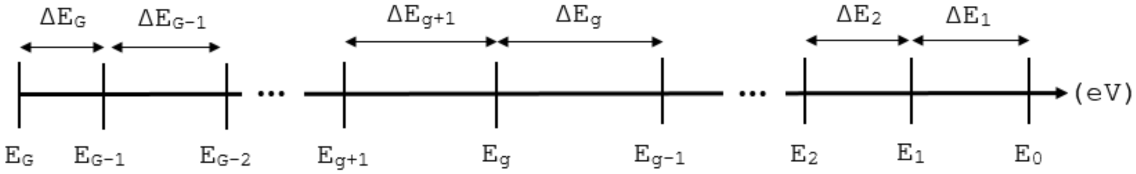


Figure 1.1: Interval structure of the multigroup methodology.

$$\Psi_g(\vec{r},\vec{\Omega}) = \int_{E_g}^{E_{g-1}} \Psi(\vec{r},E,\vec{\Omega})\,dE. \tag{1.10}$$

We can then use the energy-integrated angular flux to form the following coupled, $(g = 1, ..., G)$, discrete equations (we have dropped the spatial parameter and some of the angular parameters for further clarity):

$$\left(\vec{\Omega}\cdot\vec{\nabla} + \sigma_{t,g}\right)\Psi_g = \sum_{g'=1}^{G}\left[\frac{\chi_g}{4\pi}\nu\sigma_{f,g'}\int_{4\pi}\Psi_{g'}(\vec{\Omega}')\,d\Omega' + \int_{4\pi}\sigma_s^{g'\to g}(\vec{\Omega}',\vec{\Omega})\Psi_{g'}(\vec{\Omega}')\,d\Omega'\right] + Q_g \tag{1.11}$$

where

$$\begin{aligned}
\sigma_{t,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}}\sigma_t(\vec{r},\vec{\Omega},E)\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}{\int_{E_g}^{E_{g-1}}\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}\\
\nu\sigma_{f,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}}\nu\sigma_f(\vec{r},E)\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}{\int_{E_g}^{E_{g-1}}\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}\\
\chi_g &\equiv \int_{E_g}^{E_{g-1}}\chi(\vec{r},E)\,dE\\
\sigma_s^{g'\to g}(\vec{r},\vec{\Omega}',\vec{\Omega}) &\equiv \frac{\int_{E_{g'}}^{E_{g'-1}}\left[\int_{E_g}^{E_{g-1}}\sigma_s(\vec{r},E'\to E,\vec{\Omega}',\vec{\Omega})\,dE\right]\Psi(\vec{r},\vec{\Omega}',E')\,dE'}{\int_{E_g}^{E_{g-1}}\Psi(\vec{r},\vec{\Omega},E)\,dE}\\
Q_g(\vec{r},\vec{\Omega}) &\equiv \int_{E_g}^{E_{g-1}}Q(\vec{r},\vec{\Omega},E)\,dE
\end{aligned} \tag{1.12}$$

The above equations are mathematically exact to those presented in Eqs. (1.2 - 1.6) and we have made no approximations at this time. However, this requires full knowledge of the energy distribution of the angular flux solution at all positions in our problem domain since we weight the multigroup cross sections with this solution. This is obviously impossible since the energy distibution is part of the solution space we are trying to solve for. Instead, we now define the process to make the multigroup discretization an effective approximation method.

We first define an approximate angular flux distribution for a region s:

$$\Psi(\vec{r},\vec{\Omega},E) = \hat{\Psi}(\vec{r},\vec{\Omega}) f_s(E), \tag{1.13}$$

which is a factorization of the angular flux solution into a region-dependent energy function, $f_s(E)$, and a spatially/angularly dependent function, $\hat{\Psi}(\vec{r},\vec{\Omega})$. With this approximation, we can redefine the energy-collapsed cross sections of Eq. (1.12):

$$\begin{aligned}
\sigma_{t,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}} \sigma_t(\vec{r},E) f_s(E)\, dE}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}, \\
\nu\sigma_{f,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}} \nu\sigma_f(\vec{r},E) f_s(E)\, dE}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}, \\
\sigma_s^{g'\to g}(\vec{r},\vec{\Omega}',\vec{\Omega}) &\equiv \frac{\int_{E_{g'}}^{E_{g'-1}} \left[\int_{E_g}^{E_{g-1}} \sigma_s(\vec{r},E'\to E,\vec{\Omega}',\vec{\Omega})\, dE\right] f_s(E')\, dE'}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}.
\end{aligned} \tag{1.14}$$

It is noted that we do not need to redefine the fission spectrum or the distributed external sources since they are not weighted with the angular flux solution. With this energy factorization, we would expect, in general, that the approximation error will tend to zero as the number of discrete energy groups increases (thereby making the energy bins thinner). This is especially true if the group structure is chosen with many more bins in energy regions with large variations in the energy solution. For certain problems, the region-dependent energy function is well understood (*i.e.* almost exactly known). This means, that for these problems, we can achieve reasonable solution accuracy with only a few groups where the energy bins of the multigroup discretization are well chosen.

1.3 Angular Discretization

Now that we have provided the discretization of the energy variable, we next focus on the discretization of the transport problem in angle. We will do this in two stages: 1) expand the scattering source and the distributed external source in spherical harmonics and 2) collocate the angular flux at the interpolation points of the angular trial space. We will perform these discretization procedures by taking the steady-state equation presented in Eq. (1.6), dropping spatial parameterization, combining the fission and external sources into a single term, and using only 1 energy group:

$$\vec{\Omega} \cdot \vec{\nabla} \Psi(\vec{\Omega}) + \sigma_t \Psi(\vec{\Omega}) = \int_{4\pi} d\Omega' \, \sigma_s(\vec{\Omega}' \to \vec{\Omega}) \Psi(\vec{\Omega}') + Q(\vec{\Omega}). \tag{1.15}$$

We first develop an approximation for the scattering term in Eq. (1.15) by expanding the angular flux and the scattering cross section in spherical harmonics functions and Legendre polynomials, respectively. We begin by first assuming that the material is isotropic in relation to the radiation's initial direction. From this assumption, the parameterization of the scattering cross section can be written in terms of only the scattering angle, μ_0,

$$\sigma_s(\vec{\Omega}' \to \vec{\Omega}) = \frac{1}{2\pi} \sigma_s(\vec{\Omega}' \cdot \vec{\Omega}) = \frac{1}{2\pi} \sigma_s(\mu_0), \tag{1.16}$$

where $\mu_0 \equiv \vec{\Omega}' \cdot \vec{\Omega}$. With this simplification, the scattering cross section can now be expanded in an infinite series in terms of the Legendre polynomials,

$$\sigma_s(\vec{\Omega}' \rightarrow \vec{\Omega}) = \sum_{p=0}^{\infty} \frac{2p+1}{4\pi} \sigma_{s,p} P_p(\mu_0), \tag{1.17}$$

where $\sigma_{s,p}$ is the p angular moment of the scattering cross section. These angular moments of the scattering cross section have the form:

$$\sigma_{s,p} \equiv \int_{-1}^{1} d\mu_0 \, \sigma_s(\mu_0) P_p(\mu_0). \tag{1.18}$$

With the scattering cross section redefined, we can now expand the angular flux in terms of an infinite series of the spherical harmonics functions, Y,

$$\Psi(\vec{\Omega}) = \frac{1}{4\pi} \sum_{k=0}^{\infty} \sum_{n=-k}^{k} \Phi_{k,n} Y_{k,n}(\vec{\Omega}) \tag{1.19}$$

where the angular moments of the angular flux, $\Phi_{k,n}$, have the form:

$$\Phi_{k,n} \equiv \int_{4\pi} d\Omega \, \Psi(\vec{\Omega}) \, Y_{k,n}(\vec{\Omega}). \tag{1.20}$$

We note that the p and k orders of the scattering cross section and angular flux expansions, respectively, are not corresponding. We then take the scattering cross section expansion of Eq. (1.17) and the angular flux expansion of Eq. (1.19), and insert them into the original scattering term of the right-hand-side of Eq. (1.15). After significant algebra and manipulations, which we will not include here for brevity, the scattering term can be greatly simplified (the full details of this are located in Appendix **??**). Eq. (1.15) can now be written again with this alternate and simplified scattering term that is composed of the cross section and angular flux moments:

$$\vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{\Omega}) + \sigma_t\Psi(\vec{\Omega}) = \sum_{p=0}^{\infty} \frac{2p+1}{4\pi}\sigma_{s,p} \sum_{n=-p}^{p} \Phi_{p,n}Y_{p,n}(\vec{\Omega}) + Q(\vec{\Omega}). \tag{1.21}$$

From the initial assumption of material isotropy (which may or may not be an approximation), the scattering term of Eq. (1.21) has introduced no approximation. Unfortunately, this form requires an infinite series expansion which we cannot use with only finite computational resources. Instead, we truncate the series at some maximum expansion order, N_p, which, in general, introduces an approximate form for the scattering. However, we note that if the problem scattering can be exactly captured with moments through order N_p, then we have introduced no approximation with this truncation. With this order of truncation, we again write the angularly continuous Eq. (1.21), but also fold the source term into the spherical harmonics expansion,

$$\vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{\Omega}) + \sigma_t\Psi(\vec{\Omega}) = \sum_{p=0}^{N_p} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}) \left[\sigma_{s,p}\Phi_{p,n} + Q_{p,n}\right]. \tag{1.22}$$

At this point, one may wonder why we have altered the scattering operator so that it is terms of moments of the scattering cross sections and the angular flux. The reason is two-fold which will also be discussed in further detail later in this chapter. First, it greatly reduces the phase space of the scattering cross sections. With proper preprocessing, the scattering cross sections can be simplified into just their Legendre moments, instead of having to store angle-to-angle quantities ($\vec{\Omega}' \to \vec{\Omega}$). For every group-to-group combination in energy ($g' \to g$), there are only the N_p moments of the scattering cross section. Secondly, the contribution of the angular flux into the

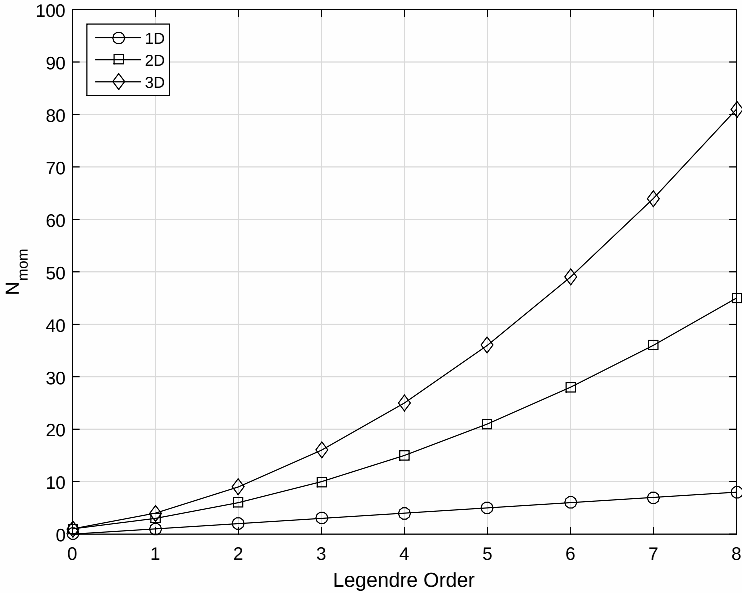


Figure 1.2: Number of spherical harmonics moments, N_{mom}, in 1D, 2D, and 3D as a function of the expansion order, p.

scattering source with its moments can also greatly reduce the dimensional space that needs to be stored in computer memory. This will be discussed later in further detail in Section 1.6.1, but it simply means that we only have to store N_{mom} angular flux moments for use in the scattering source. In 1 dimension, N_{mom} is equal to $(N_p + 1)$. In 2 dimensions, N_{mom} is equal to $\frac{(N_p+1)(N_p+2)}{2}$. In 3 dimensions, N_{mom} is equal to $(N_p + 1)^2$. For comparative purposes, we have plotted N_{mom} for 1, 2, and 3 dimensions up to order 8 in Figure 1.2.

Up to this point, we have only presented the methodology to express our source terms with expansions of the spherical harmonics functions. Next, we describe the second portion of our angular discretization by deriving the standard S_N equations using a collocation technique. We begin by choosing a set of M distinct points and weights to form a quadrature set in angular space: $\{\vec{\Omega}_m, w_m\}_{m=1}^{M}$. We will give further details about the required characteristics of this quadrature set as well as a couple of common options a little later. Using this quadrature set, we can further define a trial space for the angular flux,

$$\Psi(\vec{\Omega}) = \sum_{m=1}^{M} B_m(\vec{\Omega}) \Psi_m, \tag{1.23}$$

where the angular bases, B_m, satisfy the *Kronecker* property,

$$B_j(\vec{\Omega}_m) = \delta_{j,m}, \tag{1.24}$$

as well as the *Lagrange* property,

$$\sum_{m=1}^{M} B_m(\vec{\Omega}_m) = 1, \tag{1.25}$$

and the singular value of the angular flux along a given direction has the following notation:

$$\Psi_m = \Psi(\vec{\Omega}_m). \tag{1.26}$$

Next, we substitute Eq. (1.23) into Eq. (1.22), drop the external source for clarity, and collocate at the $(k = 1, ..., M)$ interpolation (quadrature) points,

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \left(\sum_{m=1}^{M} B_m(\vec{\Omega}_k) \Psi_m \right) + \sigma_t \left(\sum_{m=1}^{M} B_m(\vec{\Omega}_k) \Psi_m \right) \\ = \sum_{p=0}^{N_p} \frac{2p+1}{4\pi} \sigma_{s,p} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}) \left(\sum_{m=1}^{M} \Psi_m \int_{4\pi} d\Omega \, B_m(\vec{\Omega}_k) \, Y_{p,n}(\vec{\Omega}) \right), \\ k = 1, ..., M \end{aligned} \tag{1.27}$$

where we inserted Eq. (1.23) into Eq. (1.20) to form a slightly modified form for the angular flux moments:

$$\Phi_{p,n} = \sum_{m=1}^{M} \Psi_m \int_{4\pi} d\Omega\, B_m(\vec{\Omega})\, Y_{p,n}(\vec{\Omega}). \tag{1.28}$$

The *Kronecker* property of Eq. (1.24), is then used at the collocation points so that Eq. (1.27) can be simplified into the following form (where we have reintroduced the distributed source term in terms of its contribution for angle m):

$$\begin{array}{c} \vec{\Omega}_m \cdot \vec{\nabla}\Psi_m + \sigma_t \Psi_m = \displaystyle\sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sigma_{s,p} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m)\Phi_{p,n} + Q_m \\ . \\ m = 1, ..., M \end{array} \tag{1.29}$$

Equation (1.29) represents the transport equation that has been discretized into M separate equations in angle (for 1 energy group and no spatial discretization). Up to this point, we have simply stated that there is some angular quadrature set composed of M directions and weights that will satisfy some conditions of the solution, but we have not explicitly stated these conditions. For this work we will require our angular quadrature set to maintain the following properties:

1. The weights can sum to 1 by some normalization procedure: $\sum_m w_m = 1$.

2. The odd angular moments sum to $\vec{0}$: $\sum_m w_m \left(\vec{\Omega}_m\right)^n = \vec{0}$ $(n = 1, 3, 5, ...)$.

3. $\sum_m w_m \vec{\Omega}_m \vec{\Omega}_m = \frac{1}{3}\mathbb{I}$, where $\mathbb{I}$ is the identity tensor.

4. The points and weights are symmetric about the primary axes in angular space.

5. The points and weights also need to have symmetry about the problem domain boundary (this is not an issue if the domain is a rectangle in 2D or an orthogonal parallelepiped in 3D). This point is important for reflecting boundary conditions and is described in greater detail in Section 1.4.

Point 4 requires some additional explanation. In 1 dimension, this corresponds to symmetry about the point 0 on the interval $[-1, 1]$. In 2 dimensions, this corresponds to quadrant-to-quadrant symmetry about the x-y primary axes of the unit circle. In 3 dimensions, this corresponds to octant-to-octant symmetry about the x-y-z primary axes of the unit sphere.

From these properties, especially property 4, our 2D and 3D quadrature sets can be constructed in a simple and consistent manner (1D quadrature sets have different construction and our work does not include them). For both 2D and 3D problems, we can generate a subset of the quadrature points and weights on a single octant of the unit sphere, where each quadrature point in this subset has the form: $\vec{\Omega} = [\mu, \eta, \xi]$. If we are solving a 2D problem, we would then project the quadrature points onto the $(0 < \theta < \frac{\pi}{2})$ portion of the unit circle so that they have the form: $\vec{\Omega} = [\mu, \eta]$ (we can then view the primary octant as the primary quadrant). Once we have defined the quadrature points and weights for the primary quadrant or octant, we can then directly calculate the remainder of the quadrature set by mapping to the other quadrants or octants. Table 1.1 presents the mapping from the primary quadrant to the other 3 quadrants for 2D problems. Table 1.2 presents the mapping from the primary octant to the other 7 octants for 3D problems. In these tables, the '1' subscript corresponds to those angles generated in the primary quadrant or octant.

Table 1.1: 2D angle mapping from the first quadrant into the other 3 quadrants.

Quadrant	μ	η
1	$\mu_1 = \mu_1$	$\eta_1 = \eta_1$
2	$\mu_2 = -\mu_1$	$\eta_2 = \eta_1$
3	$\mu_3 = -\mu_1$	$\eta_3 = -\eta_1$
4	$\mu_4 = \mu_1$	$\eta_4 = -\eta_1$

Table 1.2: 3D angle mapping from the first octant into the other 7 octants.

Octant	μ	η	ξ
1	$\mu_1 = \mu_1$	$\eta_1 = \eta_1$	$\xi_1 = \xi_1$
2	$\mu_2 = -\mu_1$	$\eta_2 = \eta_1$	$\xi_2 = \xi_1$
3	$\mu_3 = -\mu_1$	$\eta_3 = -\eta_1$	$\xi_3 = \xi_1$
4	$\mu_4 = \mu_1$	$\eta_4 = -\eta_1$	$\xi_4 = \xi_1$
5	$\mu_5 = \mu_1$	$\eta_5 = \eta_1$	$\xi_5 = -\xi_1$
6	$\mu_6 = -\mu_1$	$\eta_6 = \eta_1$	$\xi_6 = -\xi_1$
7	$\mu_7 = -\mu_1$	$\eta_7 = -\eta_1$	$\xi_7 = -\xi_1$
8	$\mu_8 = \mu_1$	$\eta_8 = -\eta_1$	$\xi_8 = -\xi_1$

We conclude our discussion of angular discretizations by presenting two common angular quadrature sets that will be employed in this dissertation work. Section 1.3.1 presents the Level Symmetric (LS) quadrature set and Section 1.3.2 presents the Product Gauss-Legendre-Chebyshev (PGLC) quadrature set. Both of these quadrature sets can be formed from the procedure outlined before: form the primary octant and then map appropriately.

1.3.1 Level Symmetric Quadrature Set

The first quadrature set we present is the common Level Symmetric set that has had extensive use in the radiation transport community [6, 7]. Its defining characteristic is the restriction that it is rotationally symmetric (invariant) about all three axes of the primary octant. This leads to a two-fold additional set of restrictions: 1) once the location of the first ordinate is selected, then all other ordinates are known; and 2) the weights can become negative. This negativity of the weights can be problematic and lead to unphysical solutions if the angular flux is not sufficiently smooth.

We begin our description of the LS quadrature by analyzing the 3D angular coordinate system for a particular direction, $\vec{\Omega}$, as depicted in Figure 1.3. The

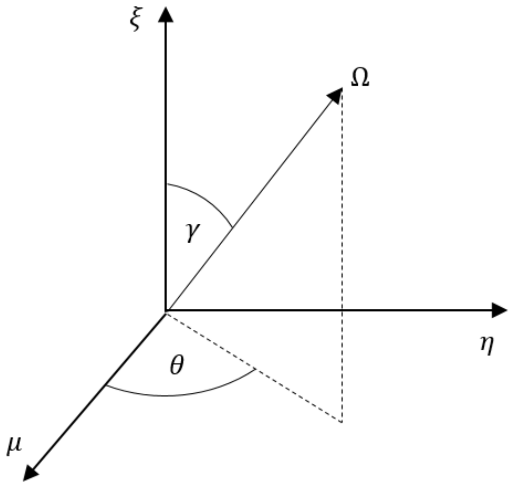


Figure 1.3: Angular coordinate system for the direction $\vec{\Omega}$.

angular direction, $\vec{\Omega} = [\vec{\Omega}_x, \vec{\Omega}_y, \vec{\Omega}_z]$, is typically described with its directional cosines: μ, η, and ξ. These are described by the angles θ and γ of the coordinate system, which are the azimuthal and polar angles, respectively, and allow us to give a functional form for each direction component:

$$\begin{aligned} \vec{\Omega}_x =& \mu = \cos(\theta)\sin(\gamma) = \cos(\theta)\sqrt{1-\xi^2} \\ \vec{\Omega}_y =& \eta = \sin(\theta)\sin(\gamma) = \sin(\theta)\sqrt{1-\xi^2} \; . \\ \vec{\Omega}_z =& \xi = \cos(\gamma) \end{aligned} \tag{1.30}$$

The direction cosines are related and necessarily must have a Euclidean norm of 1:

$$\mu^2 + \eta^2 + \xi^2 = 1. \tag{1.31}$$

We next specify the order of the quadrature set, N, which we restrict to only positive even integers. Each direction cosine (μ, η, and ξ) then contains exactly $N/2$ positive values with respect to each of the three axes. This leads to exactly $\frac{N(N+2)}{8}$ total angular directions in the primary octant. Because of the rotational invariance of the quadrature set, no ordinate axis receives preferential clustering of the nodes. This means that the index value of each ordinate is identical:

$$\mu_i = \eta_i = \xi_i, \qquad i \in (1, N/2) \tag{1.32}$$

As previously stated, once the location of the first ordinate, μ_1, is selected, then the remaining are directly known. However, to maintain the relation of Eq. (1.31), this first ordinate has restrictions placed on it. It must maintain a positive value: $\mu_1^2 \in (0, 1/3]$. Also, for the $S2$ set (N=2), there is exactly one direction cosine with no degrees of freedom. This requires that $\mu_1^2 = 1/3$ for the $S2$ case.

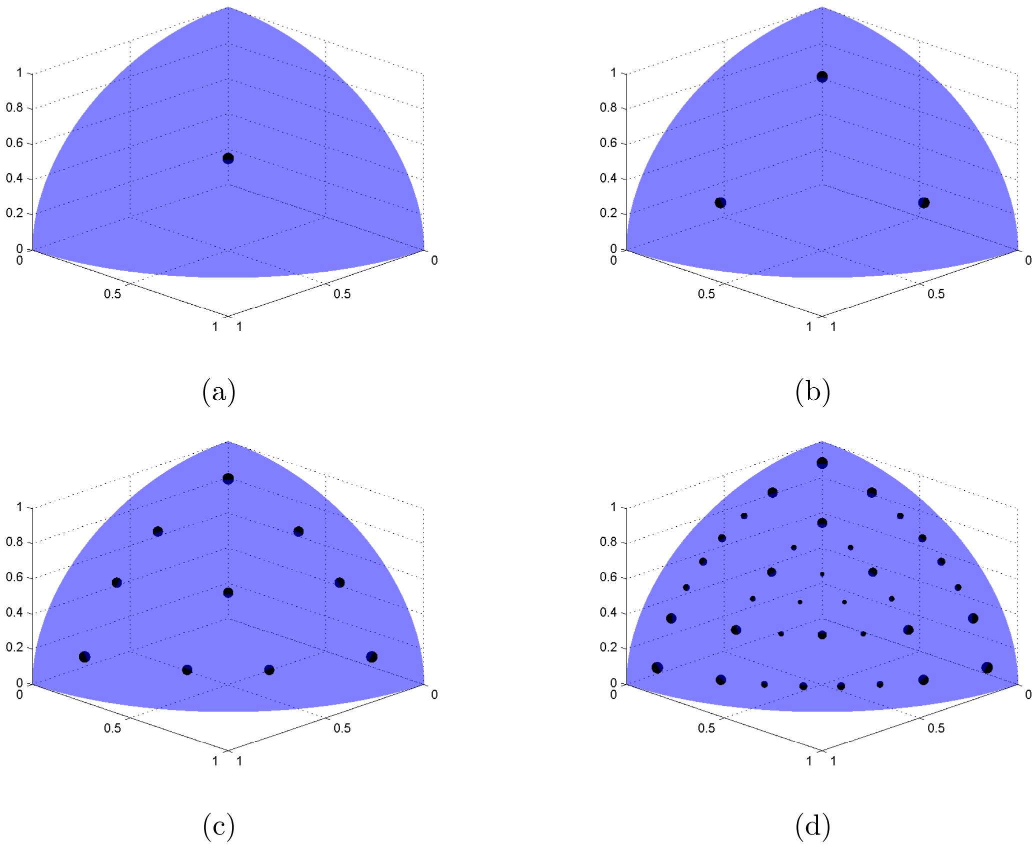
With μ_1 now selected, we can consider an ordinate set $[\mu_i, \eta_j, \xi_k]$, where $i+j+k = N/2 + 2$. To maintain the appropriate Euclidean norm, a recursion relation can be derived (which we will not do for brevity):

$$\mu_i^2 = \mu_{i-1}^2 + \Delta \tag{1.33}$$

where the spacing constant, Δ, has the form:

$$\Delta = \frac{2(1 - 3\mu_1^2)}{N - 2}. \tag{1.34}$$

Based on this recursion form, we can see that if μ_1^2 is close to 0, then the ordinates will be clustered around the poles of the primary octant. Likewise, if μ_1^2 is close to 1/3, then the ordinates will be clustered away from the poles. For this work, we choose to select values of μ_1 in conformance with the LQ_N quadrature set because they can exactly integrate the polynomials of the Legendre expansion of the scattering cross sections [8]. We finally note that the weights of the LQ_N set become negative for $N \geq 20$.



(a)

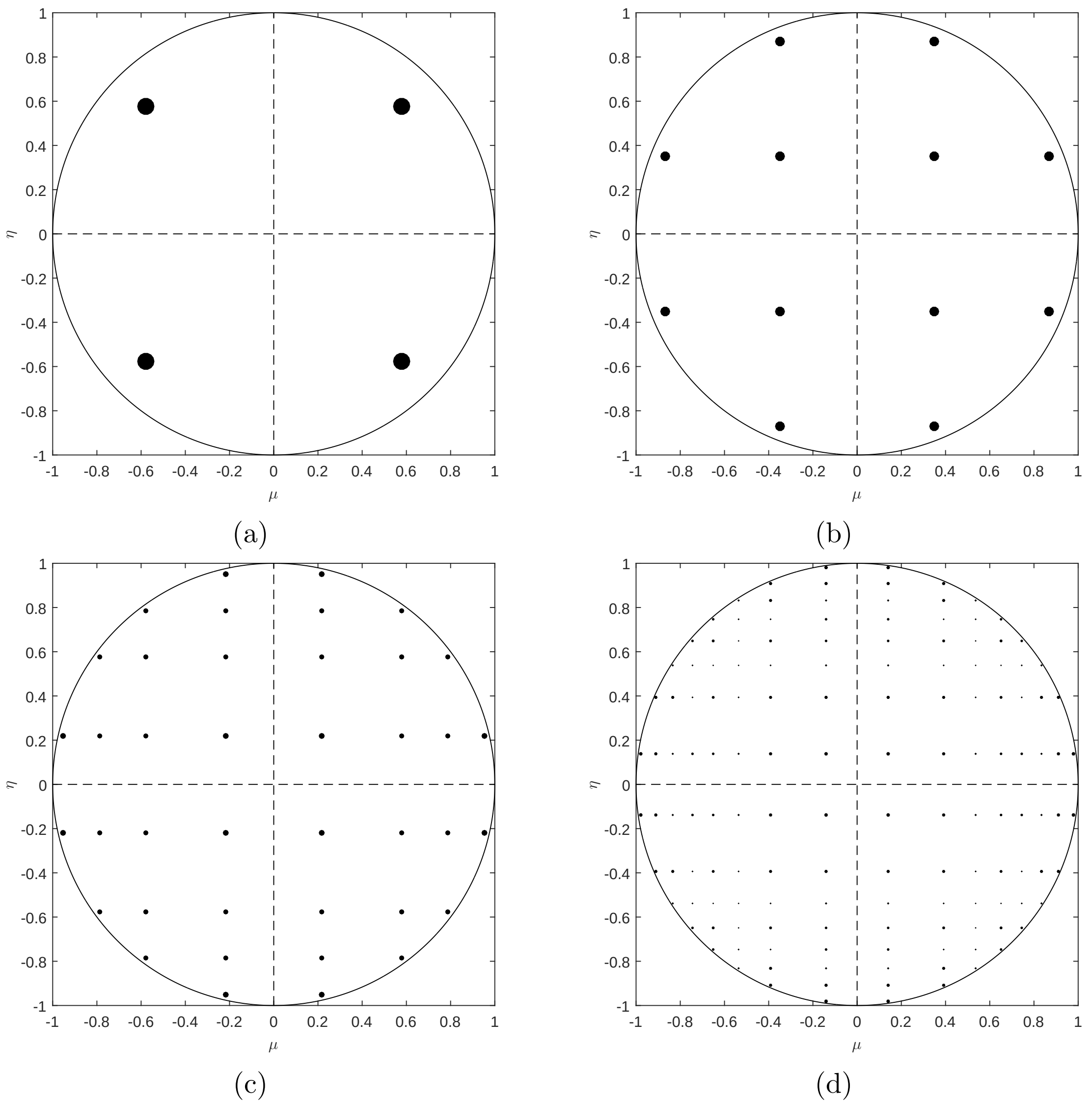
(b)

(c)

(d)

Figure 1.4: Level-Symmetric angular quadrature sets of order (a) 2, (b) 4, (c) 8, and (d) 16.

We conclude this discussion of the LS quadrature set with some examples. Figure 1.4 provides a visual depiction of the LS nodes and weights in the primary octant for varying orders. The magnitude of the weights is characterized by the relative size of the nodes. Figure 1.5 then provides the projection of the 3D LS quadrature set onto the unit circle for various orders for use in 2D problems. We have included the full quadrature set in this image including the quadrant-to-quadrant mapping.



(a)

(b)

(c)

(d)

Figure 1.5: Projection of the 3D Level-Symmetric angular quadrature set with orders (a) 2, (b) 4, (c) 8, and (d) 16 onto the x-y space on the unit circle.

1.3.2 Product Gauss-Legendre-Chebyshev Quadrature Set

The second angular quadrature set we will present is a Product Gauss-Legendre-Chebyshev (PGLC) set [9]. It is formed by the product-wise multiplication of a Gauss-Chebyshev quadrature in the azimuthal direction and a Gauss-Legendre quadrature in the polar direction. It has the following key differences from the Level Symmetric set:

- Does not have 90^o rotational invariance within the primary octant; however, we still maintain octant-to-octant symmetry via mapping;
- Has more control over the placement of the anglular directions within the primary octant;
- Quadrature weights are aligned with the polar level;
- Has strictly positive weights for all polar and azimuthal combinations.

From the listed differences, we can already discern some clear advantages and disadvantages from a fully-symmetric quadrature set like LS. If a high number of angles are required for a problem, then negative weights do not arise. This is beneficial for transport problems with significant discontinuities. Also, the quadrature directions can be preferentially distributed in the primary octant if required for a particular problem. For example, if the transport solution is smoothly varying in the polar direction and not in the azimuthal direction, then we can specify a larger number of quadrature points in the azimuthal direction, with much fewer points in the polar direction. However, this also highlights the fact that the quadrature weights are aligned with the polar level, which can lead to less accurate moment integrations for certain transport problems.

Because the PGLC quadrature set is formed by product-wise multiplication, we simply need to specify the component nodes and weights in both the azimuthal and polar directions to fully define all ordinates in the primary octant. The azimuthal direction, θ, uses the positive range of the Gauss-Chebyshev quadrature set [10]. With the azimuthal direction restricted to its positive values in the primary octant, this corresponds to the upper-right portion of the unit circle: $\theta \in [0, \pi/2]$. If we specify A azimuthal directions for our quadrature set in the primary octant, then the azimuthal nodes and weights can be directly stated as

$$\theta_m = \frac{2m-1}{4A}\pi \qquad \text{and} \qquad w_m = \frac{\pi}{2A}, \tag{1.35}$$

respectively.

For the polar direction, a Gauss-Legendre quadrature set is used [11]. Similar to the azimuthal direction, we restrict the integration of the polar direction to its positive values: $\xi \in (0, 1)$. If we specify P polar directions, then the cosine of the polar nodes, ξ, of our quadrature set are the positive roots of the $2P$-order Legendre polynomials taken over the inveral $[-1, 1]$. In this case, we simply discard the negative roots. The corresponding Legendre weights are given by the following formula,

$$w_n = \frac{2}{(1-\xi_n^2)(L'_{2P}(\xi_n))^2}, \tag{1.36}$$

where L'_{2P} is the derivative of the $2P$-order Legendre polynomial.

With the azimuthal directions specified by Eq. (1.35) and the polar cosines specified by the Legendre polynomial roots, any ordinate can now be determined by the definition of the angular directions in Eq. (1.30). The ordinate weights can be specified in a similar manner. From Eqs. (1.35) and (1.36), any ordinate weight,

$w_{m,n}$, can be calculated by the pairwise products of the azimuthal and polar weights: $w_{m,n} = w_m w_n$. This means that we can specify the integral, F, of some function $f(\theta, \gamma)$ over the primary octant of the unit sphere,

$$F = \sum_{m=1}^{A} \sum_{n=1}^{P} w_m w_n f(\theta_m, \gamma_n). \tag{1.37}$$

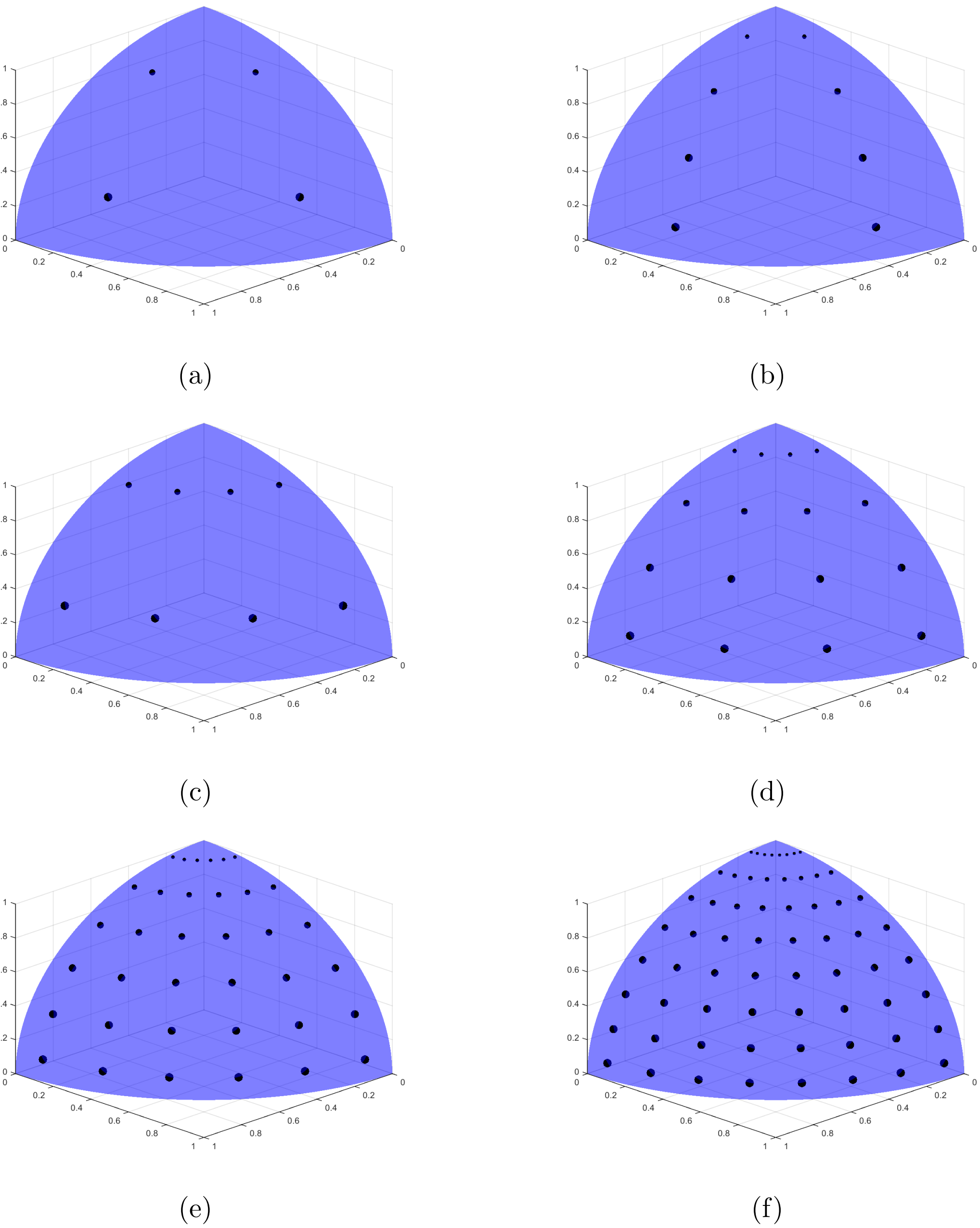
For this dissertation, we will use the following notation to define the product nature of the PGLC quadrature points: S_A^P. Here, A and P correspond to the number of azimuthal and polar directions in the primary octant, respectively. We demonstrate this definition in Figure 1.6 for the primary octant with several combinations of azimuthal and polar directions. Figure 1.7 then presents the projections of these quadrature sets onto the unit circle for use in 2D transport problems. Again, the size of the direction marker corresponds to the relative weight of the quadrature point. One can clearly see that the weights vary on the polar levels, and all azimuthal weights on a given polar level are constant.

1.4 Boundary Conditions

Using the energy and angular discretizations presented in Sections 1.2 and 1.3, respectively, we write the standard, steady-state, multigroup S_N transport equation for one angular direction, m, and one energy group, g:

$$\begin{aligned}\left(\vec{\Omega}_m \cdot \vec{\nabla} + \sigma_{t,g}\right) \Psi_{m,g} = \sum_{g'=1}^{G} \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sigma_{s,p}^{g' \to g} \sum_{n=-p}^{p} \Phi_{p,n,g'} Y_{p,n}(\vec{\Omega}_m) \\ + \frac{\chi_g}{4\pi} \sum_{g'=1}^{G} \nu\sigma_{f,g'} \Phi_{g'} + Q_{m,g}\end{aligned}, \tag{1.38}$$

where we have dropped the spatial parameter for clarity and is beholden to the



(a)

(b)

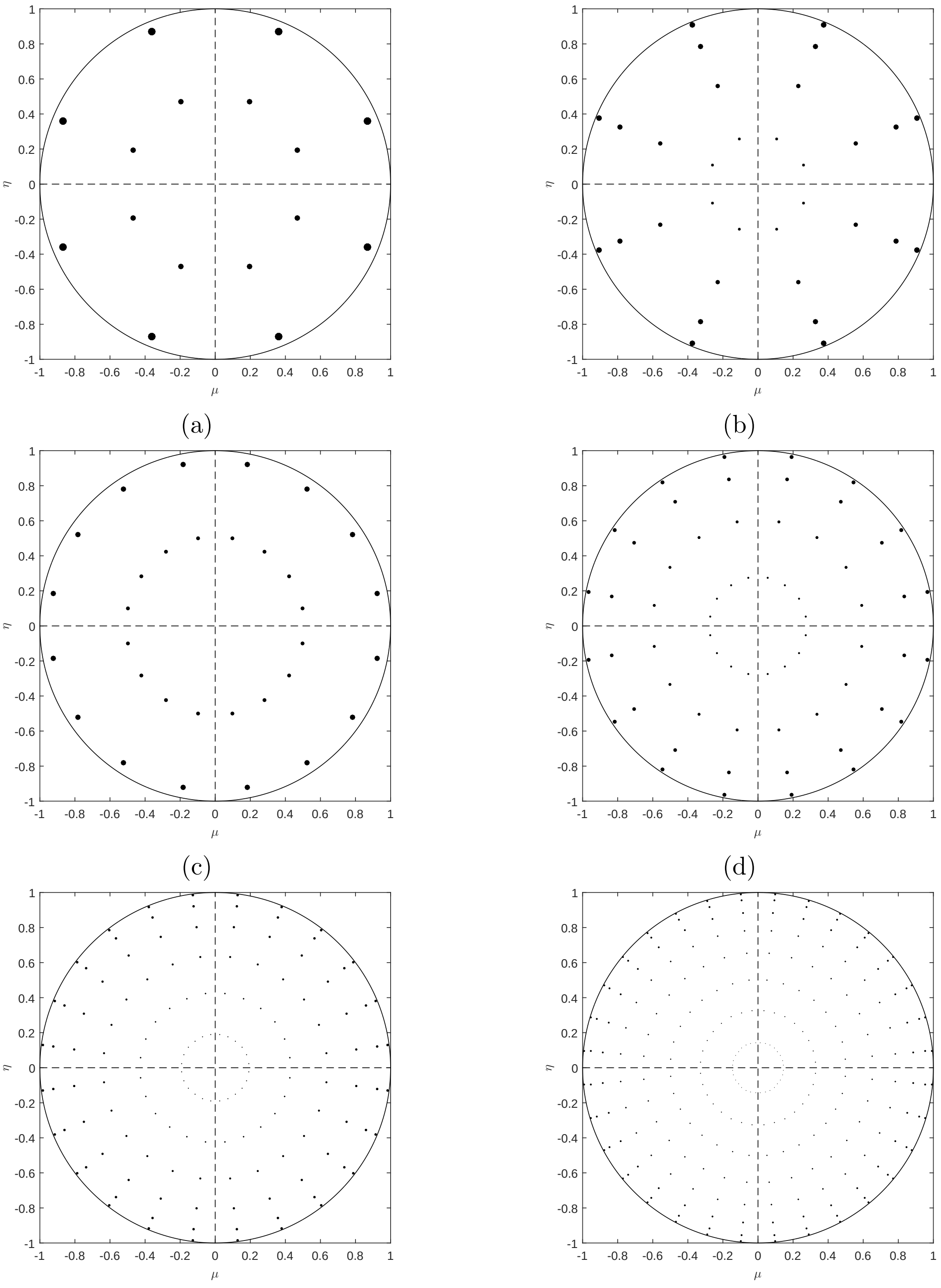
(c)

(d)

(e)

(f)

Figure 1.6: Product Gauss-Legendre-Chebyshev angular quadrature set with orders: (a) S_2^2, (b) S_2^4, (c) S_4^2, (d) S_4^4, (e) S_6^6, and (f) S_8^8.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1.7: Projection of the 3D Product Gauss-Legendre-Chebyshev angular quadrature set with orders: (a) S_2^2, (b) S_2^4, (c) S_4^2, (d) S_4^4, (e) S_6^6, and (f) S_8^8 onto the x-y space on the unit circle.

following general, discretized boundary condition:

$$\Psi_{m,g}(\vec{r}) = \Psi_{m,g}^{inc}(\vec{r}) + \sum_{g'=1}^{G} \sum_{\vec{\Omega}_{m'} \cdot \vec{n} > 0} \gamma_{g' \to g}^{m' \to m}(\vec{r}) \Psi_{m',g'}(\vec{r}). \tag{1.39}$$

These $(M\text{x}G)$ number of discrete, tightly-coupled equations are currently defined as continuous in space.

For this dissertation work, we will consider only one type of boundary conditions: Dirichlet-type boundaries (also called *first-type boundary condition* in some physics and mathematical communities). In particular, we will only utilize incoming-incident and reflecting boundary conditions which correspond to $\vec{r} \in \partial\mathcal{D}^d$ and $\vec{r} \in \partial\mathcal{D}^r$, respectively. The full domain boundary is then the union: $\partial\mathcal{D} = \partial\mathcal{D}^d \cup \partial\mathcal{D}^r$. This leads to the boundary condition being succinctly written for one angular direction, m, and one energy group, g as

$$\Psi_{m,g}(\vec{r}) = \begin{cases} \Psi_{m,g}^{inc}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^d \\ \\ \Psi_{m',g}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^r \end{cases} \tag{1.40}$$

where the reflecting angle is $\vec{\Omega}_{m'} = \vec{\Omega}_m - 2\left(\vec{\Omega}_m \cdot \vec{n}\right)\vec{n}$ and $\vec{n}$ is oriented outward from the domain. To properly utilize the reflecting boundary condition that we have proposed, the angular quadrature set defined in Section 1.3 needs the following properties:

1. The reflected directions, $\vec{\Omega}_{m'}$, are also in the quadrature set for all $\vec{r} \in \partial\mathcal{D}^r$.
2. The weights of the incident, w_m, and reflected, $w_{m'}$, angles must be equal.

For problems where the reflecting boundaries align with the x, y, z axes, this will not be an issue with standard quadrature sets (*e.g.* level-symmetric or Gauss-Legendre-

Chebyshev). However, if the reflecting boundaries do not align in this manner, then additional care must be employed in calculating appropriate angular quadrature sets.

1.5 Spatial Discretization

For the spatial discretization of the problem domain, we simplify Eq. (1.38) into a single energy group and drop the fission term (it can be lumped into the 0th order external source term and will act similarly to the total interaction term)

$$\vec{\Omega}_m \cdot \vec{\nabla}\Psi_m + \sigma_t \Psi_m = \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m) \left[\sigma_{s,p}\Phi_{p,n,} + Q_{p,n}\right] \tag{1.41}$$

to form M ($m = 1, ..., M$) angularly discrete equations. We then lay down an unstructured mesh, $\mathbb{T}_h \in \mathbb{R}^d$, over the spatial domain, where d is the dimensionality of the problem ($d = 1, 2, 3$). This mesh consists of non-overlapping spatial elements to form a complete union over the entire spatial domain: $\mathcal{D} = \bigcup_{K \in \mathbb{T}_h} K$. To form the DGFEM set of equations [12, 13], we consider a spatial cell $K \in \mathbb{R}^d$ which has N_V^K vertices and N_f^K faces. Each face of cell K resides in dimension $\mathbb{R}^{d-1}$ and is formed by a connection of a subset of vertices. For a 1D problem, each face is a single point. For a 2D problem, each face is a line segment connecting two distinct points. For a 3D problem, each face is a $\mathbb{R}^2$ closed polygon (not necessarily coplanar) which may or may not be convex. An example of this interconnection between elements is given for a $\mathbb{R}^2$ problem in Figure 1.8 between our cell of interest, K, and another cell, K', separated by the face f. We have chosen the normal direction of the face to have orientation from cell K to cell K' while we form the DGFEM equations for cell K. This means that if we were instead analyzing cell K', then the face f normal, $\vec{n}'$, would be opposite (*i.e.* $\vec{n}' = -\vec{n}$).

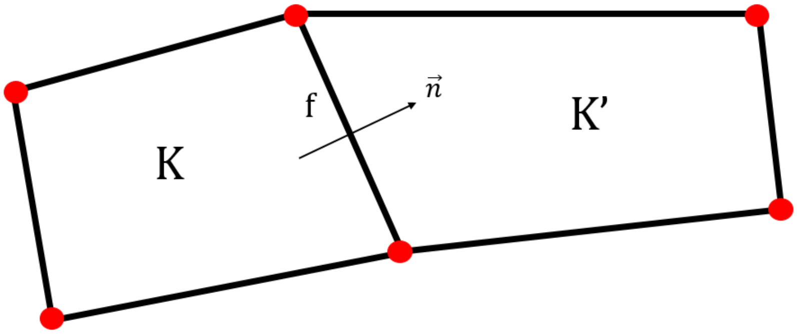


Figure 1.8: Two cells of the spatial discretization with the connecting face, f, with normal direction, $\vec{n}$, oriented from cell K to cell K'.

Next, we left-multiply Eq. (1.41) by an appropriate test function b_m, integrate over cell K, and apply Gauss' Divergence Theorem to the streaming term to obtain the Galerkin weighted-residual for cell K for an angular direction $\vec{\Omega}_m$:

$$\begin{aligned} &-\left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\right)_K + \sum_{f=1}^{N_f^K} \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_f + \left(\sigma_t b_m, \Psi_m\right)_K \\ &= \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p}\, b_m, \Phi_{p,n,}\right)_K + \left(b_m, Q_{p,n}\right)_K \right] \end{aligned}. \qquad (1.42)$$

The cell boundary fluxes, $\tilde{\Psi}_m$, will depend on the cell boundary type and will be defined shortly. The cell boundary $\partial\mathcal{D}_K = \bigcup_{f \in N_f^K} f$ is the closed set of the N_f^K faces of the geometric cell. The inner products:

$$\left(u, v\right)_K \equiv \int_K u\, v\, dr \qquad (1.43)$$

and

$$\left\langle u, v \right\rangle_f \equiv \int_f u\, v\, ds \qquad (1.44)$$

correspond to integrations over the cell and faces, respectively, where $dr \in \mathbb{R}^d$ is within the cell and $ds \in \mathbb{R}^{d-1}$ is along the cell boundary. We note that we will use this notation of the inner product for the remainder of the dissertation unless otherwise stated. We then separate the summation of the cell K boundary integration terms into two different types: outflow boundaries ($\partial K^+ = \{\vec{r} \in \partial K : \vec{n}(\vec{r}) \cdot \vec{\Omega}_m > 0\}$) and inflow boundaries ($\partial K^- = \{\vec{r} \in \partial K : \vec{n}(\vec{r}) \cdot \vec{\Omega}_m < 0\}$). The inflow boundaries can further be separated into inflow from another cell: $\partial K^- \backslash \partial \mathcal{D}$; inflow from incident flux on the domain boundary: $\partial K^- \cap \partial \mathcal{D}^d$; and reflecting domain boundaries: $\partial K^- \cap \partial \mathcal{D}^r$. At this point, we note that the derivation can comprise an additional step by using Gauss' Divergence Theorem again on the streaming term. This is sometimes performed for radiation transport work so that mass matrix lumping can be performed, but we will not do so here at this time. Therefore, with the cell boundary terminology as proposed, Eq. (1.42) can be written into the following form:

$$\begin{aligned} &- \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\right)_K + \left(\sigma_t b_m, \Psi_m\right)_K \\ &+ \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^+} + \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \backslash \partial \mathcal{D}} \\ &+ \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^d} + \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\ &= \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p}\, b_m, \Phi_{p,n,}\right)_K + \left(b_m, Q_{p,n}\right)_K \right] \end{aligned} . \tag{1.45}$$

We can now deal with the boundary fluxes, $\tilde{\Psi}_m$, by enforcing the ubiquitously-used *upwind scheme*. In simple nomenclature, the upwind scheme corresponds to using the angular flux values within the cell for outflow boundaries and angular flux values outside the cell for inflow boundaries. Mathematically, the upwind scheme can succinctly be written as the following for all boundary types,

$$\tilde{\Psi}_m(\vec{r}) = \begin{cases} \Psi_m^-, & \partial K^+ \\ \Psi_m^+, & \partial K^- \backslash \partial \mathcal{D} \\ \Psi_m^{inc}, & \partial K^- \cap \partial \mathcal{D}^d \\ \Psi_{m'}^-, & \partial K^- \cap \partial \mathcal{D}^r \end{cases}, \tag{1.46}$$

when the following trace is applied to the angular fluxes :

$$\Psi_m^{\pm}(\vec{r}) \equiv \lim_{s \to 0^{\pm}} \Psi_m\Big(\vec{r} + s(\vec{\Omega}_m \cdot \vec{n})\vec{n}\Big). \tag{1.47}$$

This trace has the notation, with $\vec{n}$ pointing outwards from cell K, of Ψ_m^- corresponding to fluxes within the cell and Ψ_m^+ corresponding to fluxes out of the cell. Now, using the upwind scheme as previously defined, we can write our complete set of DGFEM equations for cell K as

$$\begin{aligned} -\Big(\vec{\Omega}_m{\cdot}\vec{\nabla} b_m, \Psi_m\Big)_K + \Big(\sigma_t b_m, \Psi_m\Big)_K + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^-\Big\rangle_{\partial K^+} \\ + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^+\Big\rangle_{\partial K^- \backslash \partial \mathcal{D}} + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_{m'}^-\Big\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\ = \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \Big[\Big(\sigma_{s,p}\, b_m, \Phi_{p,n,}\Big)_K + \big(b_m, Q_{p,n}\big)_K \Big] \\ - \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^{inc}\Big\rangle_{\partial K^- \cap \partial \mathcal{D}^d} \end{aligned}. \tag{1.48}$$

We note that fluxes without the trace superscript are all within the cell. By completely defining our mathematical formulation for an arbitrary spatial cell, it is easy to see that the full set of equations to define our discretized solution space for a single angle and energy group comprises of a simple double integration loop. The full set of equations can be formed by looping over all spatial cells, $\mathcal{D} = \bigcup_{K \in \mathbb{T}_h} K$, while further looping over all faces within each cell, $\partial\mathcal{D}_K = \bigcup_{f \in N_f^K} f$.

1.5.1 Convergence Rates of the DGFEM S_N Equation

Because we seek to investigate the use of high-order spatial basis functions for the transport equation, we need to form an estimate of the spatial error based on some measure of the mesh. We do this by taking Eq. (1.48), performing another integration-by-parts on the streaming term, multiplying by the angular weight, w_m, and summing over all elements and all angular directions. We also change the notation of the test function from b_m to Ψ_m^* to ease notation at a later step. This leads to the variational form for the 1-group S_N equation:

$$\begin{aligned}
&\sum_{m=1}^{M} w_m \Big[\Big(\Psi_m^*, \vec{\Omega}_m \cdot \vec{\nabla} \Psi_m \Big)_{\mathcal{D}} + \Big(\sigma_t \Psi_m^*, \Psi_m \Big)_{\mathcal{D}} \Big] \\
&+ \sum_{m=1}^{M} w_m \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, \Psi_m^{*+}, [\![\Psi_m]\!] \Big\rangle_{E_h^i} \\
&+ \sum_{m=1}^{M} w_m \Big[\Big\langle (\vec{\Omega}_m \cdot \vec{n})\, \Psi_m^*, \Psi_{m'} \Big\rangle_{\partial\mathcal{D}_m^{r-}} - \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, \Psi_m^*, \Psi_m \Big\rangle_{\partial\mathcal{D}_m^{-}} \Big] \\
&= \sum_{m=1}^{M} w_m \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \Big(\Psi_m^*, \sigma_{s,p}\, \Phi_{p,n,} + Q_{p,n} \Big)_{\mathcal{D}} \\
&- \sum_{m=1}^{M} w_m \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, \Psi_m^*, \Psi_m^{inc} \Big\rangle_{\partial\mathcal{D}_m^{-}}
\end{aligned} \qquad (1.49)$$

where the inner products over the whole domain and over all interior faces are

$$\Big(u, v \Big)_{\mathcal{D}} \equiv \sum_{K \in \mathbb{T}_h} \Big(u, v \Big)_K, \qquad (1.50)$$

and

$$\Big\langle u, v \Big\rangle_{E_h^i} \equiv \sum_{f \in E_h^i} \Big\langle u, v \Big\rangle_f, \qquad (1.51)$$

respectively. The interior faces are designated as the non-repeating set: $E_h^i \in \cup_{K\in\mathbb{T}_h}\partial K\backslash\partial\mathcal{D}$. In Eq. (1.49), the interior jump term, $[\![\Psi_m]\!]$, is defined as,

$$[\![\Psi_m]\!] = \Psi_m^+ - \Psi_m^-, \tag{1.52}$$

and along with the inflow basis function, b_m^+, is beholden to the following trace condition:

$$\begin{gathered} \Psi_m^{\pm} \equiv \lim_{s\to 0^{\pm}} \Psi_m(\vec{r} - u(\vec{r})s\,\vec{\Omega}_m) \\ u(\vec{r}) = \operatorname{sgn}\left(\vec{n}(\vec{r})\cdot\vec{\Omega}_m\right) \end{gathered}. \tag{1.53}$$

We can give compact notation to the variational form of the DGFEM transport equation in Eq. (1.49) by defining the bilinear form in Eq. (1.54). We have changed the sequence of the summation over directions and over the elements for the boundary terms.

$$\begin{gathered} a(\Psi^*,\Psi) = \sum_{m=1}^{M} w_m\Big[\Big(\Psi_m^*, \vec{\Omega}_m\cdot\vec{\nabla}\Psi_m\Big)_{\mathcal{D}} + \Big(\sigma_t\Psi_m^*,\Psi_m\Big)_{\mathcal{D}}\Big] \\ +\sum_{m=1}^{M} w_m\Big\langle(\vec{\Omega}_m\cdot\vec{n})\,\Psi_m^{*+}, [\![\Psi_m]\!]\Big\rangle_{E_h^i} - \sum_{f\in\partial\mathcal{D}}\sum_{\vec{\Omega}_m\cdot\vec{n}}\Big\langle(\vec{\Omega}_m\cdot\vec{n})\,\Psi_m^*,\Psi_m\Big\rangle_f \\ +\sum_{f\in\partial\mathcal{D}^r}\sum_{\vec{\Omega}_m\cdot\vec{n}}\Big\langle(\vec{\Omega}_m\cdot\vec{n})\,\Psi_m^*,\Psi_{m'}\Big\rangle_f - \sum_{p=0}^{N_P}\sum_{n=-p}^{p}\frac{2p+1}{4\pi}\Big(\sigma_{s,p}\,\Phi_{p,n,}^*,\Phi_{p,n,}\Big)_{\mathcal{D}} \end{gathered} \tag{1.54}$$

This bilinear form is positive definite ($a(\Psi,\Psi) > 0$) but obviously not symmetric. It also holds Galerking orthogonality, which means that the jumps across elements for the exact solution are zero,

$$a(\Psi^*,\Psi-\Psi_{exact}) = 0, \qquad m = 1,...,M. \tag{1.55}$$

Because of the positive definiteness and Galerkin orthogonality of the bilinear form, we can use it to define a DG-norm,

$$||u||_{DG} = a(u,u). \tag{1.56}$$

We can also give a more simplified definition of the DG norm,

$$||u||_{DG}^2 = \sum_{K \in \mathbb{T}_h} \left[||u||_{L^2(K)}^2 + \frac{1}{2} ||u^+ - u^-||_{L^2(\partial K)} \right], \tag{1.57}$$

which contains the standard L_2 norm in the element interiors as well as additional L_2 norms on the element boundaries corresponding to the jump terms across the mesh cells.

The convergence of DGFEM methods for the hyperbolic systems has been extensively studied [14, 15, 16, 17]. With the discretized flux solutions, $\Psi_h \in W_{\mathcal{D}}^h$ and $\Phi_h \in W_{\mathcal{D}}^h$, corresponding to our unstructured, $\mathbb{T}_h$, we can define an error for our DGFEM transport solution with the DG norm for the angular fluxes,

$$||\Psi - \Psi_{exact}||_{DG} \leq C \frac{h^q}{(p+1)^q}, \tag{1.58}$$

and flux moments,

$$||\Phi - \Phi_{exact}||_{DG} \leq C \frac{h^q}{(p+1)^q}, \tag{1.59}$$

respectively. In Eqs. (1.58) and (1.59), $q = \min(p + 1/2, r - 1/2)$, h is the maximum diameter of all the mesh elements, p is the polynomial completeness of the finite element function space (this will be explained further in Chapter **??**), r is the regularity index of the transport solution, and C is a constant that is independent of the mesh employed. We can also give an estimate for the transport solution error in the standard L_2 norm,

$$||\Phi - \Phi_{exact}||_{L_2} \leq C\frac{h^{q+1/2}}{(p+1)^q}, \tag{1.60}$$

where q has the same definition as before. We can see that for the L_2 norm the convergence rate is simply $1/2$ integer more than the DG norm for both the polynomial order and the regularity index.

Investigating the definition of the q term in Eqs. (1.58), (1.59), and (1.60) further, we can see that our DGFEM transport convergence rates are all limited by the regularity, r, of the solution space. The spaces to which the transport equation can reside have been investigated by others in previous works [17, 18, 19]. If we have sufficiently smooth data, then the exact transport solution belongs, at most, in the $H^{3/2-\epsilon}(\mathcal{D})$ Hilbert space. This yields a solution regularity of $r = \frac{3}{2} - \epsilon$, where ϵ is positive and extremely small. This means that for most practical occurences, the observed converge regularity is simply $\frac{3}{2}$. For this case it is sufficient to have piecewise polynomial cross section data and incident boundary fluxes that align with our mesh. However, in the case of a pure absorber or void, the exact transport solution lives in the $H^{1/2-\epsilon}(\mathcal{D})$ Hilbert space. Again, the practical irregularity becomes $\frac{1}{2}$. From these spaces, one would think that the transport solution regularity would impede the use of higher-order finite element spaces. We note, however, that these regularity indices only apply to the asymptotic convergence range, which usually only applies to very fine meshes that are much smaller than typically employed meshes. Therefore, we expect to capture up to order $(p+1)$ convergence in preasymptotic ranges that would be employed for a wide variety of transport problems. Results capturing this irregularity behavior are presented in Chapter **??**.

Finally, we also seek to define the transport solution convergence rates in terms other than the maximum element diameter, h. For polytope meshes, this metric may not be the easiest to compute or report if there is large variability in the polygonal or polyhedral elements of the mesh. We seek to re-express the convergence rates in terms of the total number of degrees of freedom in the problem, N_{dof}, instead of the maximum element diameter, h. First, we relate the total number of elements, N_{ele}, to the maximum element diameter,

$$N_{ele} \propto h^{-d}, \tag{1.61}$$

where we assumed that the polytope elements are convex and reasonably regular. We then assume that the finite element functional space on each element is the Serendipity space (we go into further detail in Chapter **??**) [20, 21]. This means that the number of degrees of freedom per element is proportional to the polynomial order: $N_{dof} \propto ph^{-d}$ or $h \propto \left(\frac{N_{dof}}{p}\right)^{-1/d}$. We substitute this result into Eq. (1.60) to yield an L_2 convergence rate in terms of the problem's total number of degrees of freedom,

$$||\Phi - \Phi_{exact}||_{L_2} \leq C(p) N_{dof}^{-\frac{q+1/2}{d}}, \tag{1.62}$$

where the proportionality constant, $C(p)$, now has the form,

$$C(p) = \frac{p^{\frac{q+1/2}{d}}}{(p+1)^q}. \tag{1.63}$$

For a transport problem that is not bound by the solution regularity, we can express the simplified convergence rate,

$$||\Phi - \Phi_{exact}||_{L_2} \leq C(p)\, N_{dof}^{-\frac{p+1}{d}}, \qquad C(p) = \frac{p^{\frac{p+1}{d}}}{(p+1)^{p+1/2}}. \tag{1.64}$$

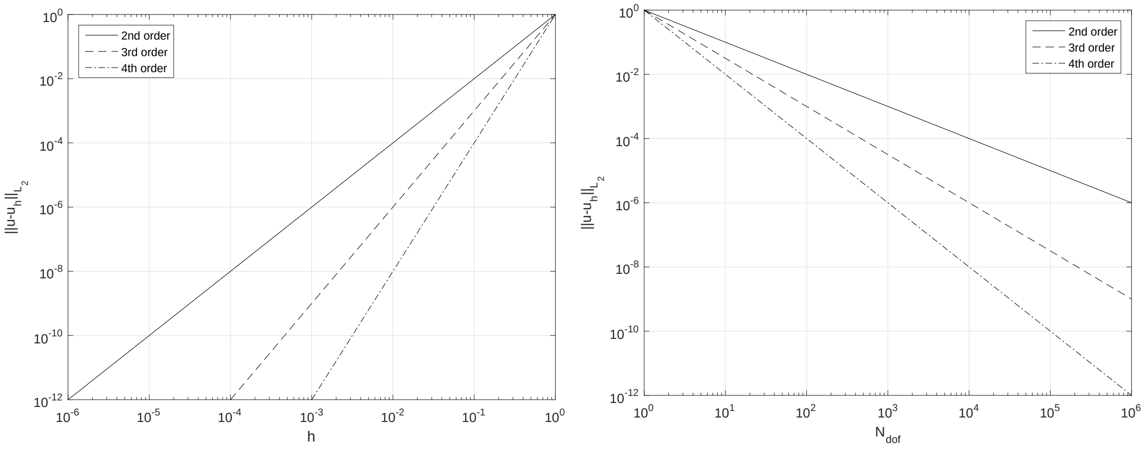


Figure 1.9: Example of the theoretical convergence rates for a DGFEM transport problem that is not bound by solution regularity in terms of the maximum element diameter (left) and number of degrees of freedom (right).

The result of Eq. (1.64) states that we expect convergence rates of N_{dof}^{-1} and $N_{dof}^{-2/3}$ for linear ($p = 1$) functional spaces in 2D and 3D, respectively. Conversely, quadratic ($p = 2$) functional spaces will yield convergence rates of $N_{dof}^{-3/2}$ and N_{dof}^{-1} in 2D and 3D, respectively. The proportionality constant, $C(p)$, in Eq. (1.64) is a monotonically decreasing function for $p \geq 1$ in 2D and 3D (this does not hold for 1D problems). Finally, we show a comparison between the convergence rate estimates of Eqs. (1.60) and (1.62) for a 2D problem not bounded by the solution regularity ($q+1/2 = p+1$) in Figure 1.9. We also simply set $C(p) = 1$ for all the curves to better show a comparison between the rates. We can get a good idea of the power of using higher-order shape functions with curves based on N_{dof}. If our transport problem takes about the same amount of calculation time to solve for the same number of degrees of freedom, we can see that we would yield a more accurate answer for little additional cost.

1.5.2 Elementary Matrices on an Arbitrary Spatial Cell

In Eq. (1.48), we presented the full set of spatially-discretized equations needed to solve the angular flux solution for cell K for a single angular direction. In the equation, several terms of various types arise including interaction, $\left(\sigma b_m, \Psi_m\right)_K$, streaming, $\left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\right)_K$, and surface, $\left\langle \left(\vec{\Omega}_m \cdot \vec{n}\right) b_m, \Psi_m\right\rangle_{\partial K}$. Each of these correspond to a different elementary matrix type. We now present how to compute the mass, streaming, and surface matrices in Sections 1.5.2.1, 1.5.2.2, and 1.5.2.3, respectively.

1.5.2.1 Elementary Mass Matrices

In the spatially discretized equations presented in Section 1.5, there are several reaction terms that appear with the form: $\left(\sigma b_m, \Psi_m\right)_K$ for a given angular direction, m, and for a spatial cell, K. In FEM analysis these reaction terms are ubiquitously referred to as the mass matrix terms [22, 23]. For cell K, we define the elementary mass matrix, $\mathbf{M}$, as

$$\mathbf{M}_K = \int_K \mathbf{b}_K \, \mathbf{b}_K^T \, dr, \tag{1.65}$$

where $\mathbf{b}_K$ corresponds to the set of N_K basis functions that have non-zero measure in cell K. Depending on the FEM basis functions utilized, the integrals in Eq. (1.65) can be directly integrated analytically. However, if in general, the basis functions cannot be analytically integrated on an arbitrary set of cell shapes, then a numerical integration scheme becomes necessary. If we define a quadrature set, $\left\{\vec{x}_q, w_q^K\right\}_{q=1}^{N_q}$, for cell K, consisting of N_q points, $\vec{x}_q$, and weights, w_q^K, then we can numerically calculate the mass matrix by the following

$$\mathbf{M}_K = \sum_{q=1}^{N_q} w_q^K \mathbf{b}_K(\vec{x}_q)\, \mathbf{b}_K^T(\vec{x}_q). \tag{1.66}$$

In this case, it is necessary that the sum of the weights of this quadrature set exactly equal the geometric measure of cell K. This means that $\sum_{q=1}^{N_q} w_q^K$ is equal to the cell width in 1 dimension, the cell area in 2 dimensions, and the cell volume in 3 dimensions.

Since $\mathbf{b}_K$ consists of a column vector for the basis functions and $\mathbf{b}_K^T$ consists of a row vector, then their multiplication will obviously yield a full $(N_K \,\mathrm{x}\, N_K)$ matrix. This matrix is written for completeness of this discussion on the mass matrix:

$$\mathbf{M}_K = \begin{bmatrix} \int_K b_1\, b_1 & \dots & \int_K b_1\, b_j & \dots & \int_K b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_i\, b_1 & \dots & \int_K b_i\, b_j & \dots & \int_K b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_{N_K}\, b_1 & \dots & \int_K b_{N_K}\, b_j & \dots & \int_K b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{1.67}$$

where an individual matrix entry is of the form:

$$M_{i,j,K} = \int_K b_i\, b_j. \tag{1.68}$$

1.5.2.2 Elementary Streaming Matrices

Next, we will consider the streaming term that has the form: $\left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\right)_K$ for a given angular direction, m, and for a spatial cell, K. $\vec{\nabla}$ is the gradient operator in physical space. It has the form of $\vec{\nabla} = \left[\frac{d}{dx}\right]$ in 1 dimension, the form of $\vec{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right]$ in 2 dimensions, and the form of $\vec{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]$ in 3 dimensions. Since for every cell, the streaming term is applied for all M angles in the angular discretization, we define the analytical elementary streaming matrix:

$$\vec{\mathbf{G}}_K = \int_K \vec{\nabla} \mathbf{b}_K \, \mathbf{b}_K^T \, dr, \tag{1.69}$$

which has dimensionality (N_KxN_Kxd). We choose to store the elementary streaming matrix in this form and not store M separate (N_KxN_K) local matrices corresponding to the application of the dot product ($\vec{\Omega}_m \cdot \int_K \vec{\nabla} \mathbf{b}_K \, \mathbf{b}_K^T \, dr$). Instead, we simply evaluate the dot product with the appropriate angular direction whenever necessary. This has great benefit when trying to run large transport problems when memory becomes a premium and processor operations are not our limiting bottleneck.

Just like the elementary mass matrix, we can use the same spatial quadrature set, $\left\{ \vec{x}_q, w_q^K \right\}_{q=1}^{N_q}$, for cell K to numerically calculate the streaming matrix:

$$\vec{\mathbf{G}}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla} \mathbf{b}_K(\vec{x}_q) \, \mathbf{b}_K^T(\vec{x}_q). \tag{1.70}$$

In this case, this local cell-wise streaming matrix has the full matrix form:

$$\vec{\mathbf{G}}_K = \begin{bmatrix} \int_K \vec{\nabla} b_1 \, b_1 & \dots & \int_K \vec{\nabla} b_1 \, b_j & \dots & \int_K \vec{\nabla} b_1 \, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_i \, b_1 & \dots & \int_K \vec{\nabla} b_i \, b_j & \dots & \int_K \vec{\nabla} b_i \, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_{N_K} \, b_1 & \dots & \int_K \vec{\nabla} b_{N_K} \, b_j & \dots & \int_K \vec{\nabla} b_{N_K} \, b_{N_K} \end{bmatrix}, \tag{1.71}$$

where an individual matrix entry is of the form:

$$\vec{G}_{i,j,K} = \int_K \vec{\nabla} b_i \, b_j. \tag{1.72}$$

1.5.2.3 Elementary Surface Matrices

Finally, the last terms to consider of the discretized transport equation are those found on the faces of the cell boundary: $\vec{\Omega}_m \cdot \left\langle \vec{n}\, b_m, \Psi_m \right\rangle_{\partial K}$. These terms are analagous to the cell mass matrix but are computed on the cell boundary with dimensionality $(d-1)$. Analyzing a single face, f, in cell K, the analytical surface matrix is of the form,

$$\vec{\mathbf{F}}_{f,K} = \int_f \vec{n}(\vec{r})\, \mathbf{b}_K\, \mathbf{b}_K^T\, ds, \tag{1.73}$$

where we allow the outward surface normal, $\vec{n}$, to vary along the cell face. For 1D problems as well as 2D problems with colinear cell faces (no curvature), the outward normals would be constant along the entire face. However, there are many cases where 3D mesh cells would not have coplanar vertices along a face. Then, the outward normal would not be constant along the face and would need to be taken into account during integration procedures. A simple example of non-coplanar face vertices would be an orthogonal hexahedral cell that has its vertices undergo a randomized displacement.

With the analytical form of the surface matrices defined in Eq. (1.73), we can see that they have dimensionality, $(N_K \mathrm{x} N_K \mathrm{x} d)$. This is the same dimensionality as the cell streaming term. However, it is possible to reduce the dimensionality of the surface matrices if it is desired to reduce the memory footprint. There are some basis sets where all but $N_b^{f,K}$ basis functions are zero along face f. If we also restrict the mesh cell faces of our transport problems to have colinear (in 2D) or coplanar (in 3D) vertices so that the outward normal is constant along a face f, then we can define the surface matrix as $\int_f \mathbf{b}_K\, \mathbf{b}_K^T\, ds$. For these basis sets with $N_b^{f,K}$ non-zero face values on colinear/coplanar face f, the surface matrix has reduced dimensionality of

$(N_b^{f,K} \text{x} N_b^{f,K})$.

Just like the cell mass and streaming matrices, it is possible that the basis functions cannot be integrated analytically. Analogous to the cell-wise quadrature, we can define a quadrature set for face f: $\left\{ \vec{x}_q, w_q^f \right\}_{q=1}^{N_q^f}$. This quadrature set is not specific for just one of the cells that face f separates. If the quadrature set can exactly integrate the basis functions of both cells K and K' (as defined by Figure 1.8), then only 1 quadrature set needs to be defined for both cells. Using this quadrature set, we can numerically calculate the surface matrix for face f along cell K:

$$\vec{\mathbf{F}}_{f,K} = \sum_{q=1}^{N_q^f} w_q^f \vec{n}(\vec{x}_q)\, \mathbf{b}_K(\vec{x}_q)\, \mathbf{b}_K^T(\vec{x}_q). \tag{1.74}$$

Similar to the cell-wise spatial quadrature sets, the sum of the weights of these face-wise quadrature sets needs to exactly equal the geometric measure of face f. This means that $\sum_{q=1}^{N_q^f} w_q^f$ is equal to 1.0 in 1 dimension, the length of the face edge in 2 dimensions and the face area in 3 dimensions.

Using the same notation as the cell-wise mass and streaming matrices, the local face-wise surface matrix for face f has the full matrix form,

$$\vec{\mathbf{F}}_{f,K} = \begin{bmatrix} \int_f \vec{n}\, b_1\, b_1 & \dots & \int_f \vec{n}\, b_1\, b_j & \dots & \int_f \vec{n}\, b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\, b_i\, b_1 & \dots & \int_f \vec{n}\, b_i\, b_j & \dots & \int_f \vec{n}\, b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\, b_{N_K}\, b_1 & \dots & \int_f \vec{n}\, b_{N_K}\, b_j & \dots & \int_f \vec{n}\, b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{1.75}$$

where an individual matrix entry is of the form:

$$\vec{F}_{i,j,f,K} = \int_f \vec{n}\, b_i\, b_j. \tag{1.76}$$

1.6 Solution Procedures

To this point, we have properly described the procedures to discretize the transport problem in energy, angle, and space. Combining the results of Sections 1.2, 1.3, and 1.5, we write the fully-discretized DGFEM multigroup S_N equations for an element K, where the test function $b_{m,g}$ for a single direction and energy group is now used:

$$\begin{aligned} -&\left(\vec{\Omega}_m\cdot\vec{\nabla}b_{m,g}, \Psi_{m,g}\right)_K + \left(\sigma_{t,g}b_{m,g}, \Psi_{m,g}\right)_K + \left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g}, \Psi^-_{m,g}\right\rangle_{\partial K^+} \\ &+\left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g}, \Psi^+_{m,g}\right\rangle_{\partial K^-\setminus\partial\mathcal{D}} + \left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g}, \Psi^-_{m',g}\right\rangle_{\partial K^-\cap\partial\mathcal{D}^r} \\ &=\sum_{g'=1}^{G}\sum_{p=0}^{N_P}\frac{2p+1}{4\pi}\sum_{n=-p}^{p}Y_{p,n}(\vec{\Omega}_m)\left(\sigma_{s,p}^{g'\to g}\,b_{m,g}, \Phi_{p,n,g'}\right)_K \\ &+\left(b_{m,g}, Q_{m,g}\right)_K - \left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g}, \Psi^{inc}_{m,g}\right\rangle_{\partial K^-\cap\partial\mathcal{D}^d} \end{aligned} \quad . \qquad (1.77)$$

All of the notations used in Eq. (1.77) remain unchanged from Section 1.5.

We now spend the remainder of this chapter discussing various methodologies to efficiently solve the tightly-coupled system of equations composing our transport problem. Section 1.6.1 details the iterative procedures used to solve the transport problem in energy and angle, and Section 1.6.2 then describes how we solve the spatial portion of the problem for a single energy/angle iteration.

1.6.1 Angle and Energy Iteration Procedures

The fully discretized transport equation has an angular flux solution, $\mathbf{\Psi}$, with dimensionality of (GxMxN_{dof}). The angular flux moments, $\mathbf{\Phi}$, have dimensionality of (GxN_{mom}xN_{dof}). Depending on the necessary fidelity of the problem, the full phase-space of the solution can become extremely large to solve. We can have *billions* of total unknowns to solve for if we simply have $N_{dof} \approx O(10^6)$, $M \approx O(10^2)$, and $G \approx O(10^1)$. These orders of number of unknowns in space, angle, and energy are of reasonable size for 3D transport problems.

In theory, if we had the computer memory, we could construct a left-hand-side matrix of dimensionality $(G\text{x}M\text{x}N_{dof})\text{x}(G\text{x}M\text{x}N_{dof})$ with a corresponding right-hand-side vector of dimensionality $(G\text{x}M\text{x}N_{dof})\text{x}1$, we could then directly solve for the full phase-space angular flux solution at once. However, because the dimensional space of the unknowns can rapidly grow and become too large for hardware memory, transport problems have traditionally been solved iteratively. We now detail the procedures that we will employ to iteratively obtain the phase-space solution in energy and angle.

Because the only coupling that arises between the set of multigroup S_N equations is between the energy groups in the scattering source, our iterative procedures principally lie in the energy domain. Figure 1.10 presents a pair of scattering matrices that show typical coupling between energy groups for neutronics problems. Depending on how the group boundaries are established along with a possible need for higher fidelity in energy, thermal upscattering may or may or may not be present in the problem. We can see from Figure 1.10 that the purely-downscattering matrix only has 1-way coupling, from high-to-low in the group energies. This means that if we progress through the energy groups from $g = 1, ..., G$ (in what is called an *outer iteration*), then we only have to do so once and we are done. However, there are many transport problems where we wish to have several energy groups in the thermal region. This leads to thermal neutron upscattering and causes the lower-right portion of the scattering matrix to be full as seen in the bottom of Figure 1.10. Then, depending on how the groups are structured, multiple outer iterations may be necessary to fully converge the scattering source.

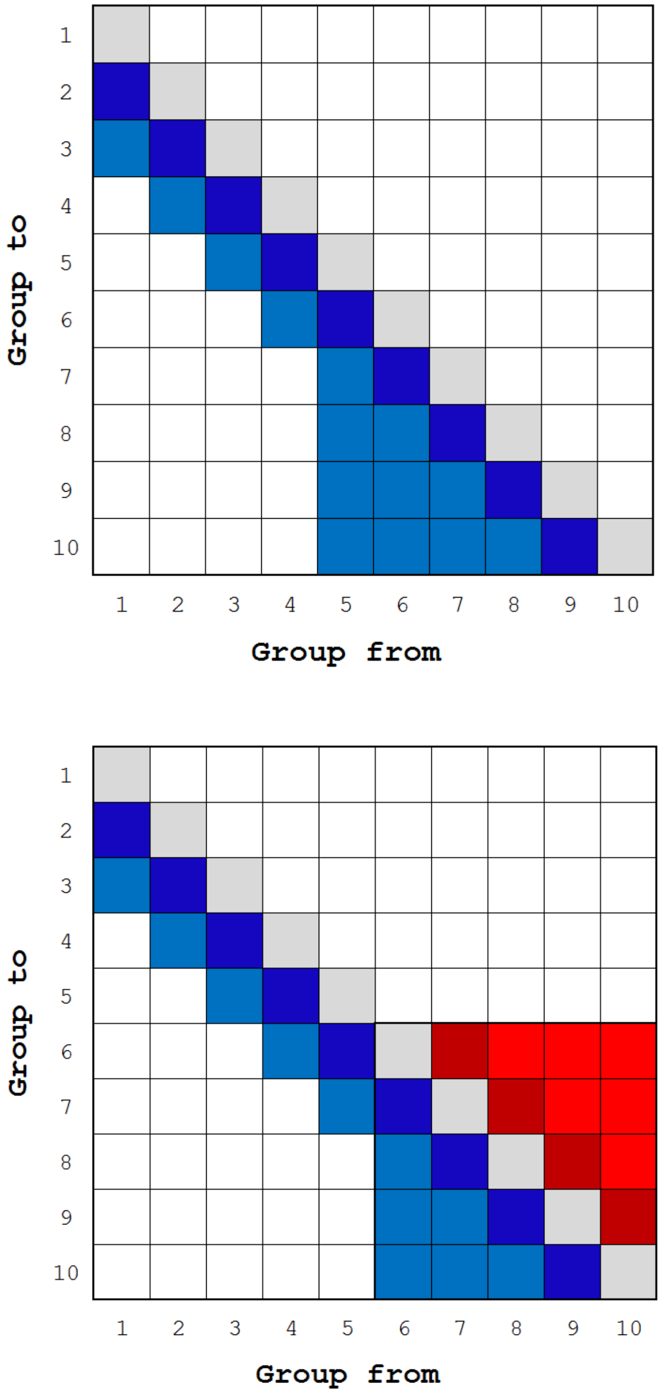


Figure 1.10: Scattering matrices (top) without and (bottom) with upscattering. The gray corresponds to within-group scattering; the blue corresponds to down-scattering in energy; and the red corresponds to up-scattering in energy.

We now describe the full iterative procedures required to converge the scattering source in detail. We begin by defining a new concept called a *group set*, which is simply a collection of contiguous groups. The group sets are ordered from high-to-low energy just like the groups and are non-overlapping with the adjoining sets. We demonstrate this concept with a simple example. If we have a problem with 10 groups ($G = 10$), we then choose to aggregate them into 3 group sets. Group set 1 contains $g \in [1, 3]$, group set 2 contains $g \in [4, 7]$, and group set 3 contains $g \in [8, 10]$. With these group sets defined we then choose to employ a double iteration loop to converge the scattering matrix. The outer iterations in this procedure perform residual calculations by looping through the group sets. Then, for each group set in every outer iteration, *inner iterations* are performed. These inner iterations perform residual calculations to solve the scattering matrix within that group set to some specified tolerance. We will sometimes refer to these inner iterations as *within-group-set* (WGS) *iterations*. Because each outer iteration performs residual calculations for every group set, we will sometimes refer to these outer iteration as *across-group-set* (AGS) *iterations*. We continue performing AGS iterations until the solution residual is below some specified tolerance. At each outer and inner iteration, we possibly perform some acceleration (or preconditioning) step, where we leave the details for this until Chapter **??**. This complete iterative procedure involving the outer and inner iteration loops is given in Algorithm 1. In this algorithm, we solve N_{gs} number of group sets by using a maximum number of AGS iterations, I_{ags}, as well as a maximum number of WGS iterations for each group set, I_{gs}.

As previously stated, the residual iterations for each group set only converge the scattering source for that group set. This means that the scattering sources from the other group sets are not modified during these iterations. We show this behavior by decomposing the transport equation into N_{gs} separate equations where we have decomposed the scattering source into group set and across-group set components. The equations for the angular flux and flux moment solutions for each group set are given by

$$\mathbf{L}_{gs}\boldsymbol{\Psi}_{gs} = \mathbf{M}_{gs}\boldsymbol{\Sigma}_{gs}\boldsymbol{\Phi}_{gs} + \mathbf{M}_{gs}\sum_{ggs\neq gs}\boldsymbol{\Sigma}_{ggs}\boldsymbol{\Phi}_{ggs} + \mathbf{Q}_{gs}, \quad gs = 1, ..., N_{gs} \tag{1.78}$$

and

$$\boldsymbol{\Phi}_{gs} = \mathbf{D}_{gs}\boldsymbol{\Psi}_{gs}, \tag{1.79}$$

respectively, where $\mathbf{L}$ is the fully-discretized loss operator which consists of total interaction and streaming terms, $\mathbf{M}$ is the moment-to-discrete operator of the angular discretization, $\mathbf{D}$ is the discrete-to-moment operator of the angular discretization, $\boldsymbol{\Sigma}$ is the scattering operator of the multigroup and angular discretizations, and $\mathbf{Q}$ is the full phase-space distributed source. In this case, the source contains contributions from boundary, domain sources, and fission sources. We note that $\mathbf{M}$ and $\mathbf{D}$ are also group set dependent since we do not enforce the exact same angular discretization on all group sets. This can be useful if certain ranges of energy groups require higher angular fidelity due to increased anisotropic scattering.

We can further express Eq. (1.78) in terms of only the flux moments. Through algebra and linear algebra techniques, we state the flux-moment-only equation as

$$\left(\mathbf{I} - \mathbf{T}_{gs}\right)\boldsymbol{\Phi}_{gs} = \mathbf{D}_{gs}\mathbf{L}_{gs}^{-1}\mathbf{M}_{gs}\sum_{ggs\neq gs}\boldsymbol{\Sigma}_{ggs}\boldsymbol{\Phi}_{ggs} + \mathbf{D}_{gs}\mathbf{L}_{gs}^{-1}\mathbf{Q}_{gs}, \tag{1.80}$$

where we define,

$$\mathbf{T}_{gs} \equiv \mathbf{D}_{gs}\mathbf{L}_{gs}^{-1}\mathbf{M}_{gs}\boldsymbol{\Sigma}_{gs}, \tag{1.81}$$

for further brevity. In Eqs. (1.80) and (1.81), the term $\mathbf{L}_{gs}^{-1}$ accounts for the inversion of the loss operator. For this work, we will utilize the full-domain transport sweep, and we leave its details until Section 1.6.2. We now provide the details for the procedures that will be used to solve Eq. (1.80) in Sections 1.6.1.1 and 1.6.1.2.

Algorithm 1 Iterative Solver in Energy for the Multigroup Transport Problem

```
Initialize: $\Phi_{g,p,n} = 0$,    $g = 1, ..., G$; $p = 0, ..., N_p$; $n = -p, ..., p$
for $a = 1, ..., I_{ags}$ do
    Loop: through group sets
    for $gs = 1, ..., N_{gs}$ do
        for $k = 1, ..., I_{gs}$ do
            Perform: residual iteration
            Apply: Acceleration for group set $gs$
            Check: convergence of group set $gs$
        end for
    end for
    Perform: residual iteration
    Apply: Acceleration for across-group-set
    Check: across-group-set convergence
end for
```

1.6.1.1 Source Iteration

One simple method to invert the $(\mathbf{I} - \mathbf{T})$ operator of Eq. (1.80) is the *source iteration* technique (SI), also known as *richardson iteration.* If we isolate a single group set, gs, then the iterative procedure for the transport equation is

$$\mathbf{L}\boldsymbol{\Psi}^{(k+1)} = \mathbf{M}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{(k)} + \mathbf{Q}, \tag{1.82}$$

where we removed the gs subscripts for brevity and note that the driving source $\mathbf{Q}$ contains scattering sources from all other group sets into the current group set of interest. From Eq. (1.82), we can see that the scattering source at iteration $(k+1)$ is calculated from a previous guess for the flux moments at iteration (k). Then, after the application of one transport sweep, we obtain a new guess for the angular flux moments:

$$\begin{aligned} \mathbf{\Psi}^{(k+1)} &= \mathbf{L}^{-1}\left(\mathbf{M\Sigma\Phi}^{(k)} + \mathbf{Q}\right) \\ \mathbf{\Phi}^{(k+1)} &= \mathbf{D\Psi}^{(k+1)} \end{aligned} \tag{1.83}$$

We continue to perform these richardson iterations until the difference between two iterate solutions is less than some specified tolerance in a given norm. This convergence criterion for SI can be succinctly written as

$$\frac{||\Phi^{(k+1)} - \Phi^{(k)}||}{||\Phi^{(k+1)}||} \leq \text{tol}, \tag{1.84}$$

where the estimate for the error is normalized by the norm of the most recent iteration solution. Steps are taken to ensure that a divide-by-zero does not occur. In general, SI is guaranteed to converge for a large range of transport problems (further precautions need to be taken for problems with high anisotropic scattering). We can estimate how quickly SI will converge by analyzing the spectral radius (SR) of the transport problem [24]. This spectral radius, ρ, can be estimated in the asymptotic convergence region by the ratio of two successive solution differences:

$$\rho \approx \frac{||\Phi^{(k+1)} - \Phi^{(k)}||}{||\Phi^{(k)} - \Phi^{(k-1)}||} \tag{1.85}$$

We can gather a lot of information about how our transport solution will converge from Eq. (1.85). If $\rho > 1$ in the asymptotic region, then the error in our residual

Table 1.3: Average number of iterations required to reduce SI error by 1 order of magnitude for different SR values.

ρ	Iterations
0.1	1.0
0.25	1.6
0.5	3.3
0.75	8.0
0.9	22
0.99	230
0.999	2301
0.9999	23024

will grow without end, and our solution will diverge. However, in the preasymptotic region, Eq. (1.85) may be a bad estimate for the spectral radius, and values greater than 1 may be observed for several iterations. This does not mean that the solution will diverge in the end. If we are in the asymptotic region, then we would like $\rho \ll 1$ because it means that our transport solution will quickly converge. However, if $\rho \approx 1$ (but still strictly less than 1), then our transport problem will slowly converge to our final solution. We demonstrate this behavior in Table 1.3 by giving the average number of SI iterations required to reduce our solution error by 1 order of magnitude. We can see that as ρ approaches 1, the iteration numbers become prohibitively large. For those problems with large spectral radii, we can accelerate our solution convergence by using synthetic acceleration on Preconditioned Richardson Iteration [24]. We leave the details for this in Chapter **??**.

1.6.1.2 Krylov Subspace Methods

Source iteration is not the only iterative technique that can be employed to invert $(\mathbf{I} - \mathbf{T})$ for a given group set. In the last 20 years, krylov subspace methods have been applied to the discretized transport equation [25, 26, 27]. Because $(\mathbf{I}-\mathbf{T})$ is not symmetric, we want to only use krylov methods that can solve non-symmetric matrices. The two krylov subspace methods that we would naturally want to employ are GMRES and BiCGSTAB [28, 29]. We will not describe the implementations of these methods here for brevity. However, we will state that most of the computational machinery required to perform richardson iterations can also be used for these krylov methods. The only modifications/extensions that are needed are summarized in the following list.

1. Construction of the right-hand-side: $\mathbf{b} = \mathbf{D}\mathbf{L}^{-1}\mathbf{Q}$. From this equation, it is obvious that we need just one initial transport sweep to properly build this right-hand-side.

2. The operation of the matrix $(\mathbf{I}-\mathbf{T})$ on a krylov vector, ν. This can easily be accomplished with the same machinery as richardson iteration by simply subtracting the original flux moment vector by the updated moments after one transport sweep.

3. Modify the calculation of the convergence criterion so that the norm of the iteration residual, normalized by the right-hand-side, is smaller than some prescribed tolerance: $\frac{||\mathbf{b}-(\mathbf{I}-\mathbf{T})\mathbf{x}||_2}{||\mathbf{b}||_2} < \text{tol}$.

Combined with the appropriate linear algebra operations, these three small alterations are all that are required to properly utilize the krylov solver for the transport equation. It is not immediately obvious how one would precondition the krylov iterations in the context of transport sweeping. However, just like the richardson iteration scheme, we will provide the implementation of DSA preconditioning for krylov in Chapter **??**.

1.6.2 Spatial Solution Procedures

Section 1.6.1 presented the methodology that we will employ to iteratively converge our transport solutions in energy and angle (flux moments). Both richardson iteration and the krylov methods were presented as methods that can invert the $(\mathbf{I}-\mathbf{T})$ operator. In both of these iterative methods, the common operation of interest is the inversion of the loss operator $(\mathbf{L})$. There are different techniques that could be used to perform this operation, including several matrix-dependent and matrix-free methodologies. For this work, the loss operator inversion on some unstructured mesh, $\mathbb{T}_h$, will be performed by use of the full-domain transport sweep as outlined next in Section 1.6.2.1. We then conclude our discussion on spatial solution procedures by briefly describing how we can utilize adaptive mesh refinement (AMR) to generate irregular polytope meshes (without hanging nodes in this work) in Section 1.6.2.2.

1.6.2.1 Transport Sweeping

Recall from earlier that the continuous definition of the loss operator for an angular direction m and energy group g is,

$$L_{m,g} = \vec{\Omega}_m \cdot \vec{\nabla} + \sigma_{t,g} \, , \tag{1.86}$$

where we suppress the spatial parameter for brevity. From the application of the test function, integration by parts of the streaming term, and the use of the upwind scheme as outlined in Section 1.5, we described the coupling between different mesh cells. The upwind scheme only couples the adjoining cells upwind of any given angular direction. This means that for a given mesh element, if the upwind cells have already had their angular fluxes updated for a given iteration, we then have all the information needed to solve for the new angular fluxes on the element. This means that if the task dependence graphs are well chosen for all directions, we can then invert the streaming operator by solving Eq. (1.77) one cell at a time for a given group of angular directions.

This cell-by-cell inversion of the streaming operator is known as a *transport sweep*. If we perform this inversion for all angular directions across the entire spatial domain without lagging, then it is called a full-domain transport sweep. The full-domain transport sweep is a beneficial matrix-free scheme because of the following:

- The iterative solver does not need to build any matrices explicitly but only requires the action of $\mathbf{L}^{-1}$.
- Does not require the formation of M separate matrices for each of the angular directions (for 1 energy group). This is both memory and computationaly intensive for any problems of appreciable size.
- The matrix-vector operations on a single element within a transport sweep can be efficiently performed depending on the group set structure and the angle aggregation as outlined in Section 1.6.1.
- The matrix-free transport sweep favors higher-order DGFEM schemes since they will yield more processor work per element with less memory caching (which is the current bottleneck with massively-parallel calculations).
- The number of sweep iterations does not grow with increasing problem size or processor counts. This is in contrast with partial-domain sweeping like *parallel block jacobi* (PBJ) [30].

We can gain knowledge of how our transport sweeps will perform by analyzing the streaming operator, $\mathbf{L}$, in space and angle. If $\mathbf{L}$ is strictly block-lower triangular

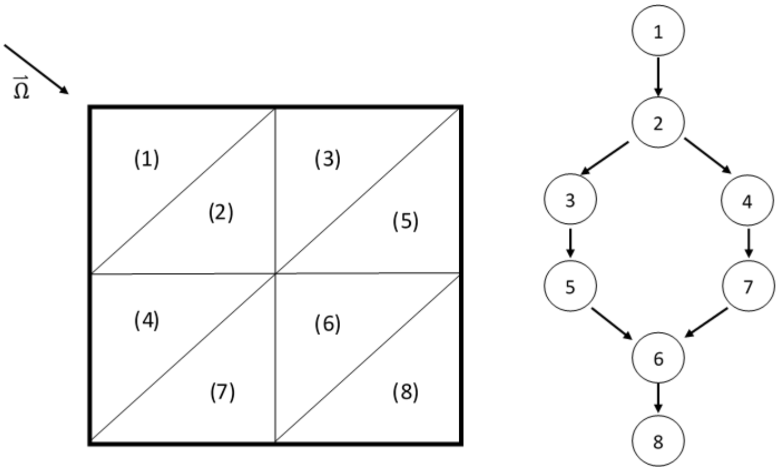


Figure 1.11: Task graph for the transport sweep for a given direction without cycles.

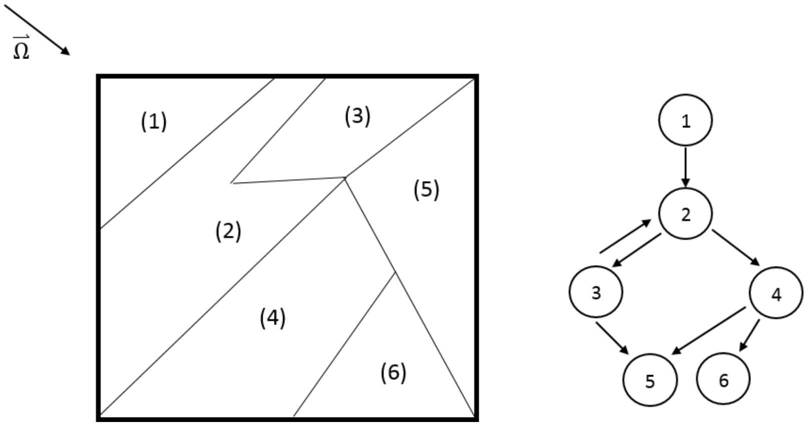


Figure 1.12: Task graph for the transport sweep for a given direction with cycles present.

for a given angular direction, then it is guaranteed that an ordering of the mesh cells can be represented by a Directed Acyclic Graph (DAG) [31]. However, situations can arise where a cycle forms in the graph and a complete sweep for those directions becomes impossible. These situations include concave polytope elements that cause re-entrance, opposing reflecting boundary conditions, and extreme distortion of an $\mathbb{R}^3$ domain (regular mesh of hexahedral cells with 1 dimensional end twisted). Figure 1.11 shows a DAG without any cycles for the given direction. Figure 1.12 then shows a graph with a cycle caused by a re-entrant quadrilateral mesh cell. There are methods to break these cycles and still perform transport sweeping, but we will not consider them for this work.

1.6.2.2 Spatial Adaptive Mesh Refinement

Adaptive Mesh Refinement (AMR) techniques have become more commonplace across many scientific and engineering fields over the last couple of decades [32, 33, 34, 35, 36, 37]. However, while this has become more common practice in other disciplines, the use of AMR for the transport equation is still in its infancy [38, 39, 40, 41, 42, 43]. For this work, we will only utilize *h*-type refinement strategies for 2D transport problems on initial meshes with only quadrilateral cells.

Figure 1.13 provides the logical flow chart used to perform the mesh adaptation procedures for our problem of interest. The AMR procedure begins with an initial and typically coarse mesh. The transport solution is calculated on this mesh with appropriate PDE solvers. Once this solution is determined, *a posteriori* error estimates are used to determine which problem regions contain the largest spatial discretization error [44, 45]. Then, based on some refinement criterion, some subset of mesh elements are flagged for local refinement. From these flagged elements, the mesh can be refined, and a new and more accurate solution can further be obtained.

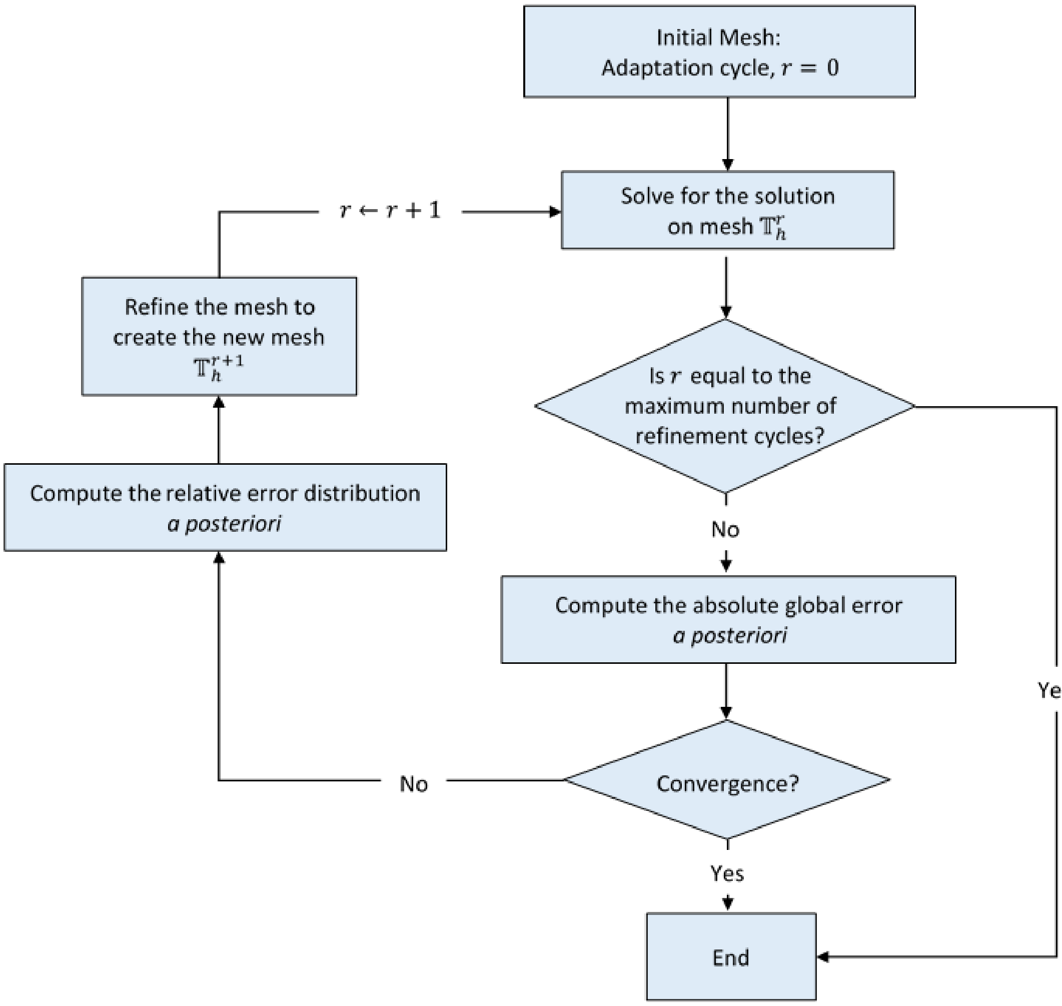

Initial Mesh:
Adaptation cycle, $r = 0$
Solve for the solution on mesh $\mathbb{T}_h^r$
$r \leftarrow r + 1$
Refine the mesh to create the new mesh $\mathbb{T}_h^{r+1}$
Is r equal to the maximum number of refinement cycles?
No
Yes
Compute the relative error distribution *a posteriori*
Compute the absolute global error *a posteriori*
Convergence?
No
Yes
End

Figure 1.13: Flow chart for mesh adaptation.

This process is repeated until some global convergence criterion is satisfied or some number of mesh adaptation cycles are performed.

At the heart of the mesh adaptation procedures is the reliance on an accurate estimation of the numerical solution error. The state-of-the-art error estimators rely on either adjoint-based methods [38, 39, 41] or projection-based methods [46]. However, these methods require an additional calculation of the solution at each refinement level in a richer (and more difficult to solve) functional space. Therefore, we will employ a simpler error estimator that does not require an additional solution calculation. For this work, we will utilize the jump-based error estimator since it is the most straightforward to define and execute. The estimate of the error for mesh cell K of the mesh at refinement level r, $\mathbb{T}_h^r$, is given by the following,

$$\eta_K^r = \int\limits_{\partial K} [\![\Phi^r]\!]^2 d\, s = \int\limits_{\partial K} \left(\sum_m w_m [\![\Psi_m^r]\!] \right)^2 , \tag{1.87}$$

where $[\![\cdot]\!]$ is the jump operator along a face defined as,

$$[\![\Phi(\vec{r})]\!] = \Phi^+(\vec{r}) - \Phi^-(\vec{r}), \tag{1.88}$$

and the terms, $\Phi^+(\vec{r})$ and $\Phi^-(\vec{r})$, are subject to the trace:

$$\Phi^\pm(\vec{r}) = \lim_{s \to 0^\pm} \Phi(\vec{r} + s\vec{n}). \tag{1.89}$$

In this case, the outward normal, $\vec{n}$, is determined with respect to the element K along its boundary, ∂K. With this trace, $\Phi^-(\vec{r})$ always corresponds to the solution within cell K. Investigating face f of cell K, the across-face solution, $\Phi^+(\vec{r})$, is dependent on the boundary type of face f. The across-face solutions can be succinctly written:

$$\Phi^+(\vec{r}) = \begin{cases} \lim_{s \to 0^+} \Phi(\vec{r} + s\vec{n}) & \vec{r} \notin \partial\mathcal{D} \\ \sum_{\vec{\Omega}_m \cdot \vec{n} > 0} \Psi_m^-(\vec{r}) + \sum_{\vec{\Omega}_m \cdot \vec{n} < 0} \Psi_m^{inc}(\vec{r}) & \vec{r} \in \partial\mathcal{D}^d \\ \Phi^-(\vec{r}) & \vec{r} \in \partial\mathcal{D}^r \end{cases} . \tag{1.90}$$

From Eq. (1.90), the across-face solutions for interior faces, incident boundaries and reflecting boundaries have different meanings. For an interior face f ($\vec{r} \notin \partial\mathcal{D}$), the across-face solution comes from the cell K' (as defined by Figure 1.8). For incident boundaries ($\vec{r} \in \partial\mathcal{D}^d$), the across-face solution is a combination of integrals of the outgoing (Ψ_m^-) and incident boundary fluxes (Ψ_m^{inc}). Finally, for reflecting boundaries ($\vec{r} \in \partial\mathcal{D}^r$), the across-face solutions are simply the within-cell solutions. Therefore, the solution jump is exactly zero for all reflecting boundaries and yields no contribution to the error estimate.

With the error estimates defined for all cells $K \in \mathbb{T}_h^r$ for refinement level r, a criterion is needed to determine which cells should be refined. For this work, we choose to employ a refinement criterion of the following form,

$$\eta_K^r \geq \alpha \max_{K' \in \mathbb{T}_h^r} \left(\eta_{K'}^r \right), \tag{1.91}$$

where α is a user-defined value $(0, 1)$. This refinement criterion has a simple meaning. If, for example, $\alpha = 0.2$, then a cell will be refined if its error estimate is greater than 20% of the cell with the largest error estimate. This does not necessarily mean that 80% of the mesh cells will be refined at level r. Instead, the criterion simply states that any cell above a particular threshold will be refined. This means that it is theoretically possible for the extreme cases of 1 or all cells being refined at a particular refinement cycle. The first extreme case could arise with a single spatial cell having a strong solution discontinuity stemming from a strong localized source. The second extreme case could arise when the intercell solution jumps are about equivalent stemming from an incredibly smooth discretized solution at that particular refinement cycle. We could also enforce a uniform refinement of all the spatial cells by setting $\alpha = 0$.

Once we have determined which elements on mesh level r to refine, we then form our new adapted mesh on level $r+1$ by a combination of logical refinement rules, hierarchical elements, and refinement trees. Each quadrilateral mesh cell with vertices flagged for refinement (which we denote as the parent element) is further subdivided into four smaller quadrilateral cells (which we denote as daughter elements I, II, III, and IV) that maintain the overall shape of the orginal mesh cell. If the parent element is denoted by the vertices (1), (2), (3), and (4), then the refinement rules, including the new local ordering of each daughter element's vertices are given in Figure 1.14. These refinement rules are detailed as follows:

1. Daughter element I is placed near original vertex (1).
2. Daughter element II is placed near original vertex (2).
3. Daughter element III is placed near original vertex (3).
4. Daughter element IV is placed near original vertex (4).
5. The local numbering of the daughter element vertices is consistent with the parent element.
6. The vertices of the daughter elements common with the original element inherit the same local numbering.

These refinement rules guarantee that we will maintain logical consistency with our counter-clockwise ordering of the element vertices. This consistency is important for the computation of the polytope basis functions that are presented in Chapter **??**.

Once a mesh element from level r has been refined, we need to incorporate its daughter elements into the adapted mesh, $\mathbb{T}_h^{r+1}$. We can see from the nesting refinement process, that these AMR procedures lead to a hierarchical representation of the mesh. All of the mesh cells are obtained by subdivisions of cells from the initial mesh, $\mathbb{T}_h^0$. This hierarchical nature of the mesh can succinctly be described in a tree structure. We give an example of this tree structure in Figure 1.15 for the simple case

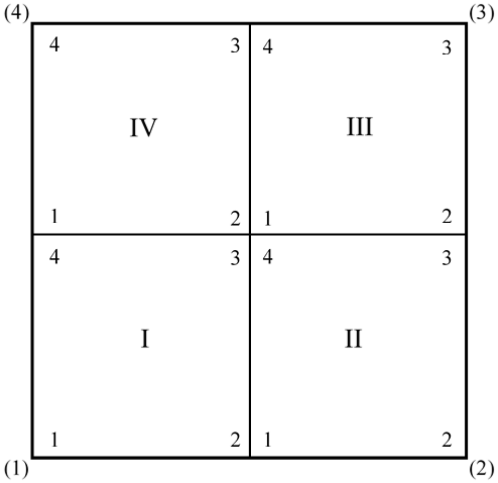
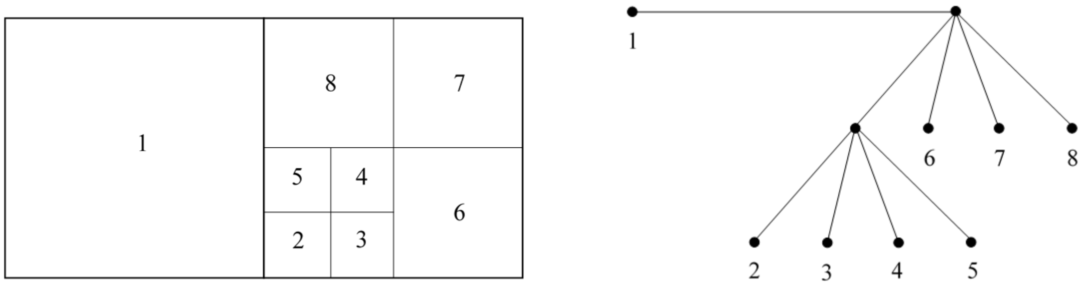


Figure 1.14: Refinement rules for a quadrilateral mesh cell.

of a 2-cell domain that undergoes two refinement cycles with 1 cell refined per cycle. From the tree structure in Figure 1.15b, we introduce the concept of an individual element's *refinement level*, $\ell(K)$, for mesh cell K. The refinement level $\ell(K)$ denotes how many times a cell from the original mesh, $\mathbb{T}_h^0$, has been refined to form element K. By convention, the refinement level for all elements in the original mesh are 0. We note that an element's refinement level is not necessarily the number of refinement cycles that have been performed. In Figure 1.15b, we can see three distinct refinement levels that are represented by the three different vertical levels of nodes. Element 1 has a level of 0 since it was never refined. Elements 2, 3, 4, and 5 have a level of 2. Finally, elements 6, 7, and 8 have a level of 1. We also use Figure 1.15 to further define the concept of an *irregularity index*. This index is the difference in refinement levels between two adjoining mesh elements. The irregularity index for the face separating elements 6 and 7 is 0 since $\ell(6) = \ell(7)$. The irregularity index for the face separating elements 4 and 8 is 1 since $\ell(8) = 1$ and $\ell(4) = 2$. Finally, the irregularity index for the face separating elements 1 and 2 is 2 since $\ell(1) = 0$ and



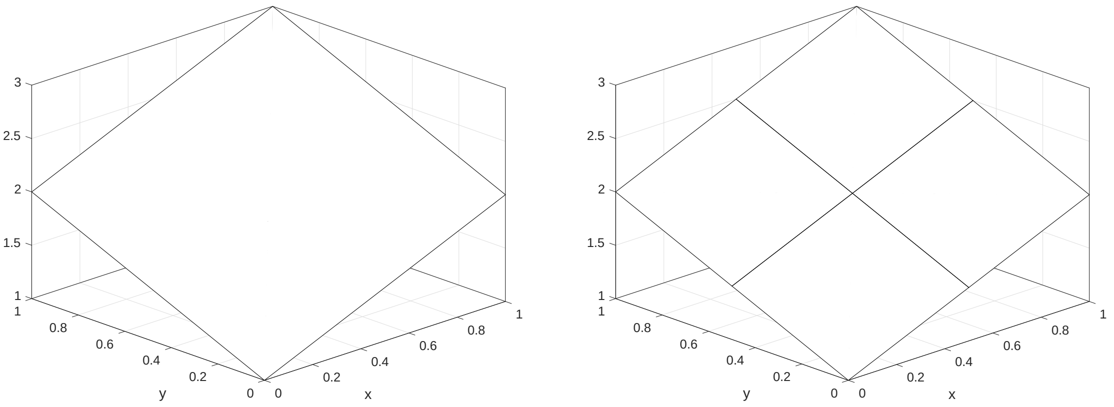
(a) Twice-refined domain

(b) Refinement tree structure

Figure 1.15: Hierarchical refinement tree for a simple domain with quadrilateral cells.

$\ell(2) = 2$. For the AMR problems in this work, we restrict the maximum allowable irregularity index for all of the elements for each of the mesh refinement levels.

Following a refinement cycle, we have a new adapted mesh, $\mathbb{T}_h^{r+1}$, but our current solution lives on $\mathbb{T}_h^r$. This means that the solution lives on a lower-dimensional space than the adapted mesh. To solve for the transport solution on $\mathbb{T}_h^{r+1}$, we could simply reinitialize our solution vectors to zero and follow the procedures in Algorithm 1. However, this means that the solution from mesh $\mathbb{T}_h^r$ would simply be discarded. This would discount all the work performed to compute the solution on level r. Furthermore, if there is little difference in the functional space between the two refinement cycles, then the calculated solution at level r would be a good initial guess for the solution to be calculated for level $r + 1$. For this reason, we define the concept of *bootstrapping*, which is the projection of the solution from $\mathbb{T}_h^r$ onto $\mathbb{T}_h^{r+1}$. Figure 1.16 provides an example of bootstrapping by refining a single unit square element onto the four identical daughter elements with $\frac{1}{4}$ the area as the parent element. Figure 1.16a shows a linear planar solution on a single unit square parent element. Then, after this element is refined, its solution is projected onto its four identical daughter elements. Figure 1.16b shows this bootstrapped solution onto the refined mesh, where we can clearly see that the projected solution lives in



(a) Before refinement

(b) After refinement

Figure 1.16: Bootstrapping the solution from a single unit square element onto the four daughter elements after refinement.

the dimensional space of the daughter elements but still lives in the functional space of the parent element.

1.7 Conclusions

In this chapter, we have presented the tightly-coupled system of equations that comprise the DGFEM S_N transport equation. We began with the fully-continuous transport equation presented in Section 1.1 and then discretized it in energy, angle and space in Sections 1.2, 1.3, and 1.5, respectively. Appropriate boundary conditions were presented in Section 1.4. For this work, we will only utilize incoming-incident and reflecting boundary conditions and not use any further albedo terms. We finished this chapter in Section 1.6 by describing the procedures that will be utilized to solve our system of equations in energy, angle, and space. We also provided the methodology that we will employ to perform AMR calculations in this work, which yields another mechanism to produce polytope meshes.

REFERENCES

[1] R. LERNER and G. TRIGG, *Encyclopaedia of Physics*, 2 ed. (1991).

[2] C. PARKER, *McGraw Hill Encyclopaedia of Physics*, 2 ed. (1994).

[3] J. J. DUDERSTADT and W. R. MARTIN, *Transport theory*, John Wiley & Sons (1979).

[4] K. OTT and W. BEZELLA, *Introductory Nuclear Reactor Statics*, American Nuclear Society (1989).

[5] J. J. DUDERSTADT and L. J. HAMILTON, *Nuclear reactor analysis*, Wiley (1976).

[6] E. E. LEWIS and W. F. MILLER, *Computational methods of neutron transport*, John Wiley and Sons, Inc., New York, NY (1984).

[7] B. CARLSON and K. LATHROP, *Computing methods in reactor physics*, Gordon and Breach Science Publishers, Inc. (1968).

[8] B. CARLSON, "On a more precise definition of discrete ordinates methods," in "Proc. Second Conf. Transport Theory," U.S. Atomic Energy Commision (1971), pp. 348–390.

[9] I. ABU-SHUMAYS, "Compatible product angular quadrature for neutron transport in xy geometry," *Nuclear Science and Engineering*, **64**, *2*, 299–316 (1977).

[10] M. ABRAMOWITZ, I. A. STEGUN, ET AL., "Handbook of mathematical functions," *Applied Mathematics Series*, **55**, 62 (1966).

[11] M. ABRAMOWITZ and I. A. STEGUN, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, 55, Courier Corporation (1964).

[12] A. ERN and J.-L. GUERMOND, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media (2013).

[13] T. A. WAREING, J. M. MCGHEE, J. E. MOREL, and S. D. PAUTZ, "Discontinuous finite element S_N methods on three-dimensional unstructured grids," *Nuclear science and engineering*, **138**, *3*, 256–268 (2001).

[14] P. LESAINT and P.-A. RAVIART, "On a finite element method for solving the neutron transport equation," *Mathematical aspects of finite elements in partial differential equations*, , *33*, 89–123 (1974).

[15] P. HOUSTON, C. SCHWAB, and E. SÜLI, "Stabilized hp-finite element methods for first-order hyperbolic problems," *SIAM Journal on Numerical Analysis*, **37**, *5*, 1618–1643 (2000).

[16] P. HOUSTON, C. SCHWAB, and E. SÜLI, "Discontinuous hp-finite element methods for advection-diffusion-reaction problems," *SIAM Journal on Numerical Analysis*, **39**, *6*, 2133–2163 (2002).

[17] Y. WANG and J. C. RAGUSA, "On the convergence of DGFEM applied to the discrete ordinates transport equation for structured and unstructured triangular meshes," *Nuclear Science and Engineering*, **163**, *1*, 56–72 (2009).

[18] E. MAGENES and J.-L. LIONS, *Problèmes aux limites non homogènes et applications* (1968).

[19] C. JOHNSON and J. PITKÄRANTA, "Convergence of a fully discrete scheme for two-dimensional neutron transport," *SIAM journal on numerical analysis*, **20**, *5*, 951–966 (1983).

[20] R. H. MACNEAL and R. L. HARDER, "Eight nodes or nine?" *International journal for numerical methods in engineering*, **33**, *5*, 1049–1058 (1992).

[21] D. N. ARNOLD and G. AWANOU, “The serendipity family of finite elements,” *Foundations of Computational Mathematics*, **11**, *3*, 337–344 (2011).

[22] O. ZEINKIEWICZ, R. TAYLOR, and J. ZHU, *The finite element method: its basis and fundamentals*, SElsevier Butterworth-Heinemann (2005).

[23] J. AKIN, *Application and implementation of finite element methods*, Academic Press, Inc. (1982).

[24] M. ADAMS and E. LARSEN, “Fast iterative methods for discrete-ordinates particle transport calculations,” *Progress in nuclear energy*, **40**, *1*, 3–159 (2002).

[25] S. OLIVEIRA and Y. DENG, “Preconditioned Krylov subspace methods for transport equations,” *Progress in Nuclear Energy*, **33**, *1*, 155–174 (1998).

[26] B. GUTHRIE, J. P. HOLLOWAY, and B. W. PATTON, “GMRES as a multi-step transport sweep accelerator,” *Transport theory and statistical physics*, **28**, *1*, 83–102 (1999).

[27] B. W. PATTON and J. P. HOLLOWAY, “Application of preconditioned GMRES to the numerical solution of the neutron transport equation,” *Annals of Nuclear Energy*, **29**, *2*, 109–136 (2002).

[28] Y. SAAD and M. H. SCHULTZ, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on scientific and statistical computing*, **7**, *3*, 856–869 (1986).

[29] Y. SAAD, *Iterative methods for sparse linear systems*, Siam (2003).

[30] R. J. ZERR, *Solution of the within-group multidimensional discrete ordinates transport equations on massively parallel architectures*, Ph.D. thesis, The Pennsylvania State University (2011).

[31] A. GERASOULIS and I. NELKEN, "Static scheduling for linear algebra DAGs," in "Proc. of 4th Conf. on Hypercubes, Monterey," (1989), vol. 1, pp. 671–674.

[32] T. PLEWA, T. LINDE, and V. G. WEIRS, *Adaptive Mesh Refinement - Theory and Applications*, Springer Science & Business Media (2005).

[33] G. F. CAREY, *Computational Grids: Generations, Adaptation & Solution Strategies*, CRC Press (1997).

[34] E. RAMM, E. RANK, R. RANNACHER, K. SCHWEIZERHOF, E. STEIN, W. WENDLAND, G. WITTUM, P. WRIGGERS, and W. WUNDERLICH, *Error-controlled adaptive finite elements in solid mechanics*, John Wiley & Sons (2003).

[35] G. KARNIADAKIS and S. SHERWIN, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press (2013).

[36] C. SCHWAB, *p-and hp-finite element methods: Theory and applications in solid and fluid mechanics*, Oxford University Press (1998).

[37] P. SOLIN, K. SEGETH, and I. DOLEZEL, *Higher-order finite element methods*, CRC Press (2003).

[38] C. FÜHRER and G. KANSCHAT, "A posteriori error control in radiative transfer," *Computing*, **58**, *4*, 317–334 (1997).

[39] R. HARTMANN, *Adaptive finite element methods for the compressible euler equations*, Ph.D. thesis, University of Heidelberg (2002).

[40] A. DEDNER and P. VOLLMÖLLER, "An adaptive higher order method for solving the radiation transport equation on unstructured grids," *Journal of Computational Physics*, **178**, *2*, 263–289 (2002).

[41] R. HARTMANN and P. HOUSTON, “Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws,” *SIAM Journal on Scientific Computing*, **24**, *3*, 979–1004 (2003).

[42] Y. WANG and J. C. RAGUSA, “A two-mesh adaptive mesh refinement technique for SN neutral-particle transport using a higher-order DGFEM,” *Journal of computational and applied mathematics*, **233**, *12*, 3178–3188 (2010).

[43] Y. WANG and J. C. RAGUSA, “Standard and goal-oriented adaptive mesh refinement applied to radiation transport on 2D unstructured triangular meshes,” *Journal of Computational Physics*, **230**, *3*, 763–788 (2011).

[44] R. VERFÜRTH, *A review of a posteriori error estimation and adaptive mesh-refinement techniques*, John Wiley & Sons Inc (1996).

[45] M. AINSWORTH and J. T. ODEN, *A posteriori error estimation in finite element analysis*, vol. 37, John Wiley & Sons (2011).

[46] L. DEMKOWICZ, *Computing with hp-Adaptive Finite Elements: Volume 1 One and Two Dimensional Elliptic and Maxwell problems*, CRC Press (2006).