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

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Current State of the Problem

1.2 Motivation and Purpose of the Dissertation

1.3 Organization of the Dissertation

In this introductory chapter, we have presented a summary of work performed. We also gave our motivation for choosing this work as well as a brief discussion of previous work that has directly influenced this dissertation. We conclude this intoduction by briefly describing the remaining chapters of this dissertation.

In Chapter 2, we present the DGFEM formulation for the multigroup, S_N transport equation. We then describe the transport equation's discretization in energy, angle, and space. We have left the FEM spatial interpolation function as arbitrary at this point to be defined in detail in Chapter 3. For the spatial variable, we provide the theoretical convergence properties of the DGFEM form. We also detail the elementary assembly procedures to form the full set of spatial equations. We conclude by providing the methodology to be used to solve the full phase-space of the transport problem.

In Chapter 3, we present all the finite element basis functions that we will use in this work. In two dimensions, we present four different linearly-complete polygonal coordinate systems that we will use to generate our finite element basis functions. We then present the methodology that converts each of these linear coordinate systems into quadratically-complete coordinates for use as higher-order basis functions. We also present the single linearly-complete polyhedral coordinate system that we will use for the 3D transport problems.

In Chapter 4, we present the methodologies to be used for DSA preconditioning of the DGFEM transport equation for optically thick problems. We give a discontinuous form of the diffusion equation which can be used on 2D and 3D polytope grids. The theoretical limits of the DSA scheme are analyzed and we conclude with a real-world problem of accelerating the thermal neutron upscattering of a large multigroup, heterogeneous transport problem. In doing so, we demonstrate that our methodology will work on massively-parallel computer architectures.

We then finalize this dissertation work by drawing conclusions and discussing open topics of research stemming from this dissertation in Chapter **??**. We note that our detailed literature reviews, numerical results, and conclusions pertaining to each topic are presented in their corresponding chapter.

Additional material that is not included in the main body of the dissertation for the sake of brevity is appended for completeness. The appendices are organized in a simple manner:

- Appendix **??**: addendum to Section 2, corresponding to additional material relating to the multigroup S_N equations.

- Appendix **??**: addendum to Section 3, corresponding to additional material relating to the various polytope coordinate systems to be utilized as finite element basis functions.

- Appendix **??**: addendum to Section 4, corresponding to additional material relating to DSA preconditioning on polytope grids.

2. 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:

$$\frac{\partial u}{\partial t} = \left(\frac{\partial u}{\partial t}\right)_{force} + \left(\frac{\partial u}{\partial t}\right)_{advec} + \left(\frac{\partial u}{\partial t}\right)_{coll} \tag{2.1}$$

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. (2.1) 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 2.1 provides the general form of the neutron transport equation with some variants. Section 2.2 describes how we discretize the transport equation in energy with the multigroup methodology and Section 2.3 presents the angular discretization via collocation. Section 2.4 details which boundary conditions will be employed for our work. Section 2.5 will conclude our discretization procedures in the spatial domain. Section 2.6 will present the iterative procedures used to converge our solution space. We then present concluding remarks for the chapter in Section 2.7.

2.1 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:

$$\begin{aligned}\frac{\partial\Psi}{\partial t} + \vec{\Omega}\cdot\vec{\nabla}\Psi(\vec{r}, E, \vec{\Omega}, t) + \sigma_t(\vec{r}, E, t)\Psi(\vec{r}, E, \vec{\Omega}, t) = Q_{ext}(\vec{r}, E, \vec{\Omega}, t)\\ + \frac{\chi(\vec{r}, E, t)}{4\pi}\int dE'\nu\sigma_f(\vec{r}, E', t)\int d\Omega'\Psi(\vec{r}, E', \vec{\Omega}', t)\\ + \int dE'\int d\Omega'\sigma_s(E' \to E, \Omega' \to \Omega)\Psi(\vec{r}, E', \vec{\Omega}')\end{aligned} \tag{2.2}$$

with the following, general boundary condition:

$$\Psi(\vec{r},E,\vec{\Omega},t) = \Psi^{inc}(\vec{r},E,\vec{\Omega},t) + \int dE' \int d\Omega' \gamma(\vec{r},E' \to E,\vec{\Omega}' \to \vec{\Omega},t)\Psi(\vec{r},E',\vec{\Omega}',t) \\ \text{for } \vec{r} \in \partial\mathcal{D}^- \left\{\partial\mathcal{D}, \vec{\Omega}' \cdot \vec{n} < 0\right\}. \tag{2.3}$$

In Eqs. (2.2) and (2.3), 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. (2.2) to:

$$\frac{\partial\Psi}{\partial t} + \mathbf{L}\Psi = \mathbf{F}\Psi + \mathbf{S}\Psi + \mathbf{Q}, \tag{2.4}$$

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

$$\begin{aligned}
\mathbf{L}\Psi &= \vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{r},E,\vec{\Omega},t) + \sigma_t(\vec{r},E,t)\Psi(\vec{r},E,\vec{\Omega},t), \\
\mathbf{F}\Psi &= \frac{\chi(\vec{r},E,t)}{4\pi} \int dE' \nu\sigma_f(\vec{r},E',t) \int d\Omega' \Psi(\vec{r},E',\vec{\Omega}',t), \\
\mathbf{S}\Psi &= \int dE' \int d\Omega' \sigma_s(E' \to E, \Omega' \to \Omega, t)\Psi(\vec{r},E',\vec{\Omega}',t), \\
\mathbf{Q} &= Q_{ext}(\vec{r},E,\vec{\Omega},t),
\end{aligned} \tag{2.5}$$

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

$$\mathbf{L}\Psi = \mathbf{F}\Psi + \mathbf{S}\Psi + \mathbf{Q}, \tag{2.6}$$

and note that the operators of Eq. (2.5) 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. (2.2), $\nu(\vec{r}, E)$ acts as a multiplicative factor on the number of neutrons emitted per fission event. We replace this multiplicative factor in the following manner:

$$\nu(\vec{r}, E) \rightarrow \frac{\tilde{\nu}(\vec{r}, E)}{k}, \tag{2.7}$$

where we have introduced the eigenvalue, k. By also dropping the external source term, the steady-state neutron transport equation in Eq. (2.6) can be rewritten into

$$(\mathbf{L} - \mathbf{S})\, \tilde{\Psi} = \frac{1}{k}\mathbf{F}\tilde{\Psi}, \tag{2.8}$$

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. (2.8) 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].

2.2 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:

$$\Delta E_g \in [E_g, E_{g-1}]. \tag{2.9}$$

Figure 2.1 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. (2.6). The time-dependent and eigenvalue forms are analagous and would be derived identically. Taking the energy interval for group g as defined in Eq. (2.9), the energy-integrated angular flux of group g is

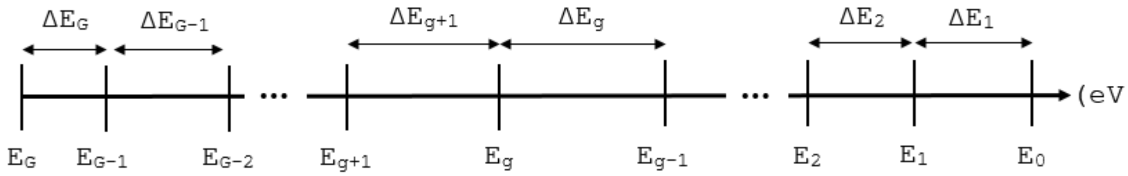


Figure 2.1: Interval structure of the multigroup methodology.

$$\Psi_g(\vec{r},\vec{\Omega}) = \int_{E_g}^{E_{g-1}} \Psi(\vec{r},E,\vec{\Omega})\,dE. \tag{2.10}$$

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):

$$\left(\vec{\Omega}\cdot\vec{\nabla} + \sigma_{t,g}\right)\Psi_g = \sum_{g'=1}^{G}\left[\frac{\chi_g}{4\pi}\nu\sigma_{f,g'}\int_{4\pi}\Psi_{g'}(\vec{\Omega}')\,d\Omega' + \int_{4\pi}\sigma_s^{g'\to g}(\vec{\Omega}',\vec{\Omega})\Psi_{g'}(\vec{\Omega}')\,d\Omega'\right] + Q_g \tag{2.11}$$

where

$$\begin{aligned}
\sigma_{t,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}}\sigma_t(\vec{r},\vec{\Omega},E)\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}{\int_{E_g}^{E_{g-1}}\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}\\
\nu\sigma_{f,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}}\nu\sigma_f(\vec{r},E)\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}{\int_{E_g}^{E_{g-1}}\int_{4\pi}\Psi(\vec{r},\vec{\Omega},E)\,dEd\Omega}\\
\chi_g &\equiv \int_{E_g}^{E_{g-1}}\chi(\vec{r},E)\,dE\\
\sigma_s^{g'\to g}(\vec{r},\vec{\Omega}',\vec{\Omega}) &\equiv \frac{\int_{E_{g'}}^{E_{g'-1}}\left[\int_{E_g}^{E_{g-1}}\sigma_s(\vec{r},E'\to E,\vec{\Omega}',\vec{\Omega})\,dE\right]\Psi(\vec{r},\vec{\Omega}',E')\,dE'}{\int_{E_g}^{E_{g-1}}\Psi(\vec{r},\vec{\Omega},E)\,dE}\\
Q_g(\vec{r},\vec{\Omega}) &\equiv \int_{E_g}^{E_{g-1}}Q(\vec{r},\vec{\Omega},E)\,dE
\end{aligned} \tag{2.12}$$

The above equations are mathematically exact to those presented in Eqs. (2.2 - 2.6) 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:

$$\Psi(\vec{r}, \vec{\Omega}, E) = \hat{\Psi}(\vec{r}, \vec{\Omega}) f_s(E), \tag{2.13}$$

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. (2.12):

$$\begin{aligned}
\sigma_{t,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}} \sigma_t(\vec{r}, E) f_s(E)\, dE}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}, \\
\nu\sigma_{f,g}(\vec{r}) &\equiv \frac{\int_{E_g}^{E_{g-1}} \nu\sigma_f(\vec{r}, E) f_s(E)\, dE}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}, \\
\sigma_s^{g'\to g}(\vec{r}, \vec{\Omega}', \vec{\Omega}) &\equiv \frac{\int_{E_{g'}}^{E_{g'-1}} \left[\int_{E_g}^{E_{g-1}} \sigma_s(\vec{r}, E' \to E, \vec{\Omega}', \vec{\Omega})\, dE\right] f_s(E')\, dE'}{\int_{E_g}^{E_{g-1}} f_s(E)\, dE}.
\end{aligned} \tag{2.14}$$

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.

2.3 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 stready-state equation presented in Eq. (2.6), dropping spatial parameterization, combining the fission and external sources into a single term, and using only 1 energy group:

$$\vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{\Omega}) + \sigma_t \Psi(\vec{\Omega}) = \int_{4\pi} d\Omega' \, \sigma_s(\vec{\Omega}' \to \vec{\Omega})\Psi(\vec{\Omega}') + Q(\vec{\Omega}). \tag{2.15}$$

We first develop an approximation for the scattering term in Eq. (2.15) 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,

$$\sigma_s(\vec{\Omega}' \to \vec{\Omega}) = \frac{1}{2\pi}\sigma_s(\vec{\Omega}' \cdot \vec{\Omega}) = \frac{1}{2\pi}\sigma_s(\mu_0), \tag{2.16}$$

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,

$$\sigma_s(\vec{\Omega}' \to \vec{\Omega}) = \sum_{p=0}^{\infty} \frac{2p+1}{4\pi} \sigma_{s,p} P_p(\mu_0), \tag{2.17}$$

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:

$$\sigma_{s,p} \equiv \int_{-1}^{1} d\mu_0 \, \sigma_s(\mu_0) P_p(\mu_0). \tag{2.18}$$

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,

$$\Psi(\vec{\Omega}) = \frac{1}{4\pi} \sum_{k=0}^{\infty} \sum_{n=-k}^{k} \Phi_{k,n} Y_{k,n}(\vec{\Omega}) \tag{2.19}$$

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

$$\Phi_{k,n} \equiv \int_{4\pi} d\Omega \, \Psi(\vec{\Omega}) \, Y_{k,n}(\vec{\Omega}). \tag{2.20}$$

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. (2.17) and the angular flux expansion of Eq. (2.19), and insert them into the original scattering term of the right-hand-side of Eq. (2.15). 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. (2.15) can now be written again with this alternate and simplified scattering term that is composed of the cross section and angular flux moments:

$$\vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{\Omega}) + \sigma_t \Psi(\vec{\Omega}) = \sum_{p=0}^{\infty} \frac{2p+1}{4\pi} \sigma_{s,p} \sum_{n=-p}^{p} \Phi_{p,n} Y_{p,n}(\vec{\Omega}) + Q(\vec{\Omega}). \tag{2.21}$$

From the initial assumption of material isotropy (which may or not be an approximation), the scattering term of Eq. (2.21) 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. (2.21), but also fold the source term into the spherical harmonics expansion,

$$\vec{\Omega} \cdot \vec{\nabla}\Psi(\vec{\Omega}) + \sigma_t \Psi(\vec{\Omega}) = \sum_{p=0}^{N_p} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}) \left[\sigma_{s,p} \Phi_{p,n} + Q_{p,n} \right]. \tag{2.22}$$

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 ($\vec{\Omega}' \rightarrow \vec{\Omega}$). For every group-to-group combination in energy ($g' \rightarrow g$), there are only the N_p moments of the scattering cross section. Secondly, the contribution of the angular flux into the

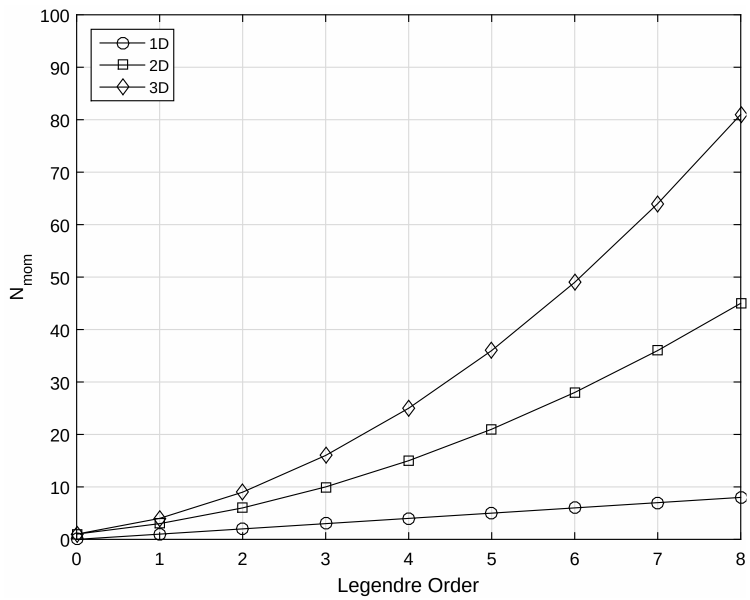


Figure 2.2: Number of spherical harmonics moments, N_{mom}, in 1D, 2D, and 3D as a function of the expansion order, p.

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 2.6.1, 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 2.2.

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,

$$\Psi(\vec{\Omega}) = \sum_{m=1}^{M} B_m(\vec{\Omega})\Psi_m, \tag{2.23}$$

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

$$B_j(\vec{\Omega}_m) = \delta_{j,m}, \tag{2.24}$$

as well as the *Lagrange* property,

$$\sum_{m=1}^{M} B_m(\vec{\Omega}_m) = 1, \tag{2.25}$$

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

$$\Psi_m = \Psi(\vec{\Omega}_m). \tag{2.26}$$

Next, we substitute Eq. (2.23) into Eq. (2.22), drop the external source for clarity, and collocate at the $(k = 1, ..., M)$ interpolation (quadrature) points,

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} \left(\sum_{m=1}^{M} B_m(\vec{\Omega}_k)\Psi_m \right) + \sigma_t \left(\sum_{m=1}^{M} B_m(\vec{\Omega}_k)\Psi_m \right) \\ = \sum_{p=0}^{N_p} \frac{2p+1}{4\pi}\sigma_{s,p} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}) \left(\sum_{m=1}^{M} \Psi_m \int_{4\pi} d\Omega\, B_m(\vec{\Omega}_k)\, Y_{p,n}(\vec{\Omega}) \right), \\ k = 1, ..., M \end{aligned} \tag{2.27}$$

where we inserted Eq. (2.23) into Eq. (2.20) to form a slightly modified form for the angular flux moments:

$$\Phi_{p,n} = \sum_{m=1}^{M} \Psi_m \int_{4\pi} d\Omega\, B_m(\vec{\Omega})\, Y_{p,n}(\vec{\Omega}). \tag{2.28}$$

The *Kronecker* property of Eq. (2.24), is then used at the collocation points so that Eq. (2.27) can be simplified into the following form (where we have reintroduced the distributed source term in terms of its contribution for angle m):

$$\begin{aligned} \vec{\Omega}_m \cdot \vec{\nabla}\Psi_m + \sigma_t \Psi_m = \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sigma_{s,p} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m)\Phi_{p,n} + Q_m \\ m = 1, ..., M \end{aligned} . \tag{2.29}$$

Equation (2.29) 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 2.4.

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 2.1 presents the mapping from the primary quadrant to the other 3 quadrants for 2D problems. Table 2.2 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 2.1: 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 2.2: 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 2.3.1 presents the Level Symmetric (LS) quadrature set and Section 2.3.2 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.

2.3.1 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 2.3. The

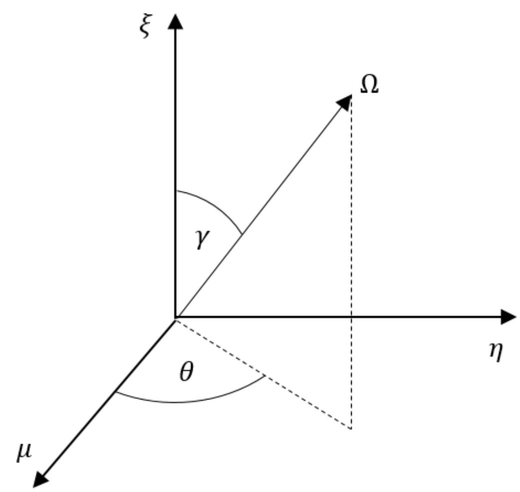


Figure 2.3: Angular coordinate system for the direction $\vec{\Omega}$.

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:

$$\begin{aligned} \vec{\Omega}_x =& \mu = \cos(\theta)\sin(\gamma) = \cos(\theta)\sqrt{1-\xi^2} \\ \vec{\Omega}_y =& \eta = \sin(\theta)\sin(\gamma) = \sin(\theta)\sqrt{1-\xi^2}\,. \\ \vec{\Omega}_z =& \xi = \cos(\gamma) \end{aligned} \tag{2.30}$$

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

$$\mu^2 + \eta^2 + \xi^2 = 1. \tag{2.31}$$

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:

$$\mu_i = \eta_i = \xi_i, \qquad i \in (1, N/2) \tag{2.32}$$

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. (2.31), 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):

$$\mu_i^2 = \mu_{i-1}^2 + \Delta \tag{2.33}$$

where the spacing constant, Δ, has the form:

$$\Delta = \frac{2(1 - 3\mu_1^2)}{N - 2}. \tag{2.34}$$

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 $N \geq 20$.

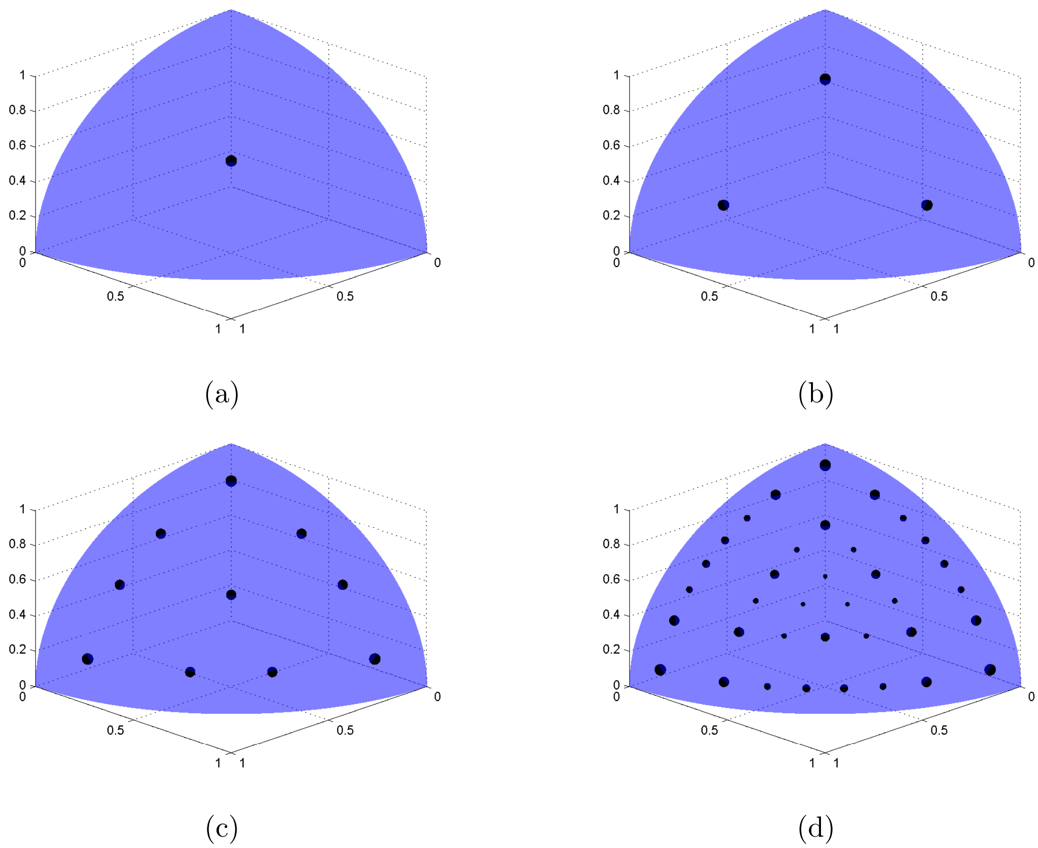


Figure 2.4: Level-Symmetric angular quadrature sets of order (a) 2, (b) 4, (c) 8, and (d) 16.

We conclude this discussion of the LS quadrature set with some examples. Figure 2.4 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 2.5 then provides the projection of the 3D LS quadrature set onto to 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.

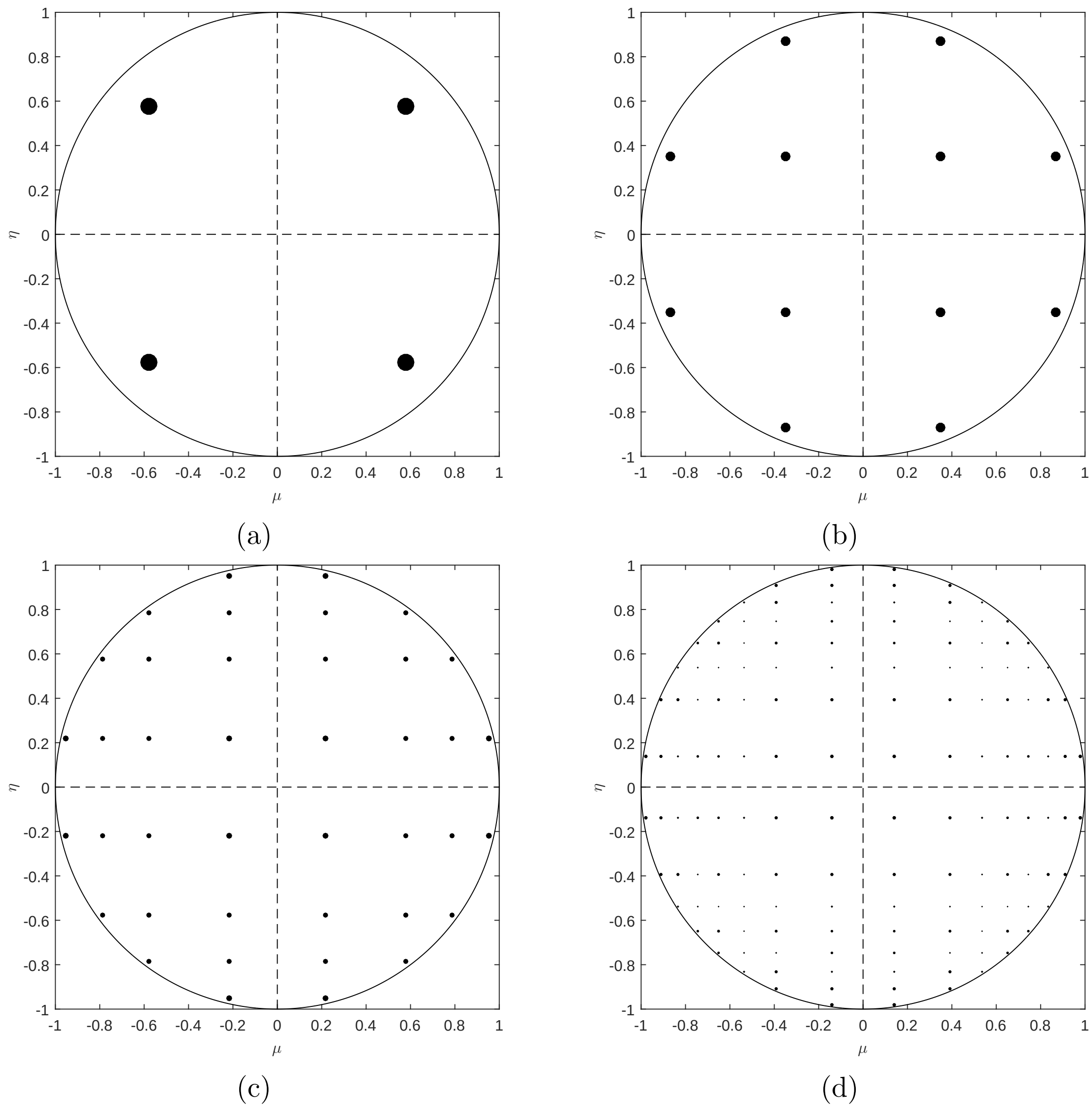


Figure 2.5: Projection of the 3D Level-Symmetric angular quadrature set with orders (a) 2, (b) 4, (c) 8, and (d) 16 onto the x-y space on the unit circle.

2.3.2 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

$$\theta_m = \frac{2m-1}{4A}\pi \qquad \text{and} \qquad w_m = \frac{\pi}{2A}, \tag{2.35}$$

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,

$$w_n = \frac{2}{(1-\xi_n^2)(L'_{2P}(\xi_n))^2}, \tag{2.36}$$

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

With the azimuthal directions specified by Eq. (2.35) 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. (2.30). The ordinate weights can be specified in a similar manner. From Eqs. (2.35) and (2.36), 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,

$$F = \sum_{m=1}^{A} \sum_{n=1}^{P} w_m w_n f(\theta_m, \gamma_n). \tag{2.37}$$

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 2.6 for the primary octant with several combinations of azimuthal and polar directions. Figure 2.7 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.

2.4 Boundary Conditions

Using the energy and angular discretizations presented in Sections 2.2 and 2.3, respectively, we write the standard, steady-state, multigroup S_N transport equation for one angular direction, m, and one energy group, g:

$$\begin{aligned} \left(\vec{\Omega}_m \cdot \vec{\nabla} + \sigma_{t,g}\right) \Psi_{m,g} = \sum_{g'=1}^{G} \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sigma_{s,p}^{g' \to g} \sum_{n=-p}^{p} \Phi_{p,n,g'} Y_{p,n}(\vec{\Omega}_m) \\ + \frac{\chi_g}{4\pi} \sum_{g'=1}^{G} \nu \sigma_{f,g'} \Phi_{g'} + Q_{m,g} \end{aligned}, \tag{2.38}$$

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

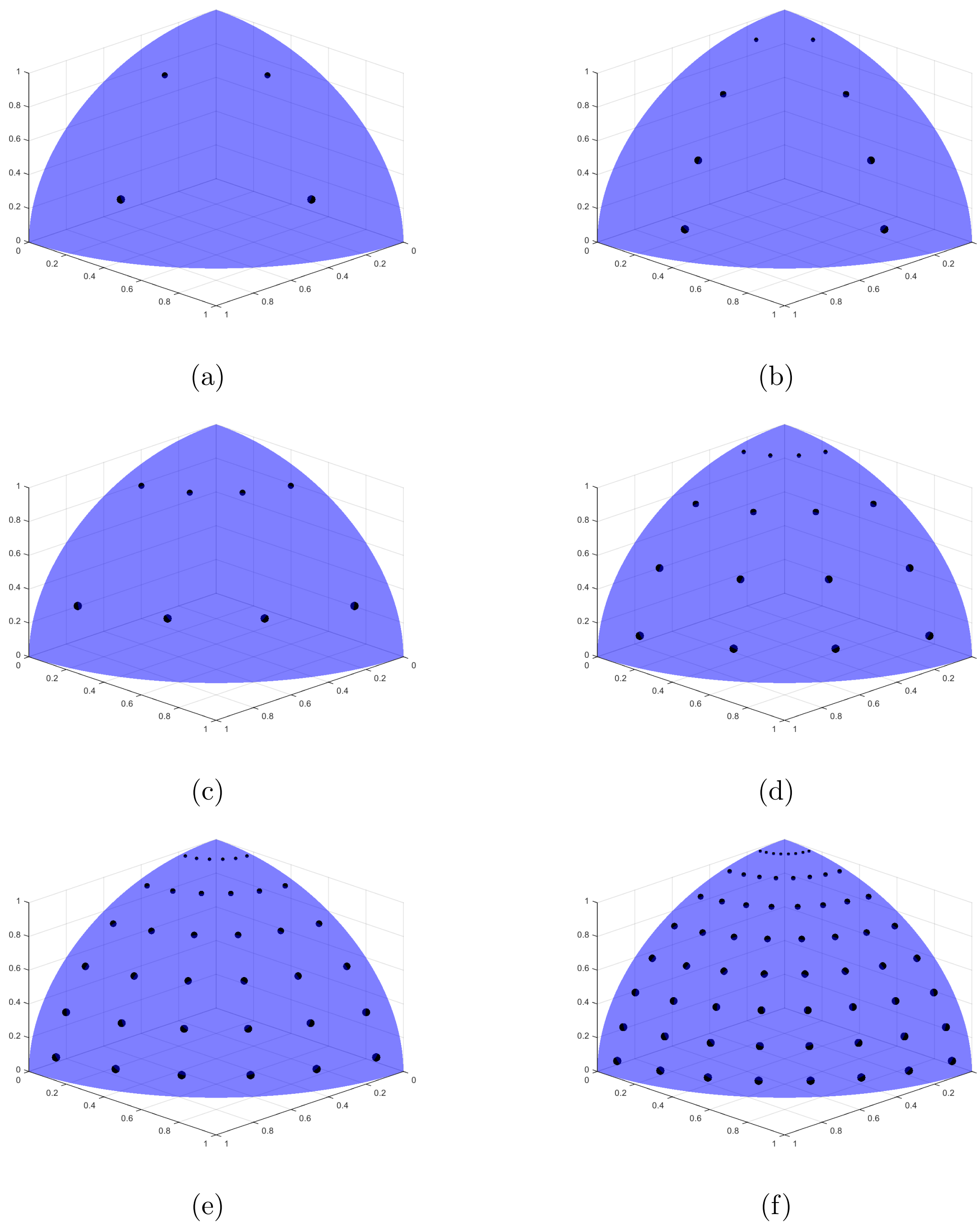
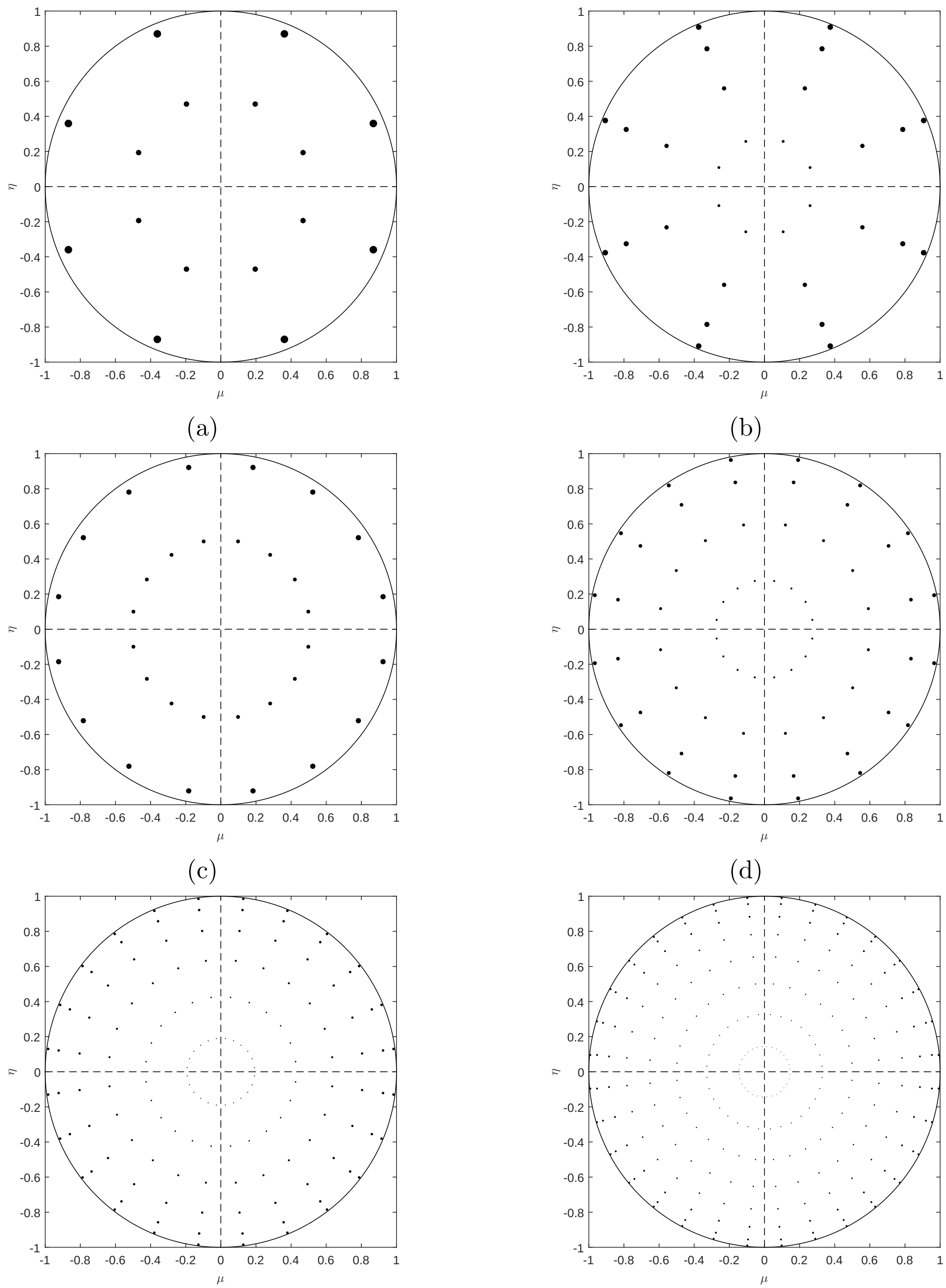


Figure 2.6: Product Gauss-Legendre-Chebyshev angular quadrature set with orders: (a) S_2^2, (b) S_2^4, (c) S_4^2, (d) S_4^4, (e) S_6^6, and (f) S_8^8.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.7: 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:

$$\Psi_{m,g}(\vec{r}) = \Psi^{inc}_{m,g}(\vec{r}) + \sum_{g'=1}^{G} \sum_{\vec{\Omega}_{m'} \cdot \vec{n} > 0} \gamma^{m' \to m}_{g' \to g}(\vec{r}) \Psi_{m',g'}(\vec{r}). \tag{2.39}$$

These (MxG) 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

$$\Psi_{m,g}(\vec{r}) = \begin{cases} \Psi^{inc}_{m,g}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^d \\ \\ \Psi_{m',g}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^r \end{cases} \tag{2.40}$$

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 2.3 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.

2.5 Spatial Discretization

For the spatial discretization of the problem domain, we simplify Eq. (2.38) 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)

$$\vec{\Omega}_m \cdot \vec{\nabla}\Psi_m + \sigma_t \Psi_m = \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m) \left[\sigma_{s,p}\Phi_{p,n,} + Q_{p,n}\right] \tag{2.41}$$

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 2.8 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 ($i.e.$ $\vec{n}' = -\vec{n}$).

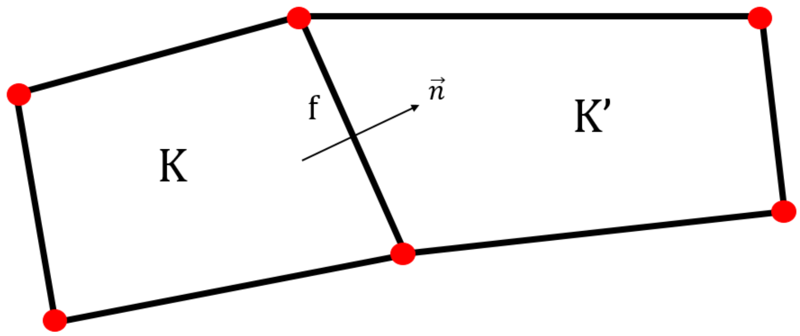


Figure 2.8: Two cells of the spatial discretization with the connecting face, f, with normal direction, $\vec{n}$, oriented from cell K to cell K'.

Next, we left-multiply Eq. (2.41) 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 $\vec{\Omega}_m$:

$$\begin{aligned} &- \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\right)_K + \sum_{f=1}^{N_f^K} \left\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \tilde{\Psi}_m \right\rangle_f + \left(\sigma_t b_m, \Psi_m\right)_K \\ &= \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p}\, b_m, \Phi_{p,n,}\right)_K + \left(b_m, Q_{p,n}\right)_K \right] \end{aligned} . \tag{2.42}$$

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:

$$\left(u, v\right)_K \equiv \int_K u\, v\, dr \tag{2.43}$$

and

$$\left\langle u, v \right\rangle_f \equiv \int_f u\, v\, ds \tag{2.44}$$

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. (2.42) can be written into the following form:

$$
\begin{aligned}
- \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m \right)_K + \left(\sigma_t b_m, \Psi_m \right)_K \\
+ \left\langle (\vec{\Omega}_m \cdot \vec{n}) \, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^+} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) \, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \backslash \partial \mathcal{D}} \\
+ \left\langle (\vec{\Omega}_m \cdot \vec{n}) \, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^d} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) \, b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\
= \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p} \, b_m, \Phi_{p,n,} \right)_K + \left(b_m, Q_{p,n} \right)_K \right]
\end{aligned}
. \tag{2.45}
$$

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,

$$\tilde{\Psi}_m(\vec{r}) = \begin{cases} \Psi_m^-, & \partial K^+ \\ \Psi_m^+, & \partial K^- \backslash \partial \mathcal{D} \\ \Psi_m^{inc}, & \partial K^- \cap \partial \mathcal{D}^d \\ \Psi_{m'}^-, & \partial K^- \cap \partial \mathcal{D}^r \end{cases}, \tag{2.46}$$

when the following trace is applied to the angular fluxes :

$$\Psi_m^{\pm}(\vec{r}) \equiv \lim_{s \to 0^{\pm}} \Psi_m\Big(\vec{r} + s(\vec{\Omega}_m \cdot \vec{n})\vec{n}\Big). \tag{2.47}$$

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

$$\begin{aligned} -\Big(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m\Big)_K + \Big(\sigma_t b_m, \Psi_m\Big)_K + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^-\Big\rangle_{\partial K^+} \\ + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^+\Big\rangle_{\partial K^- \backslash \partial \mathcal{D}} + \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_{m'}^-\Big\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\ = \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\Big(\sigma_{s,p}\, b_m, \Phi_{p,n,}\Big)_K + \big(b_m, Q_{p,n}\big)_K \right] \\ - \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^{inc}\Big\rangle_{\partial K^- \cap \partial \mathcal{D}^d} \end{aligned}. \tag{2.48}$$

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, $\partial\mathcal{D}_K = \bigcup_{f\in N_f^K} f$.

2.5.1 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. (2.48), performing another integration-by-parts on the streaming term, mulitplying by the angular weight, w_m, and summing over all elements and all angular directions. This leads to the variational form for the 1-group S_N equation:

$$
\begin{aligned}
&\sum_{m=1}^{M} w_m \Big[\Big(b_m, \vec{\Omega}_m \cdot \vec{\nabla}\Psi_m \Big)_{\mathcal{D}} + \Big(\sigma_t b_m, \Psi_m \Big)_{\mathcal{D}} \Big] \\
&+ \sum_{m=1}^{M} w_m \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m^{+}, [\![\Psi_m]\!] \Big\rangle_{E_h^i} \\
&+ \sum_{m=1}^{M} w_m \Big[\Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_{m'} \Big\rangle_{\partial\mathcal{D}_m^{r-}} - \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m \Big\rangle_{\partial\mathcal{D}_m^{-}} \Big] \\
&= \sum_{m=1}^{M} w_m \sum_{p=0}^{N_P} \sum_{n=-p}^{p} \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \Big(b_m, \sigma_{s,p}\, \Phi_{p,n,} + Q_{p,n} \Big)_{\mathcal{D}} \\
&- \sum_{m=1}^{M} w_m \Big\langle (\vec{\Omega}_m \cdot \vec{n})\, b_m, \Psi_m^{inc} \Big\rangle_{\partial\mathcal{D}_m^{-}}
\end{aligned}
\tag{2.49}
$$

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

$$\Big(u, v\Big)_{\mathcal{D}} \equiv \sum_{K\in\mathbb{T}_h} \Big(u, v\Big)_K, \tag{2.50}$$

and

$$\Big\langle u, v \Big\rangle_{E_h^i} \equiv \sum_{f\in E_h^i} \Big\langle u, v \Big\rangle_f, \tag{2.51}$$

respectively. The interior faces are designated as the non-repeating set: $E_h^i \in$

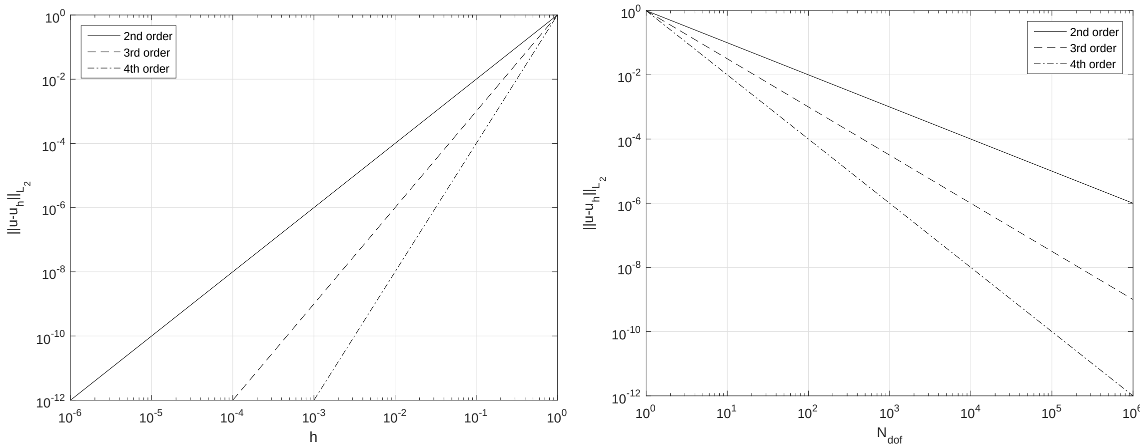


Figure 2.9: blah

$\cup_{K\in\mathbb{T}_h}\partial K\backslash\partial\mathcal{D}$. In Eq. (2.49), the interior jump term, $[\![\Psi_m]\!]$, is defined as,

$$[\![\Psi_m]\!] = \Psi_m^+ - \Psi_m^-, \tag{2.52}$$

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

$$\begin{gathered}\Psi_m^\pm \equiv \lim_{s\to 0^\pm} \Psi_m(\vec{r} - u(\vec{r})s\,\vec{\Omega}_m) \\ u(\vec{r}) = \operatorname{sgn}\left(\vec{n}(\vec{r})\cdot\vec{\Omega}_m\right)\end{gathered}. \tag{2.53}$$

$$||\Psi - \Psi_{exact}||_{DG} = C^\Psi \frac{h^q}{(p+1)^q} \tag{2.54}$$

and

$$||\Phi - \Phi_{exact}||_{DG} = C^\Phi \frac{h^q}{(p+1)^q} \tag{2.55}$$

where $q = \min(p+1/2, r-1/2)$ and r is the regularity index of the continuous transport solution.

2.5.2 Elementary Matrices on an Arbitrary Spatial Cell

In Eq. (2.48), 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 2.5.2.1, 2.5.2.2, and 2.5.2.3, respectively.

2.5.2.1 Elementary Mass Matrices

In the spatially discretized equations presented in Section 2.5, 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 [14, 15]. For cell K, we define the elementary mass matrix, $\mathbf{M}$, as

$$\mathbf{M}_K = \int_K \mathbf{b}_K \, \mathbf{b}_K^T \, dr, \tag{2.56}$$

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. (2.56) 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

$$\mathbf{M}_K = \sum_{q=1}^{N_q} w_q^K \mathbf{b}_K(\vec{x}_q)\, \mathbf{b}_K^T(\vec{x}_q). \tag{2.57}$$

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:

$$\mathbf{M}_K = \begin{bmatrix} \int_K b_1\, b_1 & \dots & \int_K b_1\, b_j & \dots & \int_K b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_i\, b_1 & \dots & \int_K b_i\, b_j & \dots & \int_K b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_{N_K}\, b_1 & \dots & \int_K b_{N_K}\, b_j & \dots & \int_K b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{2.58}$$

where an individual matrix entry is of the form:

$$M_{i,j,K} = \int_K b_i\, b_j. \tag{2.59}$$

2.5.2.2 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:

$$\vec{\mathbf{G}}_K = \int_K \vec{\nabla}\mathbf{b}_K\,\mathbf{b}_K^T\,dr, \tag{2.60}$$

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:

$$\vec{\mathbf{G}}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla}\mathbf{b}_K(\vec{x}_q)\,\mathbf{b}_K^T(\vec{x}_q). \tag{2.61}$$

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

$$\vec{\mathbf{G}}_K = \begin{bmatrix} \int_K \vec{\nabla}b_1\,b_1 & \dots & \int_K \vec{\nabla}b_1\,b_j & \dots & \int_K \vec{\nabla}b_1\,b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla}b_i\,b_1 & \dots & \int_K \vec{\nabla}b_i\,b_j & \dots & \int_K \vec{\nabla}b_i\,b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla}b_{N_K}\,b_1 & \dots & \int_K \vec{\nabla}b_{N_K}\,b_j & \dots & \int_K \vec{\nabla}b_{N_K}\,b_{N_K} \end{bmatrix}, \tag{2.62}$$

where an individual matrix entry is of the form:

$$\vec{G}_{i,j,K} = \int_K \vec{\nabla}b_i\,b_j. \tag{2.63}$$

2.5.2.3 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,

$$\vec{\mathbf{F}}_{f,K} = \int_f \vec{n}(\vec{r})\, \mathbf{b}_K\, \mathbf{b}_K^T\, ds, \tag{2.64}$$

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. (2.64), we can see that they have dimensionality, $(N_K \text{x} N_K \text{x} d)$. 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

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

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 2.8), 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:

$$\vec{\mathbf{F}}_{f,K} = \sum_{q=1}^{N_q^f} w_q^f \vec{n}(\vec{x}_q)\, \mathbf{b}_K(\vec{x}_q)\, \mathbf{b}_K^T(\vec{x}_q). \tag{2.65}$$

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,

$$\vec{\mathbf{F}}_{f,K} = \begin{bmatrix} \int_f \vec{n}\, b_1\, b_1 & \dots & \int_f \vec{n}\, b_1\, b_j & \dots & \int_f \vec{n}\, b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\, b_i\, b_1 & \dots & \int_f \vec{n}\, b_i\, b_j & \dots & \int_f \vec{n}\, b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\, b_{N_K}\, b_1 & \dots & \int_f \vec{n}\, b_{N_K}\, b_j & \dots & \int_f \vec{n}\, b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{2.66}$$

where an individual matrix entry is of the form:

$$\vec{F}_{i,j,f,K} = \int_f \vec{n}\, b_i\, b_j. \tag{2.67}$$

2.6 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 2.2, 2.3, and 2.5, 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:

$$
\begin{aligned}
&-\left(\vec{\Omega}_m\cdot\vec{\nabla}b_{m,g},\Psi_{m,g}\right)_K+\left(\sigma_{t,g}b_{m,g},\Psi_{m,g}\right)_K+\left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g},\Psi^-_{m,g}\right\rangle_{\partial K^+}\\
&\quad+\left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g},\Psi^+_{m,g}\right\rangle_{\partial K^-\setminus\partial\mathcal{D}}+\left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g},\Psi^-_{m',g}\right\rangle_{\partial K^-\cap\partial\mathcal{D}^r}\\
&\quad=\sum_{g'=1}^{G}\sum_{p=0}^{N_P}\frac{2p+1}{4\pi}\sum_{n=-p}^{p}Y_{p,n}(\vec{\Omega}_m)\left(\sigma_{s,p}^{g'\to g}\,b_{m,g},\Phi_{p,n,g'}\right)_K\\
&\quad+\left(b_{m,g},Q_{m,g}\right)_K-\left\langle(\vec{\Omega}_m\cdot\vec{n})\,b_{m,g},\Psi^{inc}_{m,g}\right\rangle_{\partial K^-\cap\partial\mathcal{D}^d}
\end{aligned}
\quad . \qquad (2.68)
$$

All of the notations used in Eq. (2.68) remain unchanged from Section 2.5.

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 2.6.1 details the iterative procedures used to solve the transport problem in energy and angle, and Section 2.6.2 then describes how we solve the spatial portion of the problem for a single energy/angle iteration.

2.6.1 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.

$$\begin{aligned} \mathbf{L}\boldsymbol{\Psi} - \mathbf{M}\boldsymbol{\Sigma}\boldsymbol{\Phi} &= \mathbf{Q} \\ \boldsymbol{\Phi} &= \mathbf{D}\boldsymbol{\Psi} \end{aligned} \tag{2.69}$$

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 and domain sources, fission sources, and scattering sources from outside the group set into the group set of interest.

$$(\mathbf{I} - \mathbf{T})\,\boldsymbol{\Phi} = \mathbf{D}\mathbf{L}^{-1}\mathbf{Q} \tag{2.70}$$

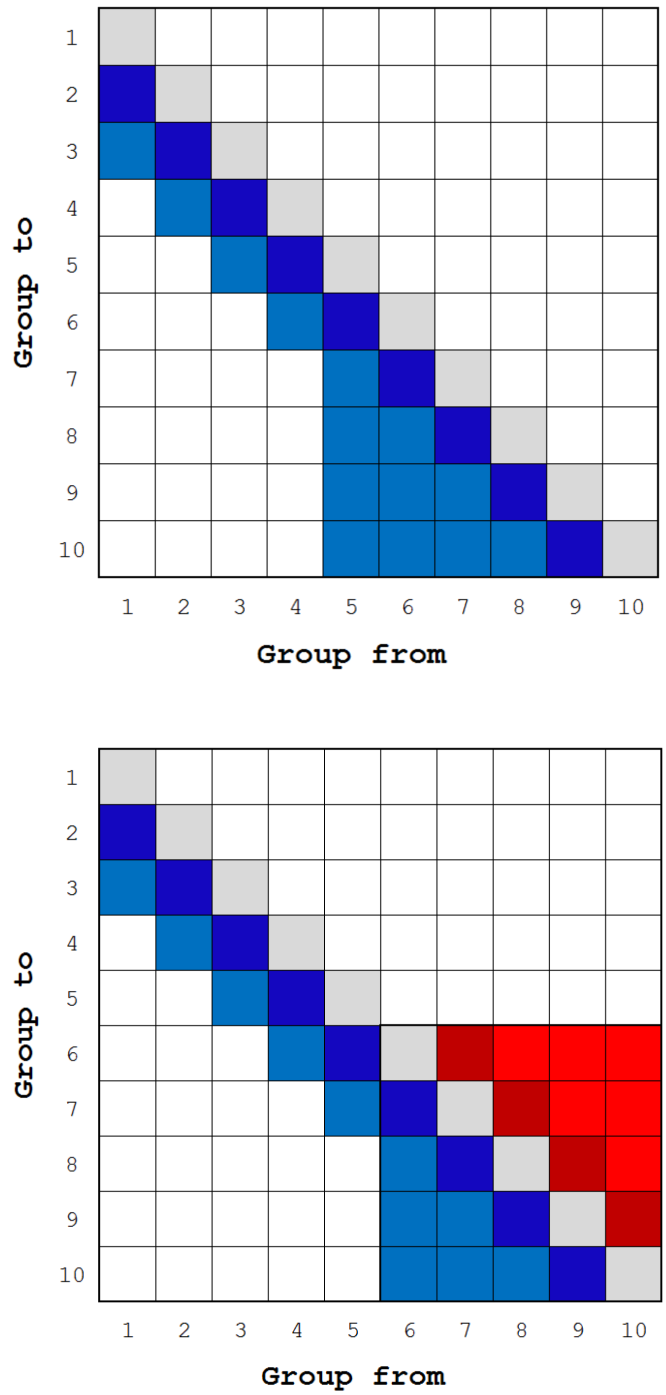


Figure 2.10: Scattering matrices (top) without and (bottom) with upscattering. The gray corresponds to within-group scattering; the blue corresponds to down-scattering in energy; and the red corresponds to up-scattering in energy.

where we define,

$$\mathbf{T} \equiv \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{\Sigma}, \tag{2.71}$$

for further brevity.

2.6.1.1 Source Iteration

One simple method to invert $(\mathbf{I}-\mathbf{T})$ of Eq. (2.70) is the *source iteration* technique, also known as *richardson iteration*. If we isolate a single group set, gs, then the

$$\mathbf{L}_{gs}\mathbf{\Psi}_{gs}^{(\ell+1)} = \mathbf{M}_{gs}\mathbf{\Sigma}_{gs}\mathbf{\Phi}_{gs}^{(\ell)} + \mathbf{Q}_{gs} \tag{2.72}$$

$$\begin{aligned} \mathbf{\Psi}_{gs}^{(\ell+1)} &= \mathbf{L}_{gs}^{-1}\left(\mathbf{M}_{gs}\mathbf{\Sigma}_{gs}\mathbf{\Phi}_{gs}^{(\ell)} + \mathbf{Q}_{gs}\right) \\ \mathbf{\Phi}_{gs}^{(\ell+1)} &= \mathbf{D}_{gs}\mathbf{\Psi}_{gs}^{(\ell+1)} \end{aligned} \tag{2.73}$$

2.6.1.2 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 [16, 17, 18]. 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 [19, 20]. 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 krlov solver for the transport equation. It is not immediately obvious how one would 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 4.

2.6.2 Spatial Solution Procedures

Section 2.6.1 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 2.6.2.1. We will generate our 2D and 3D polytope meshes by two different methods: 1) Voronoi mesh generation in Section 2.6.2.2 and 2) adaptive mesh refinement in Section 2.6.2.3.

2.6.2.1 Transport Sweeping

The full-domain transport sweep is a beneficial matrix-free scheme because of the following:

- 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) [21].
- 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 2.6.1.
- 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).

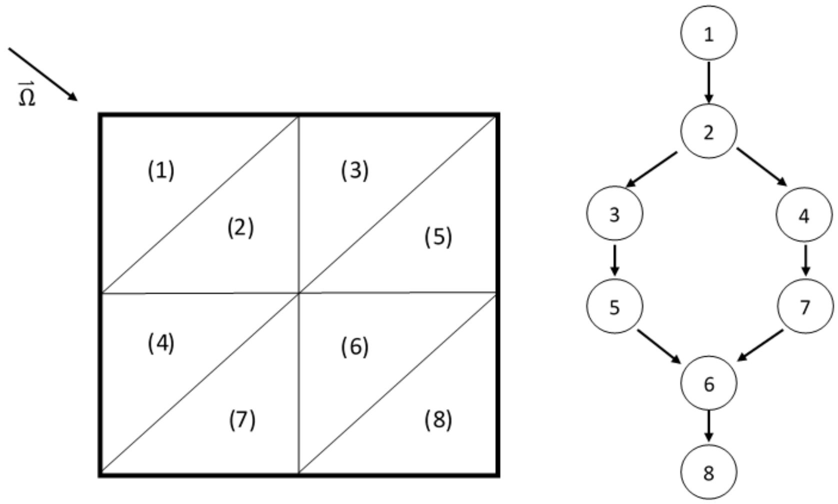


Figure 2.11: blah.

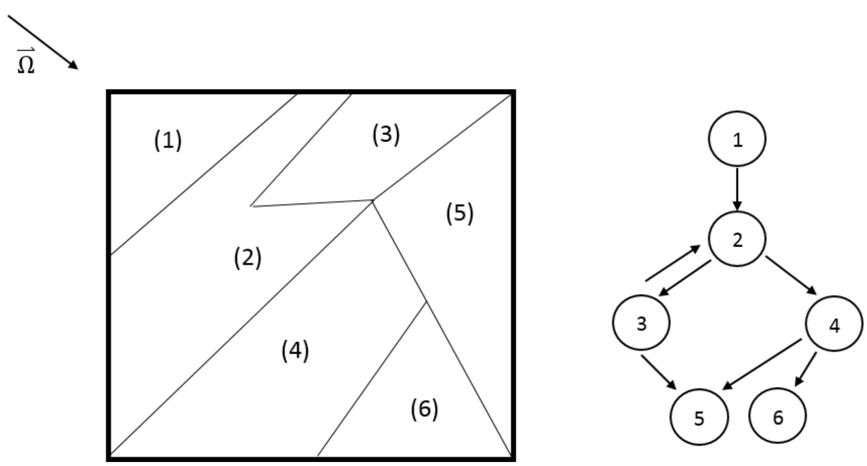


Figure 2.12: blah.

2.6.2.2 Polytope Grids Formed from Voronoi Mesh Generation

2.6.2.3 Spatial Adaptive Mesh Refinement

$$\eta_K^r = \int\limits_{\partial K} [\![\Phi^r]\!]^2 d\,s = \int\limits_{\partial K} \left(\sum_m w_m [\![\Psi_m^r]\!] \right)^2 \tag{2.74}$$

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

$$[\![\Phi(\vec{r})]\!] = \Phi^+(\vec{r}) - \Phi^-(\vec{r}), \tag{2.75}$$

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

$$\Phi^\pm(\vec{r}) = \lim_{s \to 0^\pm} \Phi(\vec{r} + s\vec{n}). \tag{2.76}$$

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:

$$\Phi^+(\vec{r}) = \begin{cases} \lim_{s \to 0^+} \Phi(\vec{r} + s\vec{n}) & \vec{r} \notin \partial\mathcal{D} \\ \sum_{\vec{\Omega}_m \cdot \vec{n} > 0} \Psi_m^-(\vec{r}) + \sum_{\vec{\Omega}_m \cdot \vec{n} < 0} \Psi_m^{inc}(\vec{r}) & \vec{r} \in \partial\mathcal{D}^d \\ \Phi^-(\vec{r}) & \vec{r} \in \partial\mathcal{D}^r \end{cases}. \tag{2.77}$$

From Eq. (2.77), 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 2.8). 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

$$\eta_K^r \geq \alpha \max_{K' \in \mathbb{T}_h^r} \left(\eta_{K'}^r\right), \tag{2.78}$$

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 $\alpha = 0$.

2.7 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 2.1 and then discretized it in energy, angle and space in Sections 2.2, 2.3, and 2.5, respectively. Appropriate boundary conditions were presented in Section 2.4. 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 2.6 by describing the procedures that will be utilized to solve our system of equations.

3. FEM BASIS FUNCTIONS FOR UNSTRUCTURED POLYTOPES

In Section 2.5, we detailed the spatial discretization of the transport equation. We then proceeded to give the functional forms for the various elementary matrices needed to form the full set of spatially-discretized PDEs. These included the mass, streaming, and surface matrices where the integrations on the element's domain and boundary require combinations of the basis functions' values and gradients.

The remainder of the this chapter is organized as follows. In Section 3.1, we present the 2D, linearly-complete, barycentric, polygonal basis functions that we will analyze in this dissertation. We give their properties along with a comparison to historical basis functions that are exact on non-polygonal elements (triangles and quadrilaterals). We then present in Section 3.2 the methodology to convert the barycentric polygonal basis functions presented in Section 3.1 into a serendipity space of basis functions with quadratic-completeness. Section 3.3 then presentes the 3D, linearly-complete, polyhedral basis functions that will be exclusively used in Chapter 4 for 3D DSA calculations. We then present numerical results pertaining to our linear and quadratic 2D basis functions in Section 3.4. We then present concluding remarks for the chapter in Section 3.5.

3.1 Linear Basis Functions on 2D Polygons

Figure 3.1, gives an image of a reference polygon along with the geometric notations we will use to define the different linear polygonal coordinates. An element, $K \in \mathbb{R}^2$, is defined by a closed set of N_K points (vertices) in $\mathbb{R}^2$. The vertices are ordered $(1, ..., N_K)$ in a counter-clockwise manner without restriction on their convexity. Face j on the polygon, e_j, is defined as the line segment between vertices j and $j+1$. The vertex $j+1$ is determined in general as $j+1 = \mod(j, N_K) + 1$, which gives a wrap-around definition of vertex $N_K + 1 = 1$.

We complete our geometric description for the polygonal coordinate system by analyzing a point $\vec{x}$ inside the polygon's domain, as also seen in Figure 3.1. α_j is the angle between the points $(\vec{x}_j, \vec{x}, \vec{x}_{j+1})$. Since element K is defined by a closed set of $\mathbb{R}^2$ points, α_j is strongly bounded: $([0, \pi])$. We conclude by defining $|\vec{u}|$ as the Euclidean distance of the vector $\vec{u}$. This means that $|\vec{x} - \vec{x}_j|$ is the distance between the points $\vec{x}$ and $\vec{x}_j$ and $|e_j|$ is the length of face j between points $\vec{x}_j$ and $\vec{x}_{j+1}$.

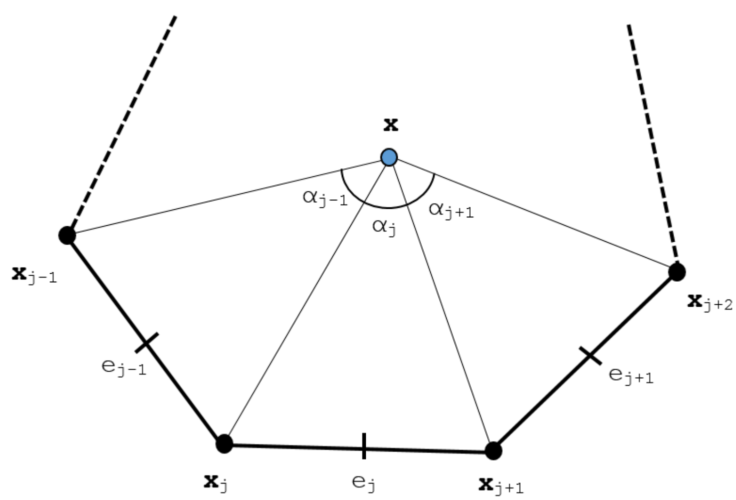


Figure 3.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

In this dissertation, all 1st-order, two-dimensional basis functions for an element K will obey the properties for barycentric coordinates. They will form a *partition of unity*,

$$\sum_{i=1}^{N_K} b_i(\vec{x}) = 1; \tag{3.1}$$

coordinate interpolation will result from an *affine combination* of the vertices,

$$\sum_{i=1}^{N_K} b_i(\vec{x})\vec{x}_i = \vec{x}; \tag{3.2}$$

and they will satisfy the *Lagrange property*,

$$b_i(\vec{x}_j) = \delta_{ij}. \tag{3.3}$$

N_K is again the number of spatial degrees with measure in element K. Using the *partition of unity* of Eq. (3.1), we can rewrite Eqs. (3.1-3.2) into a separate, compact, vectorized form for completeness

$$\sum_{i=1}^{N_K} b_i(\vec{x})\vec{c}_{i,1}(\vec{x}) = \vec{q}_1, \tag{3.4}$$

where $\vec{c}_{i,1}(\vec{x})$ and $\vec{q}_1$ are the lineary-complete constraint and equivalence terms, respectively. These terms are simply:

$$\vec{c}_{i,1}(\vec{x}) = \begin{bmatrix} 1 \\ x_i - x \\ y_i - y \end{bmatrix} \qquad \text{and} \qquad \vec{q}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.5}$$

respectively. Equation (3.4) states that our interpolation functions (the basis functions) can exactly reproduce polynomial functions up to order 1. This is why we state that our basis functions are linearly-complete. However, we will not restrict our N_K basis functions to be polynomials. In fact, of the basis functions that we will use, only the PWL coordinates are formed by combinations of polynomial functions.

3.1.1 Traditional Linear Basis Functions - $\mathbb{P}_1$ and $\mathbb{Q}_1$ Spaces

Before presenting basis function sets applicable to polygonal finite elements, we first provide two common basis functions that are exact on triangles and convex quadrilaterals: the $\mathbb{P}_1$ and $\mathbb{Q}_1$ spaces, respectively.

$$\begin{aligned} b_1(r,s) &= 1 - r - s \\ b_2(r,s) &= r \\ b_3(r,s) &= s \end{aligned} \tag{3.6}$$

and

$$\begin{aligned} b_1(r,s) &= (1-r)(1-s) \\ b_2(r,s) &= r(1-s) \\ b_3(r,s) &= rs \\ b_4(r,s) &= (1-r)s \end{aligned} \tag{3.7}$$

3.1.2 Wachspress Rational Basis Functions

$$b_j^W(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \tag{3.8}$$

where the Wachspress weight function for vertex j, w_j, has the following definition:

$$w_j(\vec{x}) = \frac{A(\vec{x}_{j-1}, \vec{x}_j, \vec{x}_{j+1})}{A(\vec{x}, \vec{x}_{j-1}, \vec{x}_j)\, A(\vec{x}, \vec{x}_j, \vec{x}_{j+1})} \tag{3.9}$$

3.1.3 Piecewise Linear (PWL) Basis Functions

$$b_j^{PWL}(x,y) = t_j(x,y) + \alpha_j^K t_c(x,y) \tag{3.10}$$

t_j is the standard 2D linear function with unity at vertex j that linearly decreases to zero to the cell center and each adjoining vertex. t_c is the 2D cell "tent" function which is unity at the cell center and linearly decreases to zero to each cell vertex.

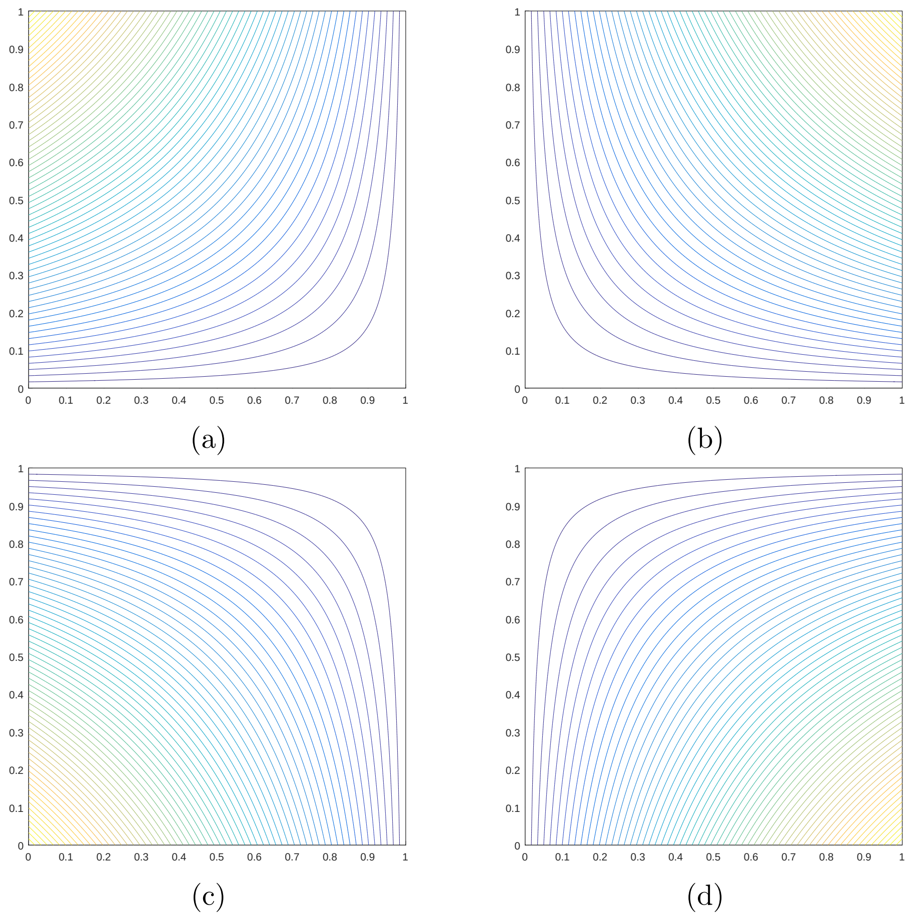


Figure 3.2: Contour plots of the Wachspress basis functions on the unit square for the vertices located at: (a) (0,1), (b) (1,1), (c) (0,0), and (d) (1,0).

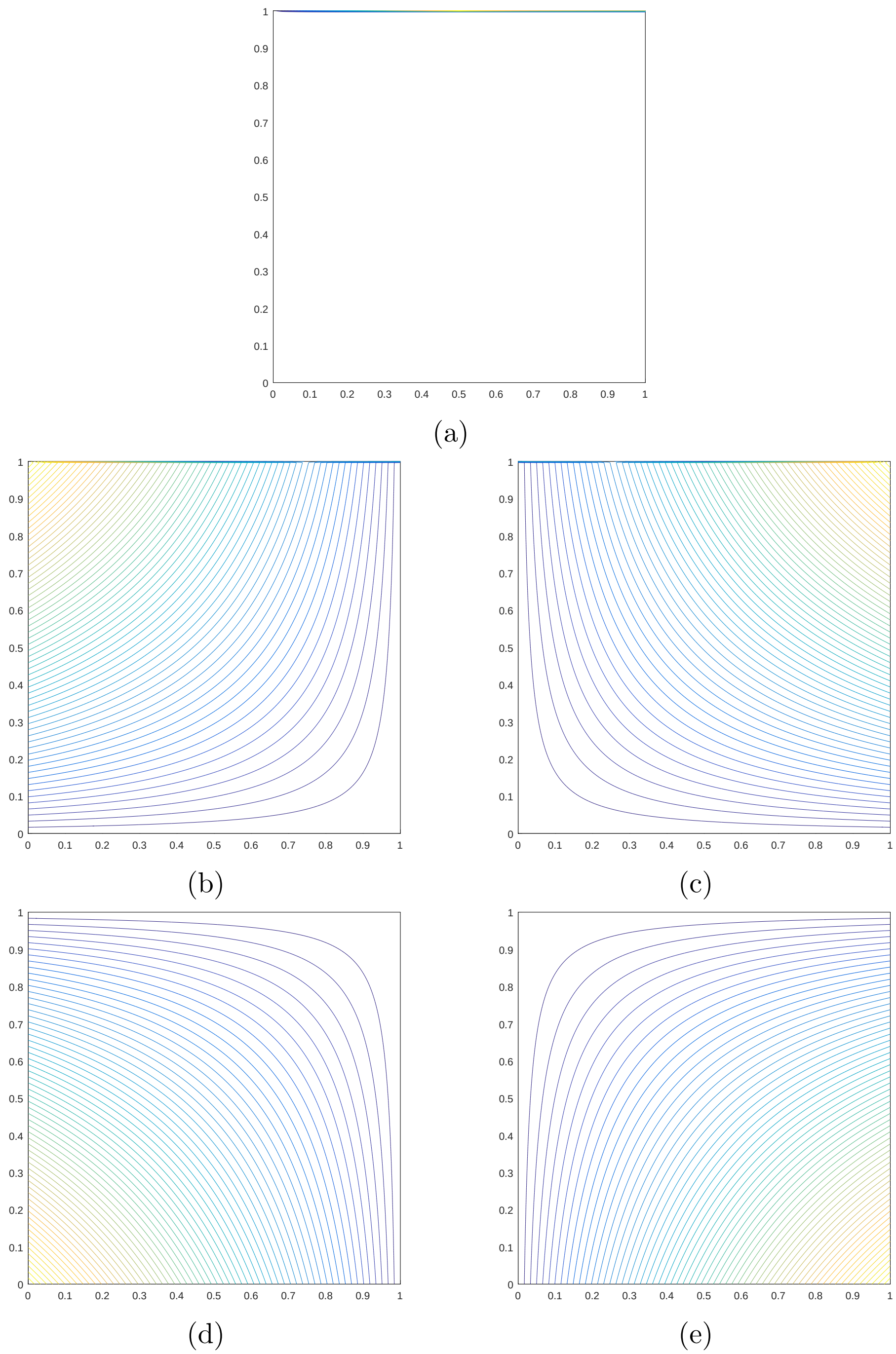


Figure 3.3: Contour plots of the Wachspress basis functions on the degnerate pentagon for the vertices located at: (a) (1/2,1), (b) (0,1), (c) (1,1), (d) (0,0), and (e) (1,0).

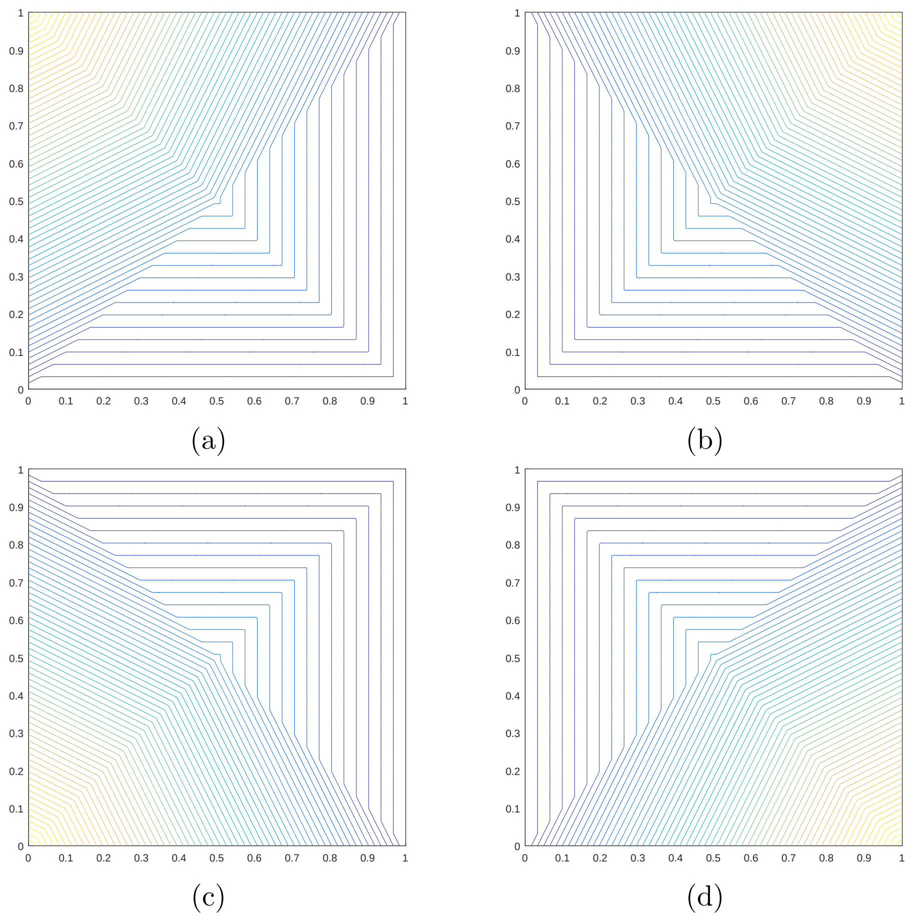
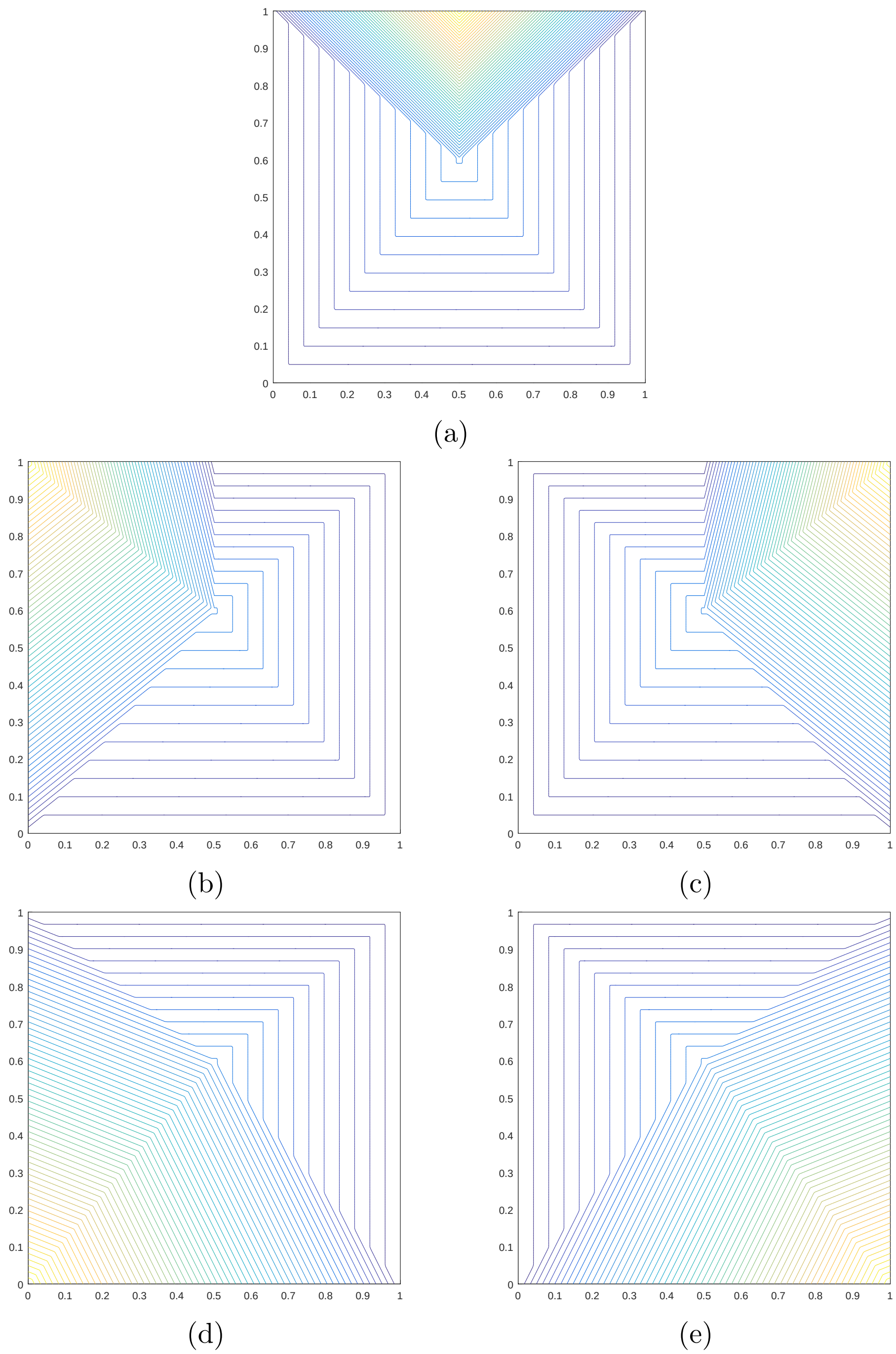


Figure 3.4: Contour plots of the Wachspress basis functions on the unit square for the vertices located at: (a) (0,1), (b) (1,1), (c) (0,0), and (d) (1,0).

α_j^K is the weight parameter for vertex j in cell K.

3.1.4 Mean Value Basis Functions

At this point, we now introduce the first new polygonal basis set for use with the transport equation: the *mean value coordinates* (MV) developed by Floater [22, 23]. The original motivation behind the MV coordinates was to approximate harmonic maps on a polygon by a set of piecewise linear maps over a triangulation of the polygon for use in computer aided graphic design.



(a) (b) (c) (d) (e)

Figure 3.5: Contour plots of the PWLD basis functions on the degnerate pentagon for the vertices located at: (a) (1/2,1), (b) (0,1), (c) (1,1), (d) (0,0), and (e) (1,0).

$$\nabla^2 u = 0, \tag{3.11}$$

with $u(\vec{r}) = u_0$ constituting a piecewise linear function

$$b_j^{MV}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})} \tag{3.12}$$

where the mean value weight function for vertex j, w_j, has the following definition:

$$w_j(\vec{x}) = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{|\vec{x}_j - \vec{x}|} \tag{3.13}$$

3.1.5 Maximum Entropy Basis Functions

The final linearly-complete 2D basis functions that we will analyze in this work are generated by use of the *maximum entropy coordinates* (ME) [24, 25, 26].

$$b_j^{ME}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}. \tag{3.14}$$

where the maximum entropy weight function for vertex j, w_j, has the following definition,

$$w_j(\vec{x}) = m_j(\vec{x}) \exp(-\kappa \cdot (\vec{x}_j - \vec{x})), \tag{3.15}$$

where κ is a vector value of dimension d that will be explained shortly. In Eq. (3.15), m_j is called the prior distribution and is a key component of Bayesian inference [27, 28]. In the context of Eq. (3.15), the prior distribution m_j can be viewed as a weight function associated with vertex j. This means that there is variability that one can employ for these weight functions. These weight functions can then be tailored depending on the application and the numerical scheme employed.

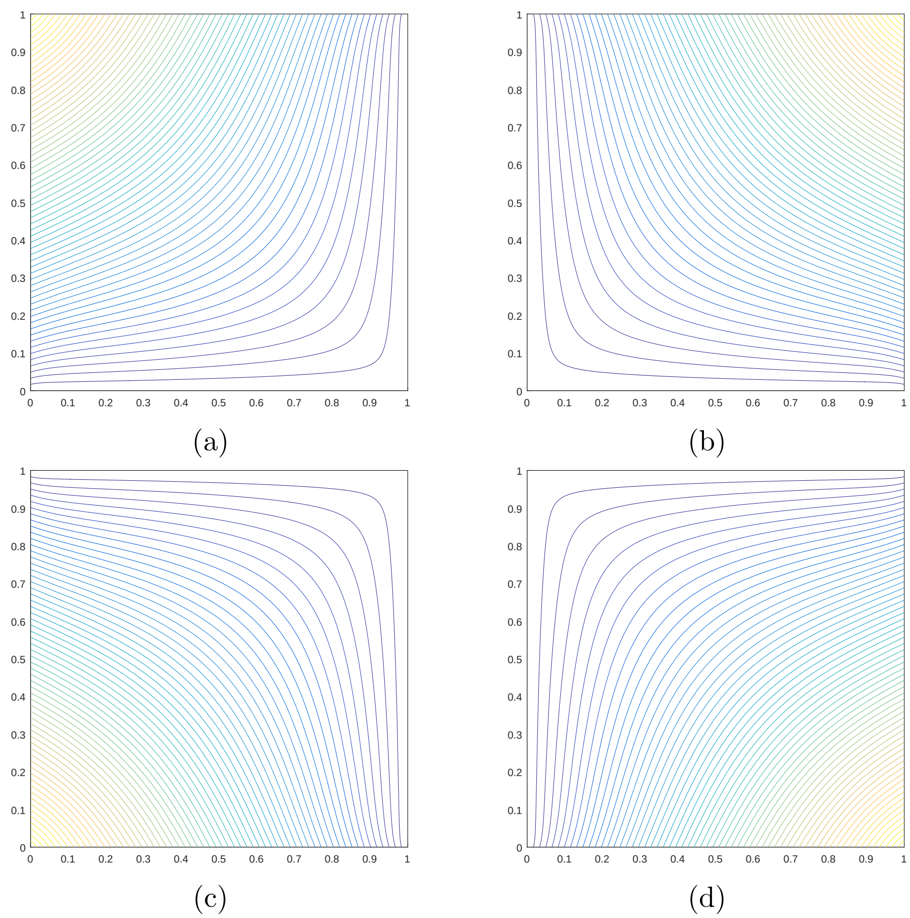


Figure 3.6: Contour plots of the mean value basis functions on the unit square for the vertices located at: (a) (0,1), (b) (1,1), (c) (0,0), and (d) (1,0).

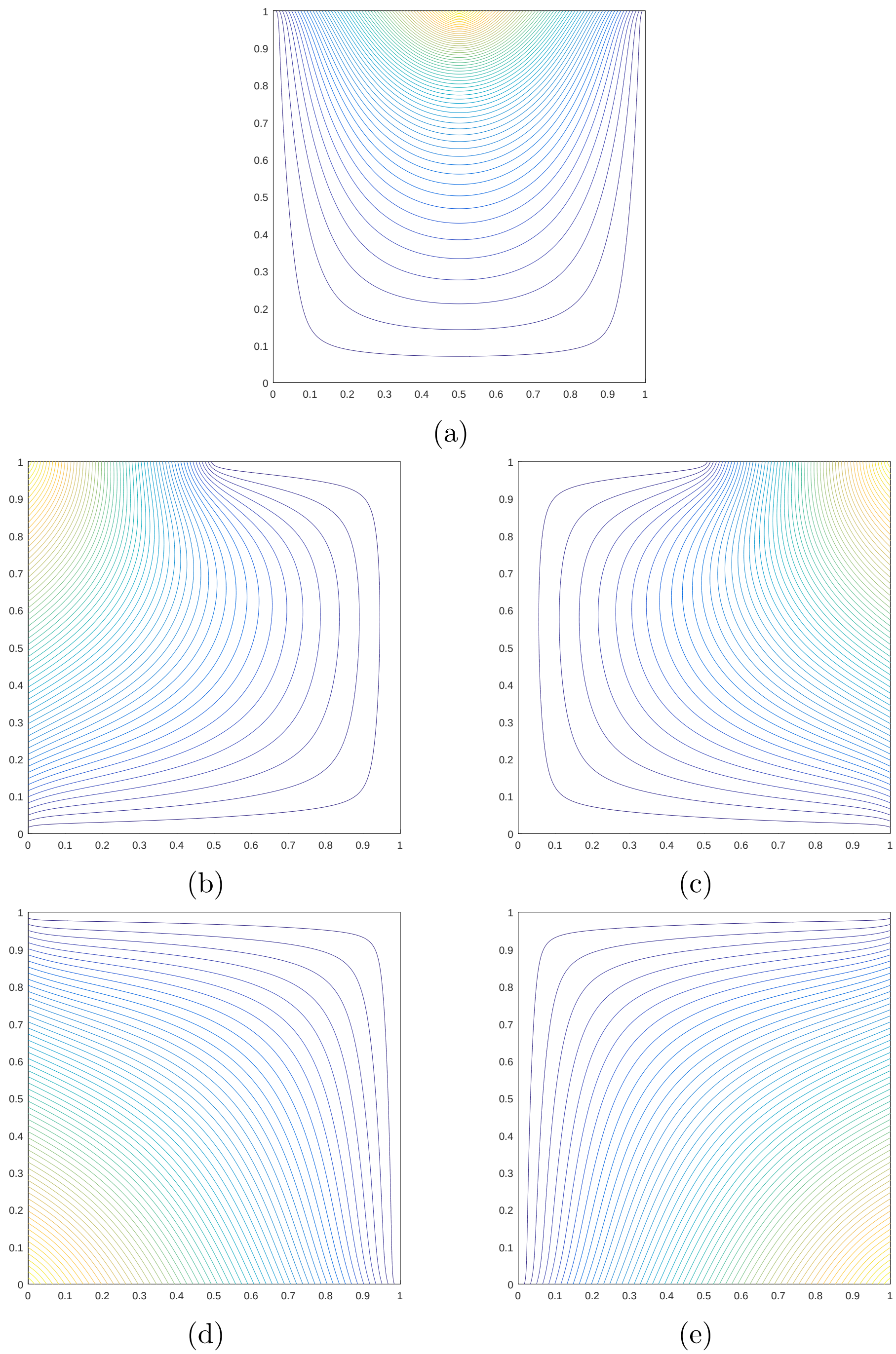


Figure 3.7: Contour plots of the mean value basis functions on the degnerate pentagon for the vertices located at: (a) (1/2,1), (b) (0,1), (c) (1,1), (d) (0,0), and (e) (1,0).

Table 3.1: Summary of the 2D coordinate systems used on polygons.

Basis Function	Dimension	Polytope Types	Integration	Direct/Iterative
Wachspress	2D/3D	Convex	Numerical	Direct
PWL	1D/2D/3D	Convex/Concave	Analytical	Direct
Mean Value	2D	Convex/Concave	Numerical	Direct
Max Entropy	1D/2D/3D	Convex/Concave	Numerical	Iterative

$$m_j(\vec{x}) = \frac{\pi_j(\vec{x})}{\sum_k \pi_k(\vec{x})} \tag{3.16}$$

where

$$\pi_j(\vec{x}) = \prod_{i \neq j-1,j} \rho_j(\vec{x}) \tag{3.17}$$

where

$$\rho_j(\vec{x}) = ||\vec{x} - \vec{x}_j|| + ||\vec{x} - \vec{x}_{j+1}|| - ||\vec{x}_{j+1} - \vec{x}_j|| \tag{3.18}$$

3.1.6 Summary of 2D Linear Basis Functions on Polygons

3.2 Quadratic Serendipity Basis Functions on 2D Polygons

$$A^+ = A^T \left(AA^T\right)^{-1} \tag{3.19}$$

$$x = A^+ b + \left[I - A^+ A\right] w \tag{3.20}$$

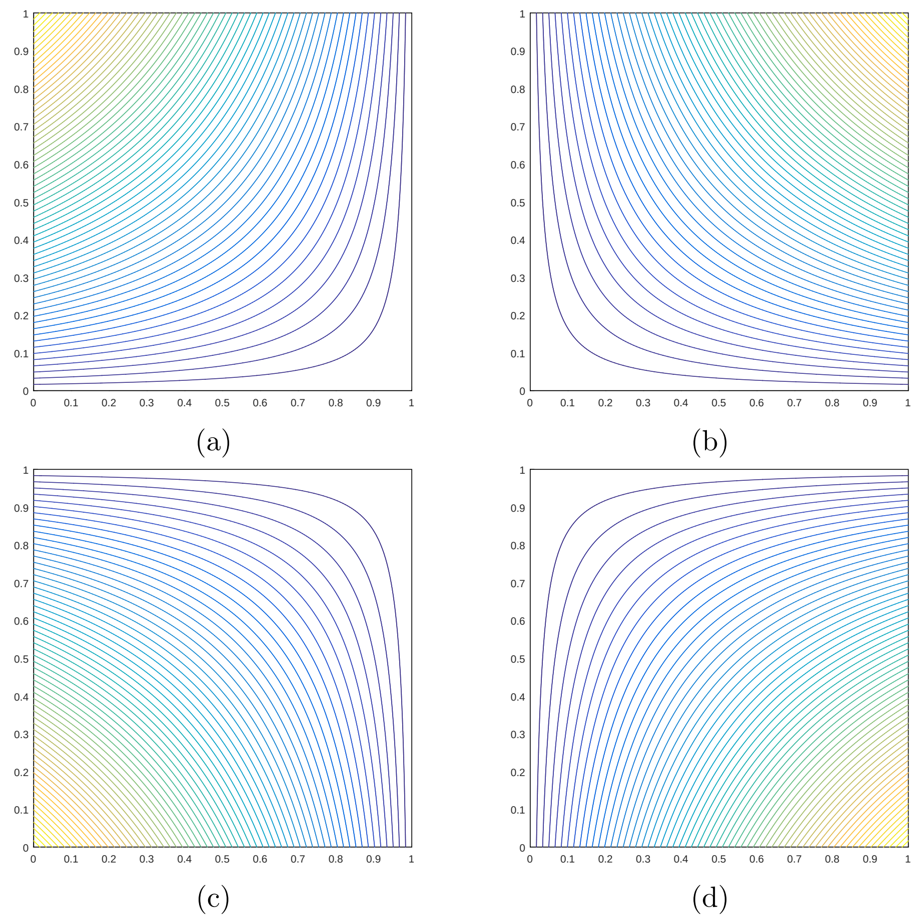


Figure 3.8: Contour plots of the maximum entropy basis functions on the unit square for the vertices located at: (a) (0,1), (b) (1,1), (c) (0,0), and (d) (1,0).

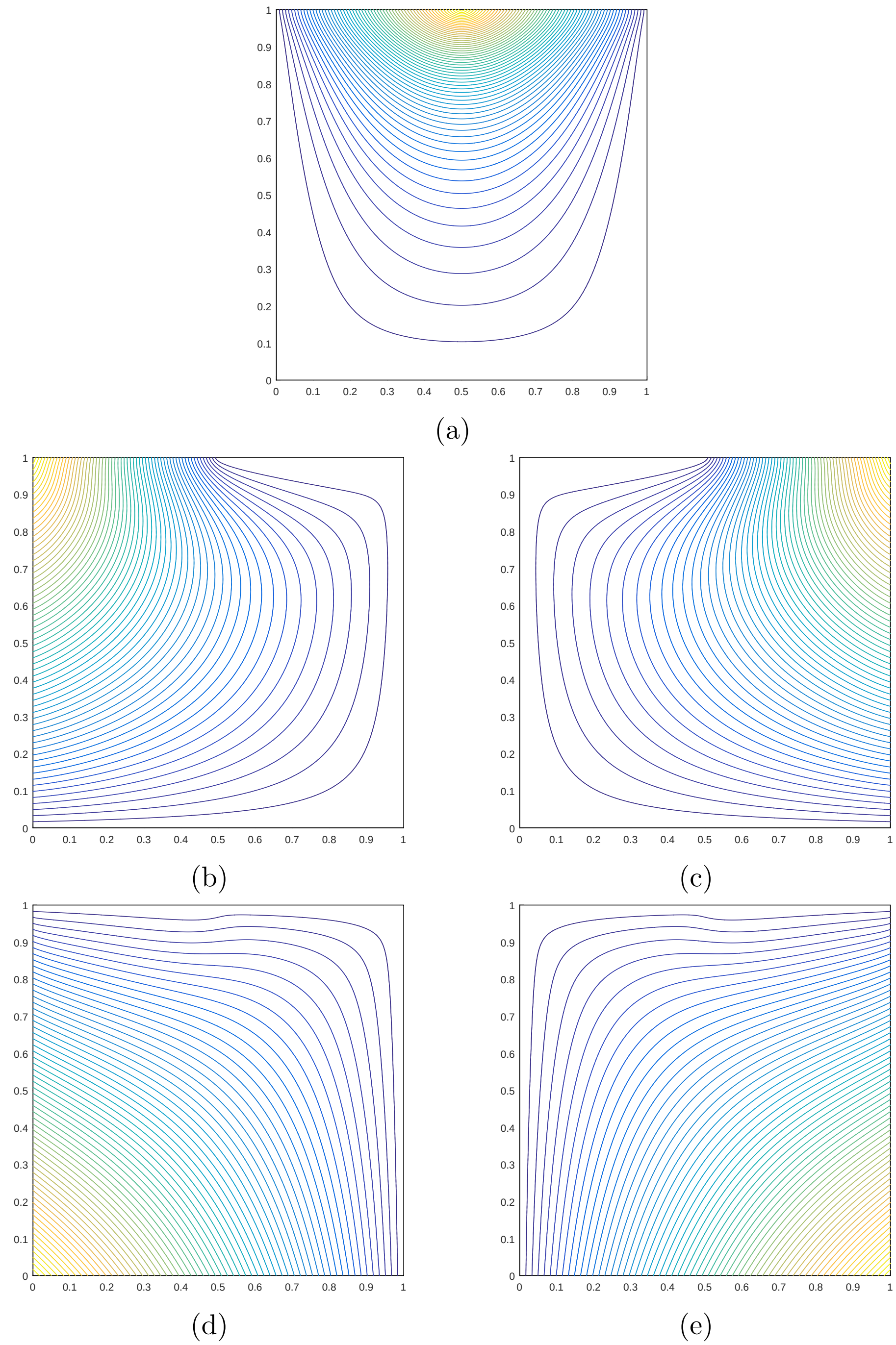


Figure 3.9: Contour plots of the maximum entropy basis functions on the degnerate pentagon for the vertices located at: (a) (1/2,1), (b) (0,1), (c) (1,1), (d) (0,0), and (e) (1,0).

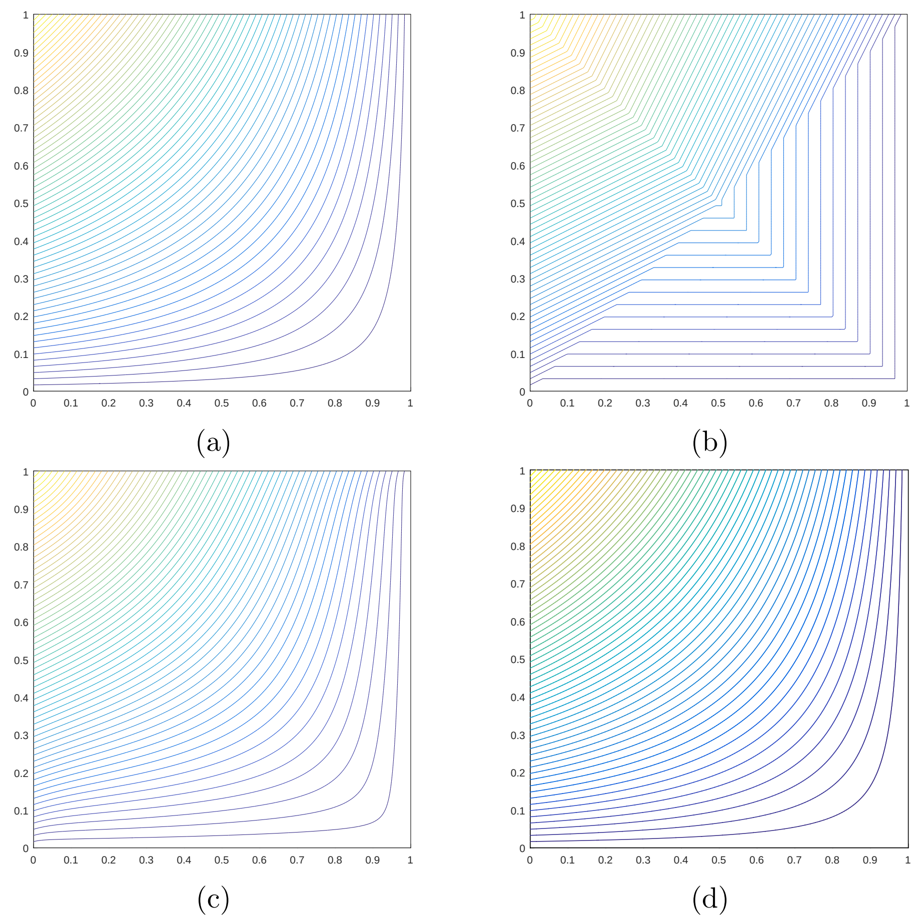
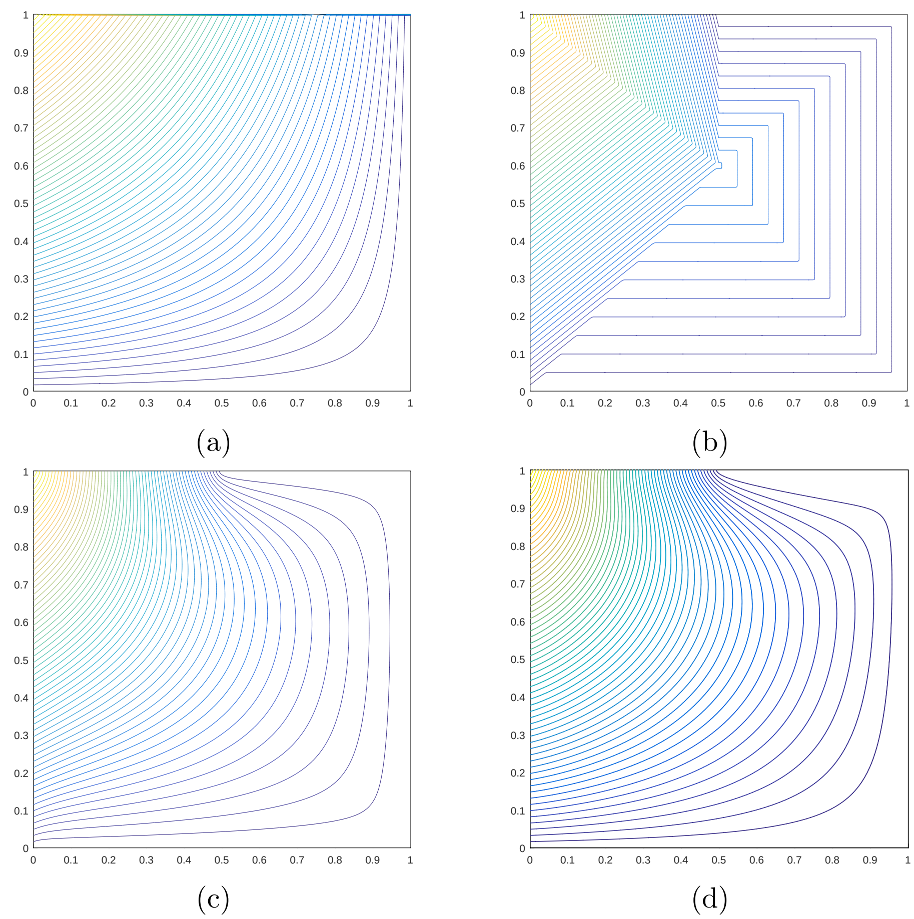


Figure 3.10: Contour plots of the different linear basis function on the unit square located at vertex (0,1): (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.



(a) (b) (c) (d)

Figure 3.11: Contour plots of the different linear basis function on the degenerate pentagon located at vertex (0,1): (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

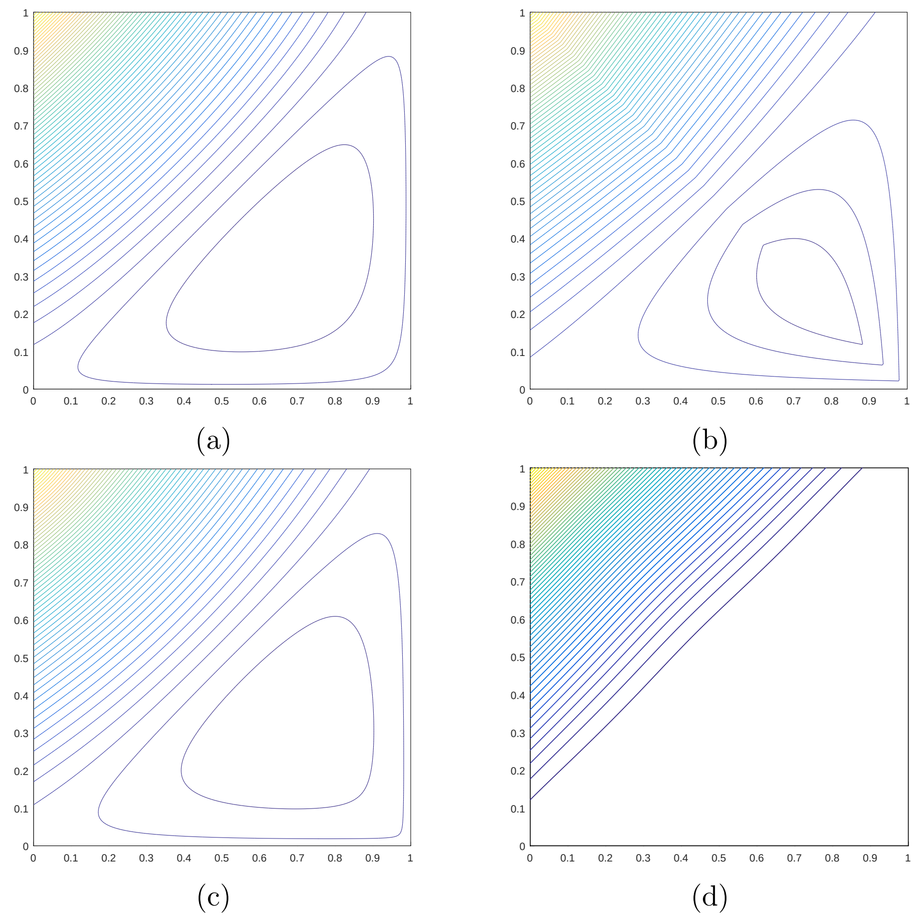


Figure 3.12: Contour plots of the different quadratic serendipity basis function on the unit square located at vertex (0,1): (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

3.2.1 Traditional Quadratic Basis Functions - $\mathbb{P}_2$ and $\mathbb{S}_2$ Spaces

3.2.2 Analytical Form of the Quadratic Serendipity Extension of the PWL Functions

3.2.3 Summary of 2D Quadratic Serendipity Basis Functions on Polygons

3.3 Linear Basis Functions on 3D Polyhedra

3.3.1 3D Linear and TriLinear Basis Functions

$$
\begin{aligned}
b_1(r,s) &= 1 - r - s - t \\
b_2(r,s) &= r \\
b_3(r,s) &= s
\end{aligned}
\tag{3.21}
$$

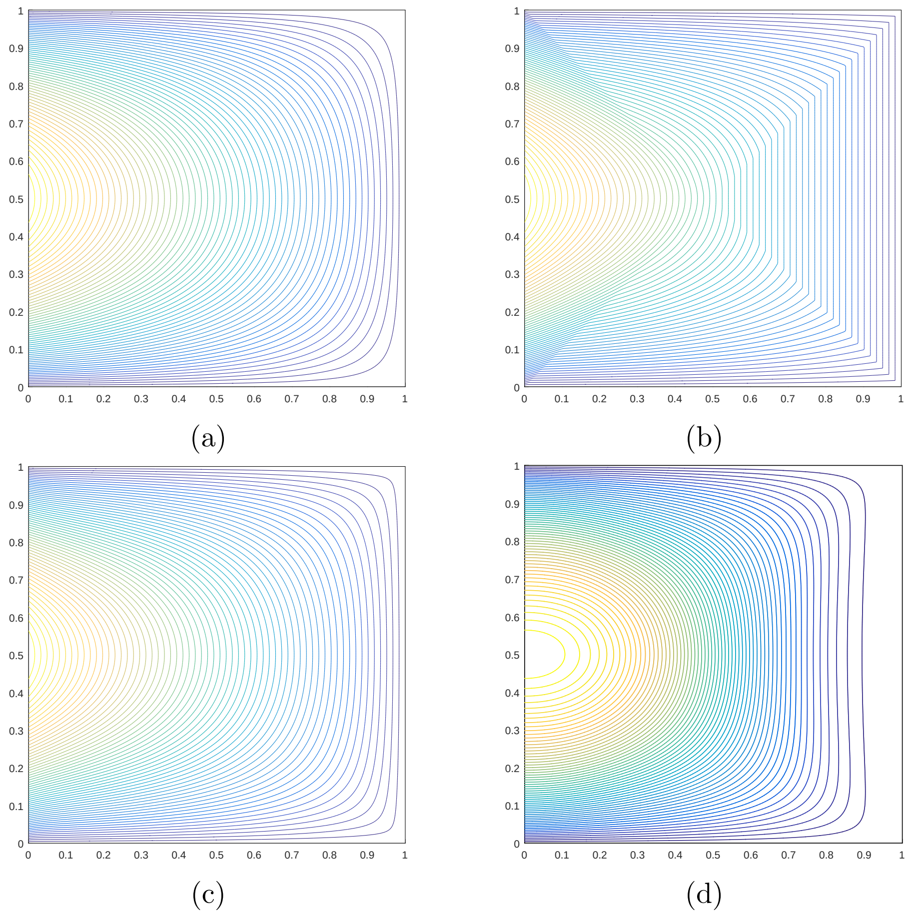


Figure 3.13: Contour plots of the different quadratic serendipity basis function on the unit square at a mid-face node located at (0,1/2): (a) Wachspress, (b) PWL, (c) mean value, and (d) maximum entropy.

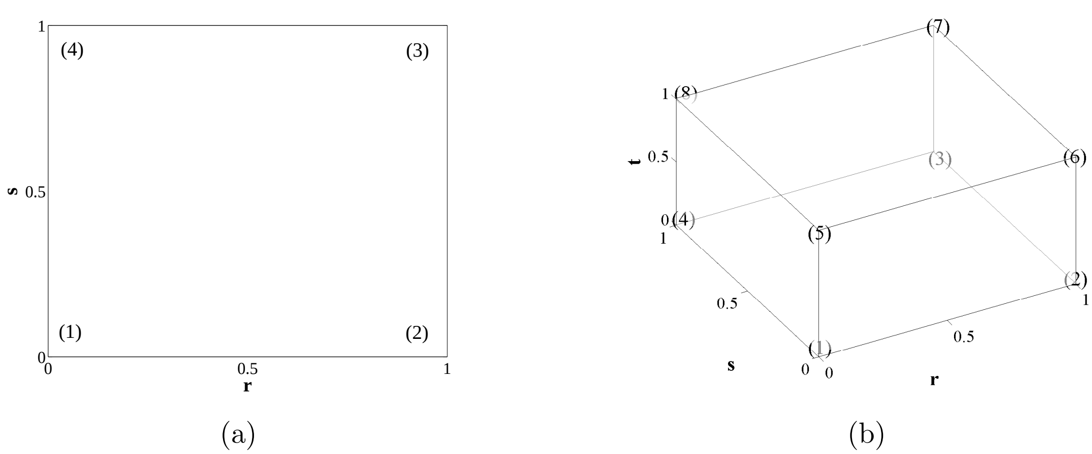


Figure 3.14: Vertex structure for the (a) unit square and (b) unit cube.

and

$$
\begin{aligned}
b_1(r,s,t) &= (1-r)(1-s)(1-t) \\
b_2(r,s,t) &= r(1-s)(1-t) \\
b_3(r,s,t) &= rs(1-t) \\
b_4(r,s,t) &= (1-r)s(1-t) \\
b_5(r,s,t) &= (1-r)(1-s)t \\
b_6(r,s,t) &= r(1-s)t \\
b_7(r,s,t) &= rst \\
b_8(r,s,t) &= (1-r)st
\end{aligned}
\tag{3.22}
$$

3.3.2 3D Piecewise Linear (PWL) Basis Functions

The 3D PWL basis functions share a similar form to the 2D PWL basis functions.

$$
b_j(x,y,z) = t_j(x,y,z) + \sum_{f=1}^{F_j} \beta_f^j t_f(x,y,z) + \alpha_j^K t_c(x,y,z) \tag{3.23}
$$

t_j is the standard 3D linear function with unity at vertex j that linearly decreases to zero to the cell center, the face center for each face that includes vertex j, and each vertex that shares an edge with vertex j. t_c is the 3D cell "tent" function which is unity at the cell center and linearly decreases to zero to each cell vertex and face center. t_f is the face "tent" function which is unity at the face center and linearly decreases to zero at each vertex on that face and the cell center. $\beta_{f,j}$ is the weight parameter for face f touching cell vertex j, and F_j is the number of faces touching vertex j. Like the previous work defining the PWLD method [29], we also choose to assume the cell and face weighting parameters are

$$\alpha_{K,j} = \frac{1}{N_K} \qquad \text{and} \qquad \beta_{f,j} = \frac{1}{N_f}, \tag{3.24}$$

respectively, where N_K is the number of vertices in cell K and N_f is the number of vertices on face f, which leads to constant values of α and β for each cell and face, respectively. This assumption of the cell weight function holds for both 2D and 3D.

3.4 Numerical Results

Now that we have presented several linear polygonal finite element basis sets along with the methodology to convert them to quadratic serendipity-like basis, we present several numerical problems to demonstrate our methodology. First, we demonstrate that the presented basis sets can capture an exactly-linear transport solution in Section 3.4.1. Next, we present some convergence properties of the basis sets using the method of manufactured solutions (MMS) in Section 3.4.2. We then present a searchlight problem and observe how the basis sets react with adaptive mesh refinement (AMR) to mitigate numerical dispersion through a vacuum in Section 3.4.3.

3.4.1 Two-Dimensional Exactly-Linear Transport Solutions

We present our first numerical example by demonstrating that the linear and quadratic polygonal finite element basis functions capture an exactly-linear solution space. We will show this by the method of exact solutions (MES). Since the coordinate interpolation of the basis functions for the linear basis functions requires exact linear interpolation (Eq. (3.2)), then an exactly-linear solution space can be captured, even on highly distorted polygonal meshes. This also applies to the quadratic serendipity space since it is formed by the product-wise pairings of the linear basis functions. We build our exact solution by investigating the 2D, 1 energy group transport problem with no scattering and an angle-dependent distributed source,

$$\mu \frac{\partial \Psi}{\partial x} + \eta \frac{\partial \Psi}{\partial y} + \sigma_t \Psi = Q(x, y, \mu, \eta), \tag{3.25}$$

where the streaming term was separated into the corresponding two-dimensional terms. We chose to drop the scattering term for this example so that the error arising from iteratively converging our solution would have no impact.

We then define an angular flux solution that is linear in both space and angle along with the corresponding 0th moment scalar flux ($\Phi_{0,0} \rightarrow \Phi$) solution:

$$\begin{array}{c} \Psi(x, y, \mu, \eta) = ax + by + c\mu + d\eta + e \\ \Phi(x, y) = 2\pi\,(ax + by + e) \end{array}. \tag{3.26}$$

One can immediately notice that our 0th moment solution is not dependent on angle. We arrive at this solution by enforcing our 2D angular quadrature set to have the following properties:

$$\sum_q w_q = 2\pi \qquad \text{and} \qquad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.27}$$

The sum of the quadrature weights is handled by simply renormalizing those that are generated in Section 2.3 to 2π.

Our boundary conditions for all inflow boundaries are then uniquely determined by the angular flux solution of Eq. (3.26). Inserting the angular flux solution of Eq. (3.26) into Eq. (3.25), we obtain the distributed source that will produce our exactly-linear solution space:

$$Q(x, y, \mu, \eta) = a\mu + b\eta + \sigma_t\,(c\mu + d\eta) + \sigma_t\,(ax + by + e)\,. \tag{3.28}$$

It is noted that the angular dependence of the source can be removed (which can ease the code development burden) if one sets

$$\begin{aligned} a &= -c\,\sigma_t, \\ b &= -d\,\sigma_t. \end{aligned} \tag{3.29}$$

For this example, we test the various 2D polygonal finite element basis functions on six different mesh types. These mesh types include triangular, quadrilateral, and polygonal meshes:

1. Orthogonal cartesian mesh formed by the intersection of 11 equally-spaced vertices in both the x and y dimensions. This forms a 10x10 array of quadrilateral mesh cells.

2. Ordered-triangular mesh formed by the bisection of the previous orthogonal cartesian mesh (forming 200 triangles all of the same size/shape).

3. Quadrilateral shestakov grid formed by the randomization of vertices based on a skewness parameter [30, 31]. With a certain range of this skewness parameter, highly distorted meshes can be generated.

4. Sinusoidal polygonal grid that is generated by the transformation of a uniform orthogonal grid based on a sinusoid functional. The transformed vertices are then converted into a polygonal grid by computing a bounded Voronoi diagram.

5. Kershaw's quadrilateral z-mesh [32]. This mesh is formed by taking an orthogonal quadrilateral grid and displacing certain interior vertices only in the y dimension.

6. A polygonal variant of the quadrilateral z-mesh. The polygonal grid is formed in a similar manner to the sinusoidal polygonal mesh with a Voronoi diagram.

We also wish that both the angular flux solution as well as the 0th moment solution are strictly positive everywhere. Therefore, we set the function parameters in Eq. (3.26) to $\sigma_t = a = c = d = e = 1.0$ and $b = 1.5$. We gave the solution the 40% tilt in space ($a \neq b$) so that it would not align with the triangular mesh. Using an S8 LS quadrature set, we ran all combinations of the polygonal basis functions and the mesh types. The linear solutions for the Wachspress, PWL, mean value, linear maximum entropy, and quadratic serendipity maximum entropy basis functions are presented in Figures 3.15, 3.16, 3.17, 3.18, and 3.19, respectively. We can see that for all the polygonal basis functions, an exact linear solution is captured as shown by the unbroken nature of the contour lines. This even holds on the highly distorted quadrilateral shestakov mesh.

3.4.2 Convergence Rate Analysis by the Method of Manufactured Solutions

The next numerical example we investigate involves calculating the convergence rate of the solution error via the method of manufactured solutions (MMS). Like MES, MMS enforces a given solution by use of a derived functional form for the driving source of the problem (Q_{ext}). However, unlike MES, we enforce a spatial solution that cannot be captured by the interpolation of the finite element space. Using a specification of the Bramble-Hilbert [33] lemma, the difference between the exact solution, Φ_e, and the discretized solution, Φ_h, residing in the Sobolev space, $W_{\mathcal{D}}^h \in \mathcal{T}_h$, for a given polynomial interpolation order, k, is

$$||\Phi_e - \Phi_h||_2 \leq C \frac{h^{q+1/2}}{(k+1)^q}, \tag{3.30}$$

where $q = \min(k + 1/2, r)$, h is the maximum diameter of all mesh cells, C is some constant independent of the mesh, r is a measure of the regularity of the transport solution, and $|| \cdot ||_2$ is the L_2 norm [34]. For transport solutions with

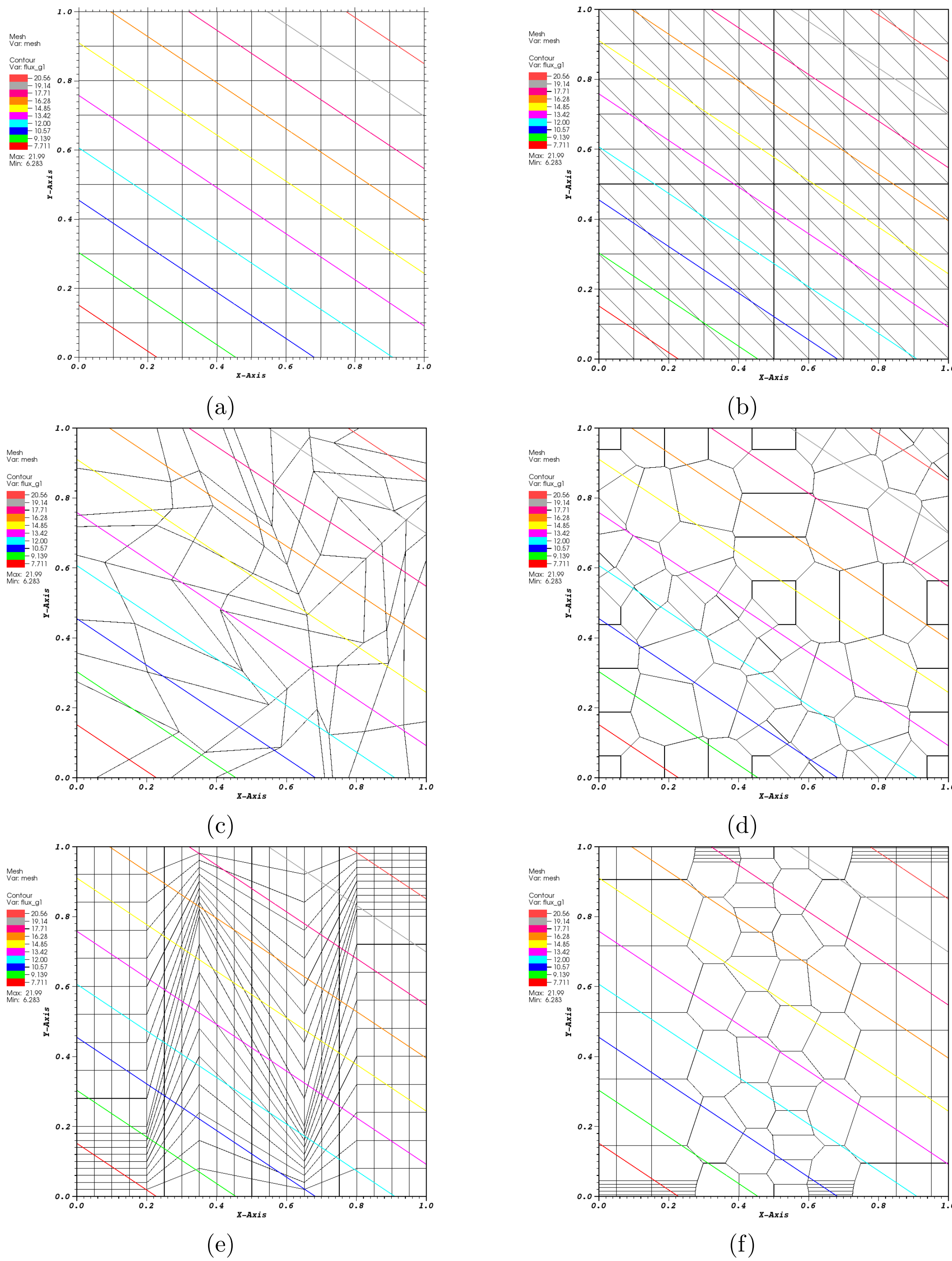


Figure 3.15: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

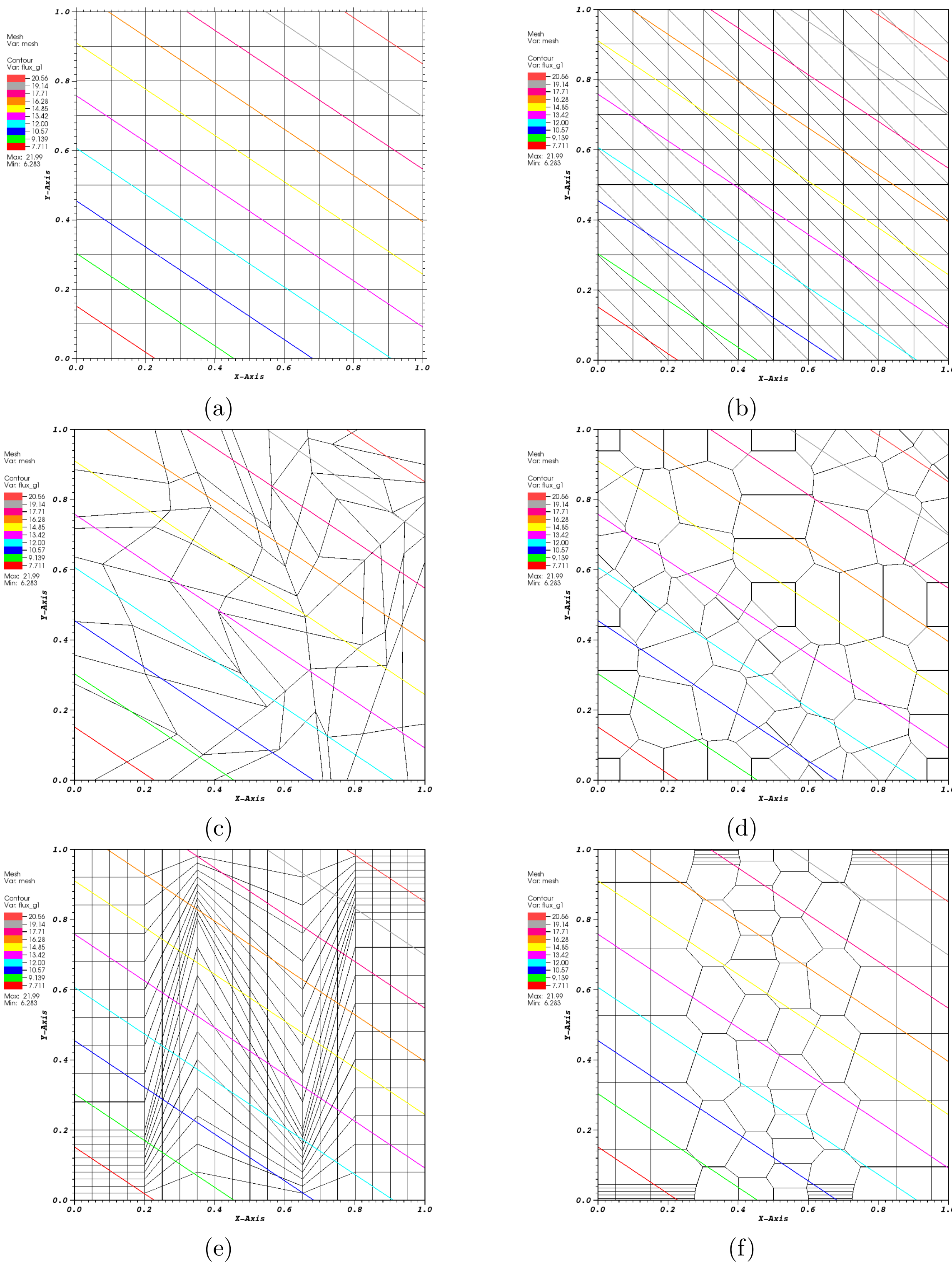


Figure 3.16: Contour plots of the exactly-linear solution with the PWL basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

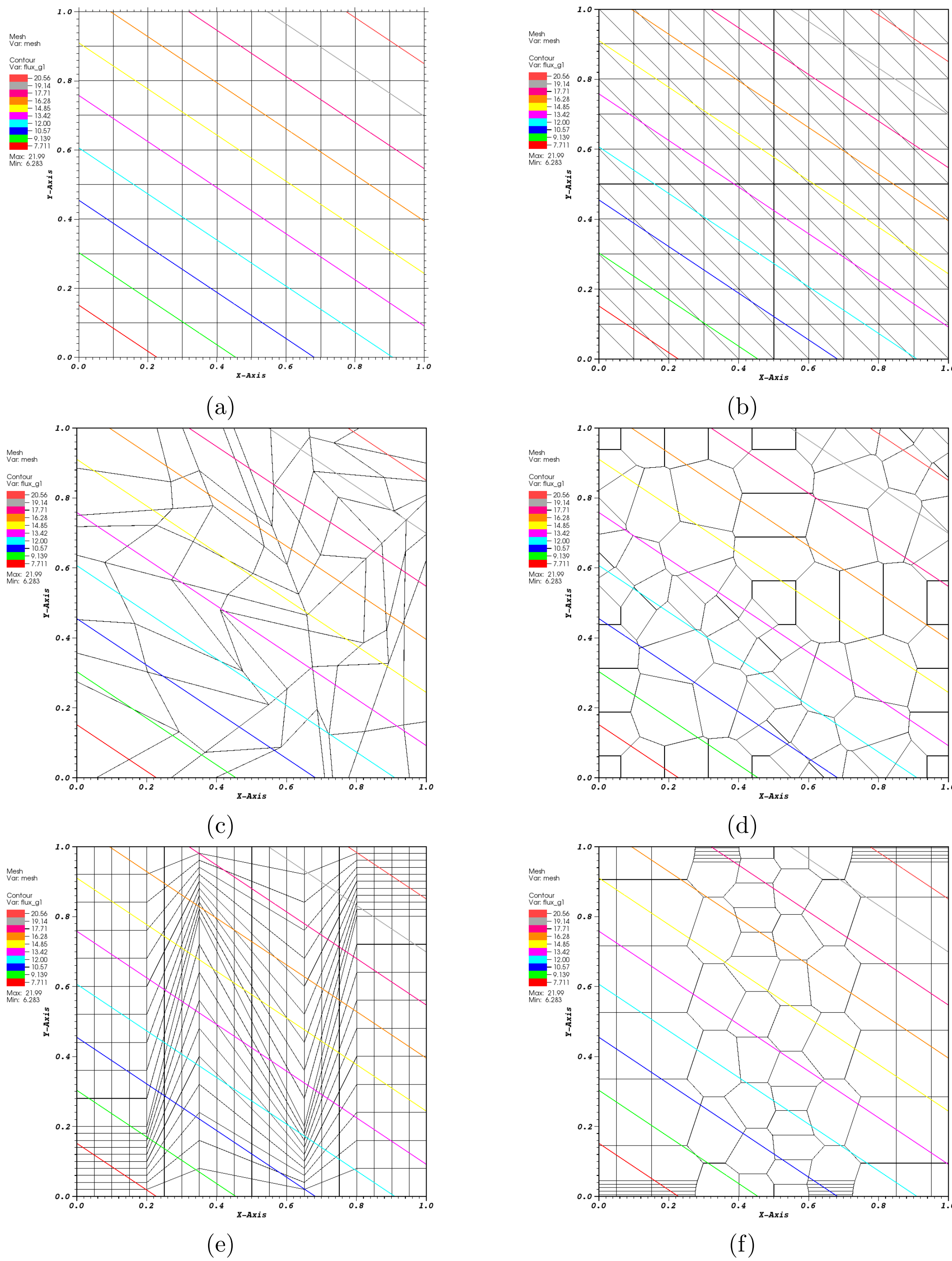
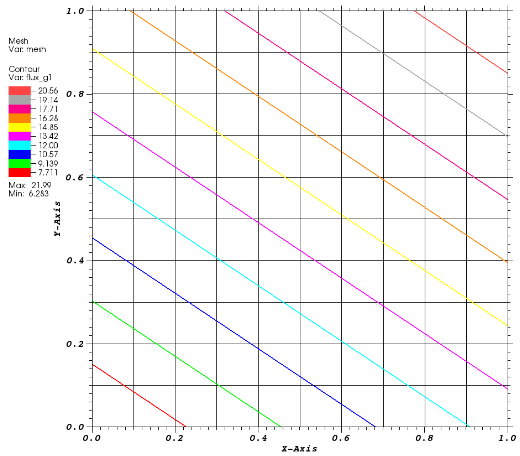
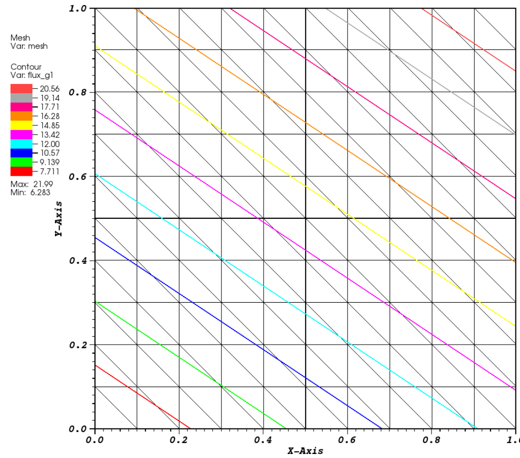


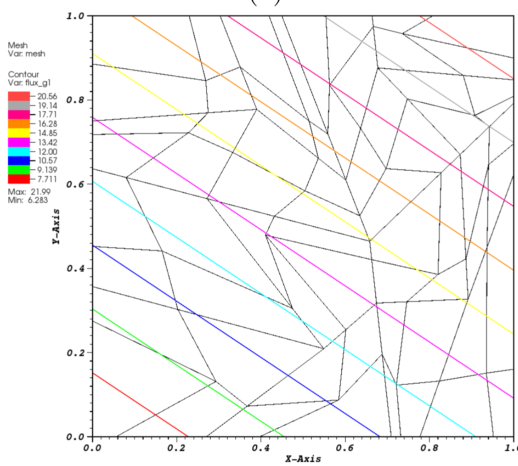
Figure 3.17: Contour plots of the exactly-linear solution with the mean value basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.



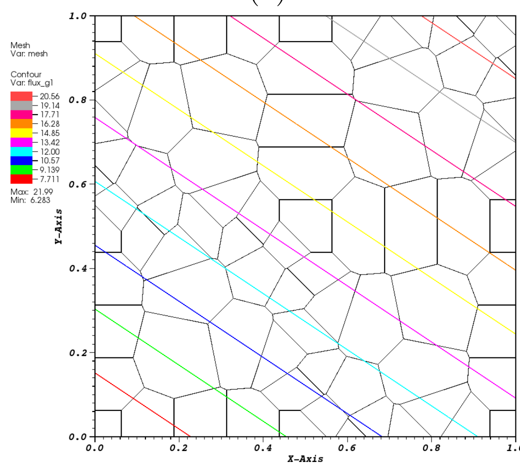
(a)



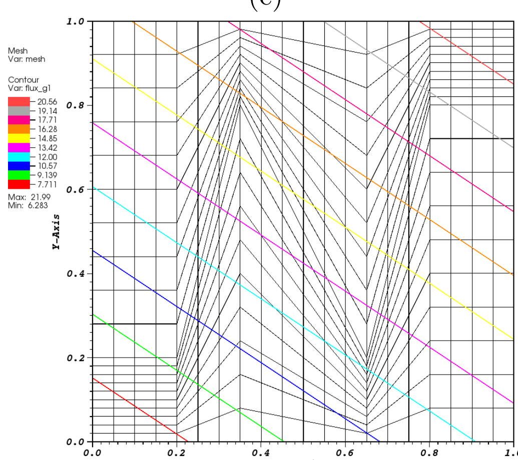
(b)



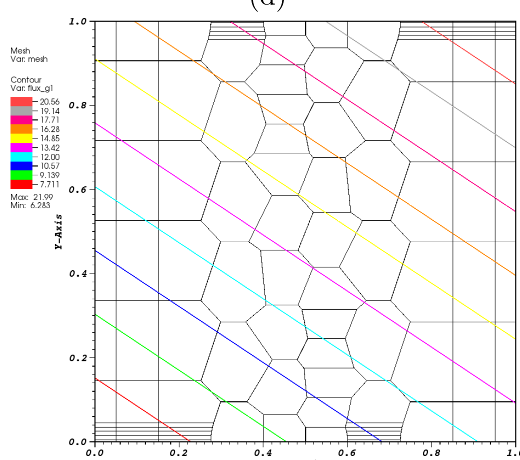
(c)



(d)



(e)



(f)

Figure 3.18: Contour plots of the exactly-linear solution with the linear maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

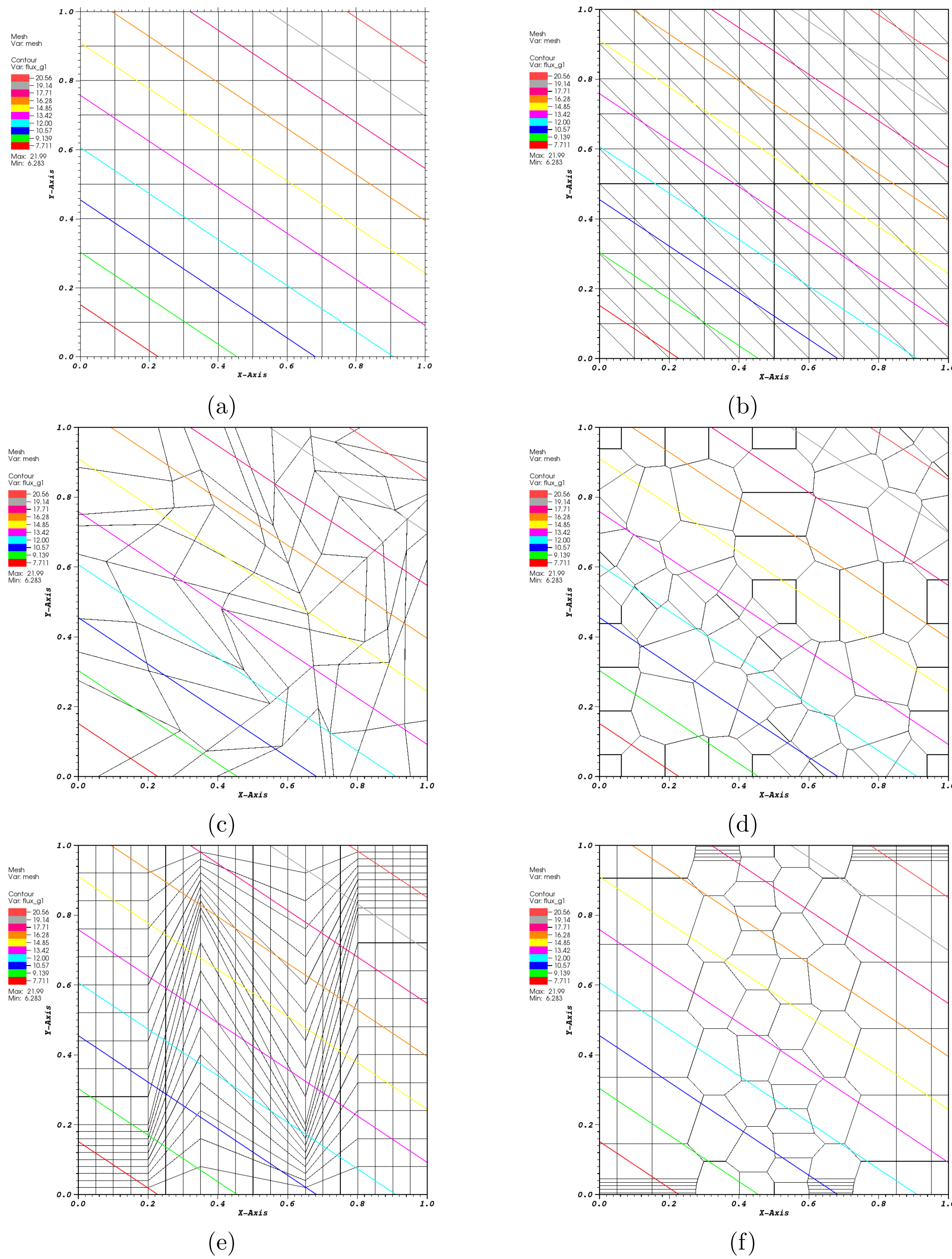


Figure 3.19: Contour plots of the exactly-linear solution with the quadratic serendipity maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

sufficient discontinuities, r can approach 0. However, if the chosen solution is at least C^k continuous and a topologically regular mesh is used, then the convergence rate becomes strictly $(k+1)$. For structured meshes, h is straightforward to define. Unfortunately, this becomes a problem in general for polytope meshes. Instead, the grid resolution can be related to the number of spatial degrees of freedom (N_{dof}),

$$N_{dof} \propto h^{-d}, \tag{3.31}$$

where d is the dimensionality of the problem. Using this relationship for the grid resolution, along with the error norm of Eq. (3.30), we can define a general relationship between the error norm and the number of spatial degrees of freedom:

$$||\Phi_e - \Phi_h||_2 \propto N_{dof}^{-\frac{k+1}{d}}. \tag{3.32}$$

The result of Eq. (3.32) states that we expect convergence rates of N_{dof}^{-1} and $N_{dof}^{-2/3}$ for linear basis functions in 2D and 3D, respectively. Conversely, quadratic basis functions will yield convergence rates of $N_{dof}^{-3/2}$ and N_{dof}^{-1} in 2D and 3D, respectively.

For this example, we choose the following solution and problem parameters and characteristics:

1. Constant total cross section so that parameterized material properties are not necessary;
2. No scattering to avoid solution discontinuities from the S_N discretization;
3. No solution dependence in angle to avoid introducing angular discretization error;
4. Analytical solutions that are C^∞ continuous in space for both the angular flux

and 0th order flux moment - this leads to integrable solution spaces that satisfy the Bramble-Hilbert lemma no matter what polynomial order is selected for the basis functions;

5. The angular flux solution is zero on the boundary for all incident directions - this is identical to vacuum boundaries which can ease code development.

To satisfy these characteristics, we choose to analyze two different solution spaces. The first is a smoothly varying sinusoid solution with no extreme local maxima. The second solution is a product of a quadratic function and a gaussian which yields a significant local maximum.

The sinusoid flux solutions, $\{\Psi^s,\ \Phi^s\}$, have the following parameterized form,

$$\begin{aligned} \Psi^s(x,y) &= \sin(\nu\frac{\pi x}{L_x})\sin(\nu\frac{\pi y}{L_y}), \\ \Phi^s(x,y) &= 2\pi\sin(\nu\frac{\pi x}{L_x})\sin(\nu\frac{\pi y}{L_y}), \end{aligned} \tag{3.33}$$

where ν is a frequency parameter. We restrict this paremeter to positive integers ($\nu = 1, 2, 3, ...$) to maintain characteristic 5 of the solution and problem space. The gaussian solution space, $\{\Psi^g,\ \Phi^g\}$, that has its local maximum centered at (x_0, y_0) has the parameterized form,

$$\begin{aligned} \Psi^g(x,y) &= C_M x(L_x - x)y(L_y - y)\exp(-\frac{(x-x_0)^2 + (y-y_0)^2}{\gamma}), \\ \Phi^g(x,y) &= 2\pi C_M x(L_x - x)y(L_y - y)\exp(-\frac{(x-x_0)^2 + (y-y_0)^2}{\gamma}), \end{aligned} \tag{3.34}$$

where the constants in the equations are:

$$C_M = \frac{100}{L_x^2 L_y^2} \qquad \gamma = \frac{L_x L_y}{100}. \tag{3.35}$$

For this example, we choose the dimensionality of our problem to be $[0,1]^2$ which makes $L_x = L_y = 1$ for both the sinusoid and gaussian solutions. For the sinusoid solution, we select the frequency parameter, ν, to be 3 and for the gaussian solution we set the local maximum: $x_0 = y_0 = 0.75$. With these parameters, the sinusoid solution will have local minima and maxima of -2π and 2π, respectively, and the gaussian solution will have a global maximum of $\frac{225}{32}\pi \approx 22.1$.

3.4.3 Searchlight Problem

The next example models a beam or searchlight. Similar problems were investigated in Dedner and Vollmöller [35] and Wang and Ragusa [36]. In this problem, an incident beam of neutrons is shined onto a small portion of a boundary, propagates through a vacuum, and then exits through a small portion of a different boundary. As the beam propagates through the vacuum, the spatial discretization causes radiation outflow through all downwind cell faces. This leads to numerical dispersion and will cause to beam to artificially broaden.

In this problem, we investigate an $\mathbb{R}^2$ domain of size $[0,1]^2$ cm. The radiation enters the left boundary between $0.2 \leq y \leq 0.4$ with an un-normalized angular direction of $[1, 0.4]$. For this chosen direction, the radiation beam would analytically leave the right boundary between $0.6 \leq y \leq 0.8$. This means that any radiation leaving the right boundary for all other y values is due to the numerical dispersion of the beam.

We investigated this problem using several of the 2D polygonal basis functions as outlined in Sections 3.1 and 3.2 as well as

3.5 Conclusions

In this chapter, we have presented

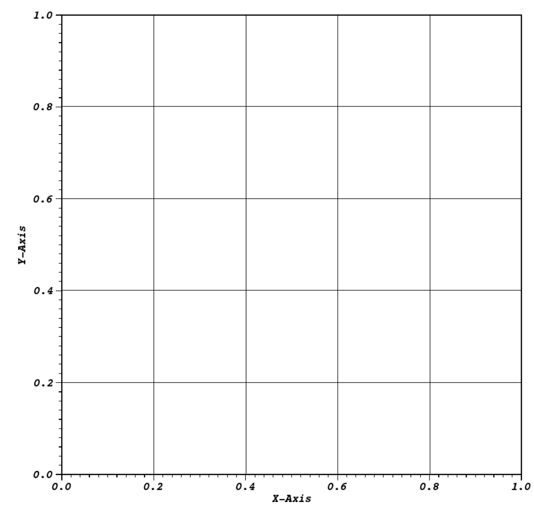
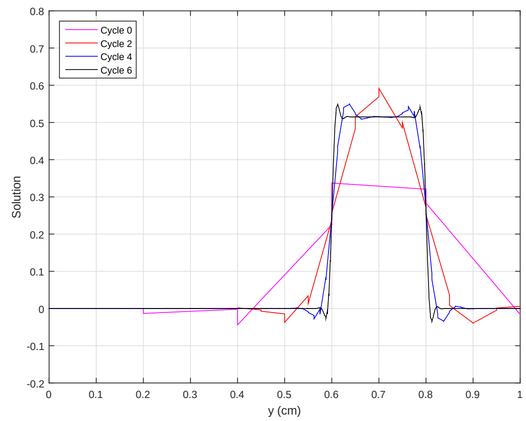
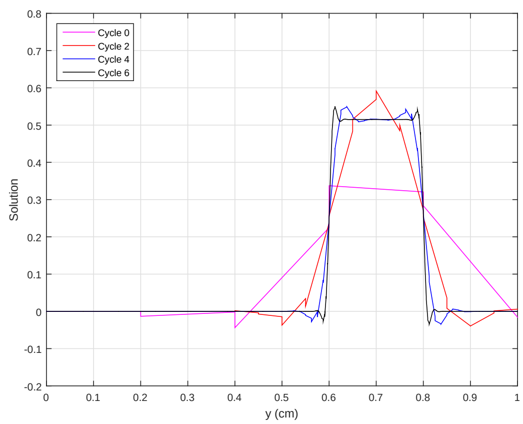


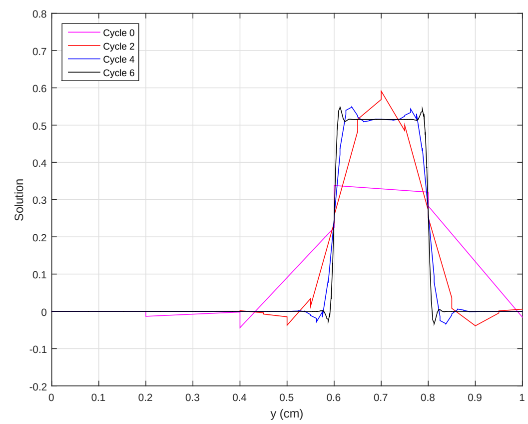
Figure 3.20: Initial mesh configuration for the searchlight problem before any refinement cycles.



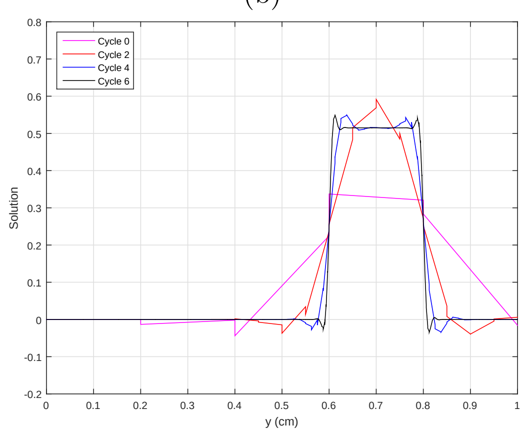
(a)



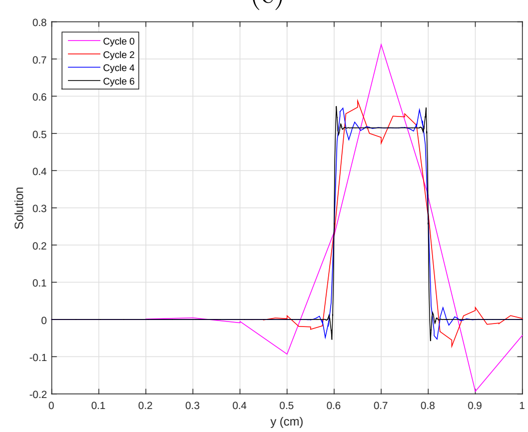
(b)



(c)



(d)



(e)

Figure 3.21: Exiting angular flux on the right boundary with uniform refinement using the (a) Wachspress basis functions, (b) PWL basis functions, (c) mean value basis functions, (d) linear maximum entropy coordinates and (e) quadratic serendipity maximum entropy coordinates.

4. DIFFUSION SYNTHETIC ACCELERATION FOR DISCONTINUOUS FINITE ELEMENTS ON UNSTRUCTURED GRIDS

4.1 Introduction

In this chapter, we analyze the Modified Interior Penalty (MIP) form of the diffusion equation as a discretization scheme for use with Diffusion Synthetic Acceleration (DSA) of the DFEM S_N transport equation on unstructured grids. Specifically, we wish to analyze its efficacy on massively-parallel computer architectures where scalability of solution times and memory footprints can become burdensome. This chapter is laid out in the following manner. The remainder of this Section will provide an overview of synthetic acceleration techniques as well as a review of DSA schemes. Section 4.2 provides the DSA methodologies that we will employ for 1-group and thermal neutron upscattering acceleration. We then present the discontinuous Symmetric Interior Penalty (SIP) form of the diffuion equation in Section 4.3 as well as the MIP variant that we will use for our DSA analysis in Section 4.4. The numerical procedures that we will use to solve the diffusion system of equations is given in Section 4.5. The theoretical Fourier analysis tool is given in Section 4.6. Results are provided in Section 4.7 and we finish the chapter with some concluding remarks in Section 4.8

4.1.1 Review of Diffusion Synthetic Accleration Schemes

cleanup the beginning here later...

The ability to efficiently invert the transport (streaming and collision) operator does not necessarily mean that transport solutions can be easily obtained. In general, radiation transport solutions are obtained iteratively. The simplest and widely-used method is a fixed-point scheme (*i.e.* richardson iteration) ubiquitously called source iteration (SI) in the transport community. Unfortunately, the iteration process of SI can converge arbitrarily slowly if the problem is optically thick [37]. This corresponds to long mean free paths for neutronics problems. This also corresponds to time steps and material heat capacities tending to infinity and zero, respectively, for thermal radiative transport (TRT) problems.

For these problem regimes in which solution is prohibitively slow, additional steps should be taken to speed up, or accelerate, solution convergence [37]. The most used methods to assist in solution convergence are often called synthetic acceleration techniques. These techniques were first introduced by Kopp [38] and Lebedev [39, 40, 41, 42, 43, 44, 45] in the 1960's. From Kopp's and Lebedev's work, Gelbard and Hageman then introduced two synthetic acceleration options for the low-order operator: diffusion and S_2 [46]. Their diffusion preconditioning led to efficient convergence properties on fine spatial meshes. Reed then showed that Gelbard and Hageman's diffusion preconditioning would yield a diverging system for coarse meshes [47]. At this point in time, no one knew if an unconditionally efficient acceleration method could be derived.

Then in 1976, Alcouffe proposed a remedy to Gelbard and Reed that he called diffusion synthetic acceleration (DSA) [48, 49, 50]. He showed that if you derived the diffusion operator consistently with the discretized transport operator, then SI could be accelerated with DSA in an efficient and robust manner. Larsen and McCoy then demonstrated that unconditional stability required that consistency be maintained in both spatial and angular discretization in their four-step procedure [51, 52]. However, Adams and Martin then showed that partially-consistent diffusion discretizations could effectively accelerate DFEM discretizations of the neutron transport equation [53]. Their modified-four-step procedure (M4S), based on Larsen and McCoy's work, was shown to be unconditionally stable for regular geometries, but divergent for unstructured multi-dimensional meshes [54]. In more recent years, alternate discretizations for the diffusion operator have been applied to unstructured multi-dimensional grids. These include the partially consistent Wareing-Larsen-Adams (WLA) DSA [55], the fully consistent DSA (FCDSA) [54], and the partially consistent MIP DSA [56, 57, 58].

Most recently, the partially consistent MIP DSA method has been shown to be an unconditionally stable acceleration method for the 2D DFEM transport equation on unstructured meshes. Wang showed that it acted as an effective preconditioner for higher-order DFEM discretizations on triangles [56, 57]. Turcksin and Ragusa then extended the work to arbitrary polygonal meshes [58]. The MIP diffusion operator is symmetric positive definite (SPD) and was shown to be efficiently invertible with preconditioned conjugate gradient (PCG) and advanced preconditioners such as algebraic multi-grid (AMG) [58].

4.1.2 Synthetic Acceleration Overview

Synthetic acceleration techniques have been widely used in the nuclear engineering community to improve solution convergence for prohibitively slow problems. We now provide a general framework for how synthetic acceleration methods are derived. We begin by expressing our neutron transport equation in the following form,

$$\left(\mathbf{A} - \mathbf{B}\right) \Psi = \mathbf{Q}, \tag{4.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are both linear operators, Ψ is the full angular flux solution in space, angle, and energy, and $\mathbf{Q}$ is the source or driving function. If we had the ability to efficiently invert $(\mathbf{A} - \mathbf{B})$ directly, then Ψ could be directly computed:

$$\Psi = \left(\mathbf{A} - \mathbf{B}\right)^{-1} \mathbf{Q}. \tag{4.2}$$

However, since in practice the discretized version of $(\mathbf{A} - \mathbf{B})$ is much more costly to directly invert than the discretized version of $\mathbf{A}$, we instead choose to iteratively solve for Ψ.

To compute Ψ, the following iterative system of equations is usually employed,

$$\mathbf{A}\Psi^{(k+1)} - \mathbf{B}\Psi^{(k)} = \mathbf{Q}, \tag{4.3}$$

where directly solving for $\Psi^{(k+1)}$ yields the following:

$$\Psi^{(k+1)} = \mathbf{A}^{-1}\mathbf{B}\Psi^{(k)} + \mathbf{A}^{-1}\mathbf{Q}. \tag{4.4}$$

For brevity, we define a new operator $\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}$ which is known as the iteration operator. The spectral radius, ρ, of this operator is simply the supremum of the absolute values of its eigenvalues. For this work, we assume that ρ is less than unity to guarantee convergence. We next define the residual, $r^{(k)}$, as the difference between two successive solution iterates,

$$r^{(k)} = \Psi^{(k)} - \Psi^{(k-1)}, \tag{4.5}$$

which can also be written as the following:

$$r^{(k)} = \mathbf{C}\Psi^{(k-1)}. \tag{4.6}$$

With the iteration operator, $\mathbf{C}$, and the residual for iterate k , $r^{(k)}$, defined, we can then write the true, converged solution in terms of the solution at iteration k and an infinite series of residuals:

$$\Psi = \Psi^{(k)} + \sum_{n=1}^{\infty} r^{(k+n)}. \tag{4.7}$$

Using both Eqs. (4.6) and (4.7), we can rewrite Eq. (4.7) using the iteration operator and the last residual,

$$\Psi = \Psi^{(k)} + \left(\mathbf{I} + \mathbf{C} + \mathbf{C}^2 + ...\right) \mathbf{C} r^{(k)}. \tag{4.8}$$

Since we have assumed that the spectral radius of $\mathbf{C}$ is less than unity, the infinite operator series of Eq. (4.8) converges to $(\mathbf{I} - \mathbf{C})^{-1} \mathbf{C}$. This means that we can succinctly write Eq. (4.8) as the following:

$$\Psi = \Psi^{(k)} + (\mathbf{I} - \mathbf{C})^{-1} \mathbf{C} r^{(k)}. \tag{4.9}$$

By using the definition of $\mathbf{C}$ along with some linear algebra, Eq. (4.9) becomes

$$\Psi = \Psi^{(k)} + (\mathbf{A} - \mathbf{B})^{-1} \mathbf{B} r^{(k)}. \tag{4.10}$$

We would like to use the results of Eq. (4.10) to immediately compute our exact transport solution, Ψ. However, this would require the inversion of $(\mathbf{A} - \mathbf{B})$ which we did not employ originally in Eq. (4.1) because of the difficulty. This means that, in its current form, Eq. (4.10) is no more useful to us than Eq. (4.1). This would then be an exercise in futility if we were restricted to only working with the $(\mathbf{A} - \mathbf{B})$ operator. Instead, suppose that we could define an operator, $\mathbf{W}$, that closely approximates $(\mathbf{A} - \mathbf{B})$ but it is easily invertible. If $\mathbf{W}$ efficiently approximates the slowest converging error modes of $(\mathbf{A} - \mathbf{B})$, then Eq. (4.10) can be modified to form a new iterative procedure.

The new iterative procedure begins by simply taking the half-iterate of Eq. (4.4) instead of its full version: $(k+1/2)$ instead of (k+1). This half-iterate has the form

$$\Psi^{(k+1/2)} = \mathbf{C}\Psi^{(k)} + \mathbf{A}^{-1}\mathbf{Q}. \tag{4.11}$$

We can then express the full-iterate by the suggestion of Eq. (4.10). Using the low-order operator, we express the full-iterate as the following,

$$\Psi^{(k+1)} = \Psi^{(k+1/2)} + \mathbf{W}^{-1}\mathbf{B}r^{(k+1/2)}, \tag{4.12}$$

where $r^{(k+1/2)} = \Psi^{(k+1/2)} - \Psi^{(k)}$. We can also express Eq. (4.12) in terms of just the previous iterate ,$\Psi^{(k)}$, and a new operator:

$$\Psi^{(k+1)} = \left[\mathbf{I} - \mathbf{W}^{-1}\left(\mathbf{A}-\mathbf{B}\right)\right]\mathbf{C}\Psi^{(k)} + \left(\mathbf{I}+\mathbf{W}^{-1}\mathbf{B}\right)\mathbf{A}^{-1}\mathbf{Q}. \tag{4.13}$$

Observe that as $\mathbf{W}$ more closely approximates $(\mathbf{A}-\mathbf{B})$, the operator $\mathbf{W}^{-1}\left(\mathbf{A}-\mathbf{B}\right)$ converges to the identity matrix, $\mathbf{I}$. This means that the spectral radius of this new iteration matrix will approach zero as $\mathbf{W}$ gets closer to $(\mathbf{A}-\mathbf{B})$ and therefore more quickly and efficiently converge to the true solution.

4.2 Diffusion Synthetic Acceleration Methodologies

The procedures outlined in Section 4.1.2 define a general methodology to perform synthetic acceleration on the transport equation. We could utilize any of the acceleration strategies that have been developed over the years including DSA, TSA, BPA, etc. The only difference arises in what form the low-order operator, $\mathbf{W}$, will take. We obviously are focusing on DSA for this dissertation work, and we do so by first describing in Section 4.2.1 a simple 1-group specification of the synthetic acceleration methodology just presented. We then present a pair of strategies that can be employed to accelerate thermal neutron upscattering in Section 4.2.2.

4.2.1 Simple 1-group DSA Strategy

$$\vec{\Omega}_m \cdot \vec{\nabla}\psi_m + \sigma_t \psi_m = c\frac{\sigma_t}{4\pi}\phi + \frac{Q}{4\pi} \tag{4.14}$$

Recall the operator form of the fully discretized transport equation as defined in Section 2.6.1,

$$\begin{aligned} \mathbf{L\Psi} &= \mathbf{M\Sigma\Phi} + \mathbf{Q} \\ \mathbf{\Phi} &= \mathbf{D\Psi} \end{aligned} , \tag{4.15}$$

where $\mathbf{L}$ is the total interaction and streaming operator, $\mathbf{M}$ is the moment-to-discrete operator, $\mathbf{D}$ is the discrete-to-moment operator, $\mathbf{\Sigma}$ is the scattering operator, and $\mathbf{Q}$ is the forcing function. In this case, we simply treat this discretized problem as only having 1 energy group. The functional form of the discretized moment-to-discrete and discrete-to-moment operators are

$$M_m \equiv \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m), \tag{4.16}$$

and

$$D_{p,n} \equiv \sum_{m=1}^{M} w_m Y_{p,n}(\vec{\Omega}_m)\Psi(\vec{\Omega_m}), \tag{4.17}$$

respectively. This means that the operation $\mathbf{DL}^{-1}$ corresponds to a full-domain transport sweep followed by the computation of the flux moments from the angular flux. We next apply our half-iterate and previous iterate indices on Eq. (4.15) to form the iteration procedure for our transport equation

$$\mathbf{L\Psi}^{(k+1/2)} = \mathbf{M\Sigma\Phi}^{(k)} + \mathbf{Q} \tag{4.18}$$

We can then use the $\mathbf{DL}^{-1}$ operator to present our transport equation of Eq. (4.18) in terms of just the half-iterate flux moments,

$$\mathbf{\Phi}^{(k+1/2)} = \mathbf{DL}^{-1}\mathbf{M\Sigma\Phi}^{(k)} + \mathbf{DL}^{-1}\mathbf{Q}. \tag{4.19}$$

4.2.2 DSA Acceleration Strategies for Thermal Neutron Upscattering

4.2.2.1 Multigroup Richardson Acceleration

$$\mathbf{L_{gg}}\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{G}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k)} + \mathbf{Q}_g \tag{4.20}$$

4.2.2.2 Two-Grid Acceleration and a Variant

The second acceleration methodology that we will investigate is the two-grid acceleration scheme devised by Adams and Morel [59]. They originally derived the scheme to accelerate 1D multigroup transport problems that are dominated by thermal neutron upscattering.

$$\mathbf{L_{gg}}\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{g}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k+1/2)} + \mathbf{M}\sum_{g'=g+1}^{G}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k)} + \mathbf{Q}_g \tag{4.21}$$

In operator form, the full solution and half-iterate equations are given by

$$\mathbf{L}\Psi = \mathbf{MS_D}\Phi + \mathbf{MS_U}\Phi + \mathbf{Q}, \tag{4.22}$$

and

$$\mathbf{L}\Psi^{(k+1/2)} = \mathbf{MS_D}\Phi^{(k+1/2)} + \mathbf{MS_U}\Phi^{(k)} + \mathbf{Q}, \tag{4.23}$$

respectively. $\mathbf{S_D}$ and $\mathbf{S_U}$ are the downscatter and upscatter portions of the scattering matrix, respectively. These matrices still contain all of the scattering moments; they are simply restricted in energy. By moving the downscattering portion to the left side of the equation, inverting the $\mathbf{L}$ operator, and applying the discrete-to-moment operator, $\mathbf{D}$, we can rewrite the iteration equation of Eq. (4.23) in terms of only the flux moments:

$$\left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{D}}\right] \Phi^{(k+1/2)} = \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{U}}\Phi^{(k)} + \mathbf{D}\mathbf{L}^{-1}\mathbf{Q} \tag{4.24}$$

By inverting the left-side operator, we directly solve for the half-iterate flux moments,

$$\begin{aligned} \Phi^{(k+1/2)} &= \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{D}}\right]^{-1} \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{U}}\Phi^{(k)} + \mathbf{b} \\ \mathbf{b} &= \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{D}}\right]^{-1} \mathbf{D}\mathbf{L}^{-1}\mathbf{Q} \end{aligned}, \tag{4.25}$$

where we reduced the driving function to $\mathbf{b}$ for brevity.

4.3 Symmetric Interior Penalty Form of the Diffusion Equation

So far, we have presented several strategies in Section 4.2 in which DSA can be used to accelerate both within-group scattering and upscattering. We have also simply stated that our low-order operator will be the diffusion equation. However, we have not presented the exact form of the diffusion equation that we will utilize. In Section 4.4, we present the full form of the modified interior penalty (MIP) form of the diffusion equation that we will use as our low-order operator for DSA calculations. However, we first present in this Section a more generalized version of the interior penalty form that we could use as a stand-alone solver for the standard diffusion equation: the symmetric interior penalty (SIP) form [60, 61, 62].

We begin our discussion of the SIP form by analyzing the standard form of the diffusion equation,

$$-\vec{\nabla}\cdot D(\vec{r})\vec{\nabla}\Phi(\vec{r})+\sigma\Phi(\vec{r})=Q(\vec{r}),\qquad \vec{r}\in\mathcal{D},\tag{4.26}$$

with Dirichlet boundary conditions

$$\Phi(\vec{r})=\Phi_0(\vec{r}),\qquad \vec{r}\in\partial\mathcal{D}^d,\tag{4.27}$$

Neumann boundary conditions

$$-D\partial_n\Phi(\vec{r})=J_0(\vec{r}),\qquad \vec{r}\in\partial\mathcal{D}^n,\tag{4.28}$$

and Robin boundary conditions

$$\frac{1}{4}\Phi(\vec{r})+\frac{D}{2}\partial_n\Phi(\vec{r})=J^{inc}(\vec{r}),\qquad \vec{r}\in\partial\mathcal{D}^r.\tag{4.29}$$

$$\Phi(\vec{r})+\frac{1}{\kappa}D\partial_n\Phi(\vec{r})=\Phi_0(\vec{r}),\qquad \vec{r}\in\partial\mathcal{D}^d,\qquad \kappa\gg 1\tag{4.30}$$

$$\begin{aligned}\left(D\vec{\nabla}b,\vec{\nabla}\Phi\right)_{\mathcal{D}}-\left\langle D\partial_n b,\Phi\right\rangle_{\partial\mathcal{D}^d}-\left\langle b,D\partial_n\Phi\right\rangle_{\partial\mathcal{D}^d}\\ +\left\langle\kappa b,(\Phi-\Phi_0)\right\rangle_{\partial\mathcal{D}^d}=-\left\langle D\partial_n b,\Phi_0\right\rangle_{\partial\mathcal{D}^d}\end{aligned}\tag{4.31}$$

$$\left\langle\kappa[\![b]\!],[\![\Phi]\!]\right\rangle_{E_h^i}+\left\langle[\![b]\!],\{\{D\partial_n\Phi\}\}\right\rangle_{E_h^i}+\left\langle\{\{D\partial_n b\}\},[\![\Phi]\!]\right\rangle_{E_h^i}=0\tag{4.32}$$

The mean value and the jump of the terms on a face are

$$\{\{\Phi\}\}\equiv\frac{\Phi^+ + \Phi^-}{2}\qquad\text{and}\qquad[\![\Phi]\!]\equiv\Phi^+ - \Phi^-,\tag{4.33}$$

respectively. The directionality of the terms across a face can be defined in negative, Φ^-, and positive, Φ^+ directions by their trace:

$$\Phi^{\pm} \equiv \lim_{s \to 0^{\pm}} \Phi(\vec{r} + s\vec{n}), \tag{4.34}$$

where the face's unit normal direction, $\vec{n}$, has been ar

Using Eqs. (4.31) and (4.32), the SIP form of the diffusion equation can be succinctly written as

$$a^{SIP}(b, \Phi) = b^{SIP}(b), \tag{4.35}$$

with the following bilinear matrix:

$$\begin{aligned} a^{SIP}(b,\Phi) = \Big(D\vec{\nabla} b, \vec{\nabla}\Phi\Big)_{\mathcal{D}} + \Big(\sigma b, \Phi\Big)_{\mathcal{D}} + \frac{1}{2}\Big\langle b, \Phi\Big\rangle_{\partial\mathcal{D}^r} \\ + \Big\langle \kappa_e^{SIP} [\![b]\!], [\![\Phi]\!]\Big\rangle_{E_h^i} + \Big\langle [\![b]\!], \{\{D\partial_n \Phi\}\}\Big\rangle_{E_h^i} + \Big\langle \{\{D\partial_n b\}\}, [\![\Phi]\!]\Big\rangle_{E_h^i}, \\ + \Big\langle \kappa_e^{SIP} b, \Phi\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle D\partial_n b, \Phi\Big\rangle_{\partial\mathcal{D}^d} \end{aligned} \tag{4.36}$$

and with the following linear right-hand-side:

$$\begin{aligned} b^{SIP}(b) = \Big(b, Q\Big)_{\mathcal{D}} - \Big\langle b, J_0\Big\rangle_{\partial\mathcal{D}^n} + 2\Big\langle b, J^{inc}\Big\rangle_{\partial\mathcal{D}^r}. \\ + \Big\langle \kappa_e^{SIP} b, \Phi_0\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle D\partial_n b, \Phi_0\Big\rangle_{\partial\mathcal{D}^d} \end{aligned} \tag{4.37}$$

As previously stated, the general penalty term, κ of Eqs. (4.30 - 4.32) needs to have sufficient positive measure to ensure stability. From previous investigations [56, 57, 58], we choose the penalty coefficient to be face dependent:

$$\kappa_e^{SIP} = \begin{cases} \frac{c}{2}\left(\frac{D^+}{h^+} + \frac{D^-}{h^-}\right) & e \in E_h^i \\ c\frac{D^-}{h^-} & e \in \partial\mathcal{D} \end{cases}, \tag{4.38}$$

for interior, E_h^i, and boundary, $\partial\mathcal{D}$, faces respectively. In Eq. (4.38), $h^{\pm}$ is the orthogonal projection of the face, e, into the cells defined by the trace in Eq. (4.34). Turcksin and Ragusa, [58], defined $h^{\pm}$ for 2D polygons, whose definitions can be seen in Table 4.1. The orthogonal projection for both triangles and quadrilaterals can be explicitly defined from simple geometric relationships. However, for polygons with > 4 faces, there is no explicit geometric relationship to define the orthogonal projection. Instead, the polygon is approximated as regular, and the orthogonal projection is no longer face-dependent. For polygons with an even number of faces greater than 4, the orthogonal projection is twice the apothem, which is the line segment between the polygon's center and the midpoint of each polygon's side. For odd number of faces greater than 4, the polygon's orthogonal projection becomes the sum of the apothem and the circumradius.

In a similar manner to the 2D orthogonal projections defined in Table 4.1, we define our choice for the extension of the orthogonal projections to 3D in Table 4.2. Like triangles and quadrilaterals in 2D, the orthogonal projections for tetrahedra and hexahedra can be explicitly defined from the volume equations for pyramids and parallelepipeds, respectively. For cells that are not tetrahedra or hexahedra, we introduce an approximation similar to 2D where we treat the cell as a regular polyhedron. In 3D there is no compact formula that can be given, unlike the definitions of the apothem and circumradius for 2D. Instead, we take the geometric limit of a polyhedra as the number of faces tends to infinity (a sphere). In this limiting case, the orthogonal projection simply becomes the sphere's diameter. We can then define the sphere's diameter with geometric information that would also be available to polyhedra by dividing a sphere's volume (the polyhedral volume) by its surface area (the sum of the areas of the polyhedral faces). While this leads to a gross approximation of the orthogonal projection for polyhedra that are not tetrahedra or hexahedra, it will provide appropriate geometric measure, especially for strictly convex polyhedra.

Table 4.1: Orthogonal projection, h, for different polygonal types: A_K is the area of cell K, L_e is the length of face e, and P_K is the perimeter of cell K.

Number of Vertices	3	4	> 4 and even	> 4 and odd
h	$2\frac{A_K}{L_e}$	$\frac{A_K}{L_e}$	$4\frac{A_K}{P_K}$	$2\frac{A_K}{P_K} + \sqrt{\frac{2A_K}{N_K \sin(\frac{2\pi}{N_K})}}$

Table 4.2: Orthogonal projection, h, for different polyhedral types: V_K is the volume of cell K, A_e is the area of face e, and SA_K is the surface area of cell K.

Number of Faces	4	6	otherwise
h	$3\frac{V_K}{A_e}$	$\frac{V_K}{A_e}$	$6\frac{V_K}{SA_K}$

4.3.1 Elementary Stiffness Matrices

In Eqs. (4.36) and (4.37), there are two additional elementary matrix types required other than those presented in Chapter 2. The first has the form of $\left(D\vec{\nabla}b, \vec{\nabla}\Phi\right)_K$ and is referred to as the stiffness matrix [15]. For a cell K, we define the stiffness matrix $\mathbf{S}$ as

$$\mathbf{S}_K = \int_K \vec{\nabla}\mathbf{b}_K \cdot \vec{\nabla}\mathbf{b}_K^T \, dr, \tag{4.39}$$

which has dimensionality (N_KxN_K). Just like the cell-wise elementary matrices presented in Chapter 2, it is possible that these integrals can be computed analytically, depending on the FEM basis functions used. However, for most of the 2D basis functions presented in Chapter 3, this cannot be done. Instead, we can again employ a numerical 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. Using this quadrature set, the stiffness matrix can be calculated by the following

$$\mathbf{S}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla}\mathbf{b}_K(\vec{x}_q) \cdot \vec{\nabla}\mathbf{b}_K^T(\vec{x}_q). \tag{4.40}$$

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

$$\mathbf{S}_K = \begin{bmatrix} \int_K \vec{\nabla}b_1 \cdot \vec{\nabla}b_1 & \dots & \int_K \vec{\nabla}b_1 \cdot \vec{\nabla}b_j & \dots & \int_K \vec{\nabla}b_1 \cdot \vec{\nabla}b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla}b_i \cdot \vec{\nabla}b_1 & \dots & \int_K \vec{\nabla}b_i \cdot \vec{\nabla}b_j & \dots & \int_K \vec{\nabla}b_i \cdot \vec{\nabla}b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla}b_{N_K} \cdot \vec{\nabla}b_1 & \dots & \int_K \vec{\nabla}b_{N_K} \cdot \vec{\nabla}b_j & \dots & \int_K \vec{\nabla}b_{N_K} \cdot \vec{\nabla}b_{N_K} \end{bmatrix}, \tag{4.41}$$

where an individual matrix entry is of the form:

$$S_{i,j,K} = \int_K \vec{\nabla}b_i \cdot \vec{\nabla}b_j. \tag{4.42}$$

4.3.2 Elementary Surface Gradient Matrices

The second new elementary matrix corresponds to the integrals of the product of the basis functions with their gradients on a given surface. For a given face f, these matrices are of the general form: $\left\langle D\partial_n b, \Phi \right\rangle_f$. These terms are analagous to the cell streaming matrix but are computed on the cell boundary with dimensionality $(d-1)$. Analyzing a single face, f, in cell K, the analytical surface gradient matrix is of the form,

$$\mathbf{N}_{f,K} = \int_f \vec{n}(s) \cdot \vec{\nabla}\mathbf{b}_K \, \mathbf{b}_K^T \, ds, \tag{4.43}$$

where the surface normal, $\vec{n}$, is directed outwards from cell K and is allowed to vary along the cell face. From the analytical integral, we can see that these matrices have dimensionality, $(N_K \text{x} N_K)$. We include the operation of the dot product between the outward normal and the basis function gradient for two reasons. First, it reduces the dimensionality of the matrices. Second, because the interior face terms in the SIP bilinear form are independent of the orientation of the normal unit vector along the face, we do not need to perform any additional handling of the face normals. We can see that the bilinear form is independent of the face normal orientation by observing the following relations:

$$\begin{aligned} \left\langle [\![u]\!], \{\{\partial_n v\}\} \right\rangle_f &= -\left\langle \{\{\vec{n}u\}\}, \{\{\vec{n}\partial_n v\}\} \right\rangle_f \\ \left\langle \{\{\partial_n u\}\}, [\![v]\!] \right\rangle_f &= -\left\langle \{\{\vec{n}\partial_n u\}\}, \{\{\vec{n}v\}\} \right\rangle_f \end{aligned}. \tag{4.44}$$

If the direction of the normal is changed from $\vec{n}$ to $-\vec{n}$ in Eq. (4.44), we can see that these terms are not modified.

Similar to the surface matrix defined in Chapter 2, it is possible that the gradients of the basis functions cannot be integrated analytically along a cell face. Using the same face-wise quadrature notation as before, $\left\{\vec{x}_q, w_q^f\right\}_{q=1}^{N_q^f}$, we can numerically calculate the surface gradient matrix for face f along cell K:

$$\mathbf{N}_{f,K} = \sum_{q=1}^{N_q^f} w_q^f \vec{n}(\vec{x}_q) \, \vec{\nabla}\mathbf{b}_K(\vec{x}_q) \, \mathbf{b}_K^T(\vec{x}_q). \tag{4.45}$$

In this case, the local face-wise surface gradient matrix for face f has the full matrix form,

$$\mathbf{N}_{f,K} = \begin{bmatrix} \int_f \vec{n}\cdot\vec{\nabla} b_1\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_1\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\cdot\vec{\nabla} b_i\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_i\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\cdot\vec{\nabla} b_{N_K}\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_{N_K}\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla} b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{4.46}$$

where an individual matrix entry is of the form:

$$N_{i,j,f,K} = \int_f \vec{n}\cdot\vec{\nabla} b_i\, b_j. \tag{4.47}$$

4.4 Modified Interior Penalty Form of the Diffusion Equation used for Diffusion Synthetic Acceleration Applications

$$a^{MIP}(b,\delta\Phi) = b^{MIP}(b), \tag{4.48}$$

with the following bilinear matrix:

$$\begin{aligned} a^{MIP}(b,\delta\Phi) &= \Big(D\vec{\nabla} b, \vec{\nabla}\delta\Phi\Big)_{\mathcal{D}} + \Big(\sigma b, \delta\Phi\Big)_{\mathcal{D}} \\ +\Big\langle \kappa_e^{MIP} [\![b]\!], [\![\delta\Phi]\!]\Big\rangle_{E_h^i} &+ \Big\langle [\![b]\!], \{\{D\partial_n \delta\Phi\}\}\Big\rangle_{E_h^i} + \Big\langle \{\{D\partial_n b\}\}, [\![\delta\Phi]\!]\Big\rangle_{E_h^i}, \\ +\Big\langle \kappa_e^{MIP} b, \delta\Phi\Big\rangle_{\partial\mathcal{D}^d} &- \frac{1}{2}\Big\langle b, D\partial_n\delta\Phi\Big\rangle_{\partial\mathcal{D}^d} - \frac{1}{2}\Big\langle D\partial_n b, \delta\Phi\Big\rangle_{\partial\mathcal{D}^d} \end{aligned} \tag{4.49}$$

and with the following linear right-hand-side:

$$b^{MIP}(b) = \Big(b, Q\Big)_{\mathcal{D}} + \Big\langle b, \delta J^{inc}\Big\rangle_{\partial\mathcal{D}^r}. \tag{4.50}$$

$$\kappa_e^{MIP} = \max\left(\kappa_e^{SIP}, \frac{1}{4}\right) \quad (4.51)$$

4.5 Solving the MIP Diffusion Problem

Equations (4.49) and (4.50) form the system matrix and right-hand-side, respectively, that we need to solve for a given DSA step. Just like the system of equations for the transport problem is too large to solve in a direct manner, we again employ an iterative scheme. Because the MIP bilinear form is SPD, we have natural recourse to use the simple Preconditioned Conjugate Gradient (PCG) method [20]. If the system of equations you are trying to solve for is SPD, then Conjugate Gradient (CG) is the most light-weight iterative method possible. It has a low memory footprint and is guaranteed to converge in $(N_{dof}/2)$ iterations, even if it is ill-conditioned. With a good preconditioner, then the iteration counts with PCG can be reduced substantially further. If we define $\mathbf{A}$ as the MIP system matrix to be inverted and $\mathbf{b}$ as the right-hand-side vector, then the CG algorithm acts by minimizing the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$. This is accomplished by taking successive operations of matrix-vector multiplications on conjugate krylov vectors, $\mathbf{p}_k$. The PCG algorithm performs one additional step by solving the equation $\mathbf{M}\mathbf{z_k} = \mathbf{r_k}$ at each CG iteration, where $\mathbf{M}$ is some preconditioner. We provide the simple pseudocode for PCG in algorithm 1.

There are several choices of preconditioners that can be employed with PCG [20]. Depending on the structure and conditioning of the matrix to be inverted, some simple preconditioners can be effective. We can decompose the MIP system matrix, $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$, into its strictly lower-triangular portion, $\mathbf{L}$, its strictly diagonal portion, $\mathbf{D}$, and its strictly lower-triangular portion, $\mathbf{L}^T$. The simplest preconditioner we could employ is Jacobi preconditioning, which is just the strictly diagonal portion of the system matrix: $\mathbf{M} = \mathbf{D}$. This preconditioner is effective for diagonallydominant matrices. The next preconditioner would be a simple Symmetric Successive Over-Relaxation (SSOR) method: $\mathbf{M}(\omega) = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}\mathbf{D}+\mathbf{L}\right)\mathbf{D}^{-1}\left(\frac{1}{\omega}\mathbf{D}+\mathbf{L}\right)^T$. The PCG algorithm can be simplified with this preconditioner choice by the Eisenstat trick [63]. The final simple preconditioner that we will consider is Incomplete LU Factorization (ILU). Instead of simply decomposing the system matrix, we factorize it into a unit-lower triangular portion, $\tilde{\mathbf{L}}$, and an upper triangular portion, $\tilde{\mathbf{U}}$. These factorized matrices then form our preconditioner: $\mathbf{M} = \tilde{\mathbf{L}}\tilde{\mathbf{U}}$. From this factorization, the preconditioning step to solve $\mathbf{M}\mathbf{z} = \mathbf{r}$ is accomplished by first solving $\tilde{\mathbf{L}}\mathbf{y} = \mathbf{r}$, followed by $\tilde{\mathbf{U}}\mathbf{z} = \mathbf{y}$. While this leads to a second preconditioning step, the factorized matrices are easy to invert since they are triangular.

Besides Jacobi, SSOR and ILU preconditioning, we also seek to investigate multigrid (MG) methods to invert the MIP system matrix.

Algorithm 1 PCG Algorithm

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
for k=1,2,... **do**
    Solve: $\mathbf{M}\mathbf{z}_{i-1} = \mathbf{r}_{i-1}$
    $\rho_{i-1} = \mathbf{r}_{i-1}^T\mathbf{z}_{i-1}$
    if $k = 1$ **then**
        $\mathbf{p}_i = \mathbf{z}_{i-1}$
    else
        $\mathbf{p}_i = \mathbf{z}_{i-1} + \frac{\rho_{i-1}}{\rho_{i-2}}\mathbf{p}_{i-1}$
    end if
    $\mathbf{q}_i = \mathbf{A}\mathbf{p}_i$
    $\alpha_i = \frac{\rho_{i-1}}{\mathbf{p}_i^T\mathbf{q}_i}$
    $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
    $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\mathbf{q}_i$
    if $\frac{||\mathbf{r}_i||}{||\mathbf{b}||} < \text{tol}$ **then**
        Exit
    end if
end for

4.6 Fourier Analysis

With the acceleration methodologies and diffusion discretization scheme described in detail, we now present the Fourier Analysis tool. Fourier Analysis is commonly used to analyze the performance characteristics of acceleration schemes for the transport equation. It is a powerful tool because it allows us to express the iteration error in terms of a fourier series. Then, because of the orthogonality of the terms in the series, each fourier mode can be analyzed independently. If these modes were not independent, then we would have no ability to simultaneously solve the infinite spectrum of fourier modes.

$$\begin{aligned} \Psi_m(\vec{r}) &= \hat{\Psi}_m e^{i\vec{\lambda}\cdot\vec{r}} \\ \Phi(\vec{r}) &= \hat{\Phi} e^{i\vec{\lambda}\cdot\vec{r}} \end{aligned} \tag{4.52}$$

where $i = \sqrt{-1}$

iteration matrices here...

spectral radius = largest eigenvalue here...

For this work, all fourier analysis was performed in MATLAB. All the eigenmodes corresponding to a fourier wave number for a given iteration matrix can be easily computed with MATLAB's built-in *eig* function. The maximum eigenvalue is found over the wave number space by use of the Nelder-Mead simplex algorithm [64]. We stress that some sort of minimization algorithm must be employed because some problem configurations can have extremely narrow local maxima. These difficult to find local maxima can correspond to the global maximum which is our desired spectral radius that we wish to compute.

We illustrate the necessity for a minimization algorithm in Figure 4.2. We have modeled a single 2D square mesh cell with dimensions $X = 1$ and $Y = 1$. The total

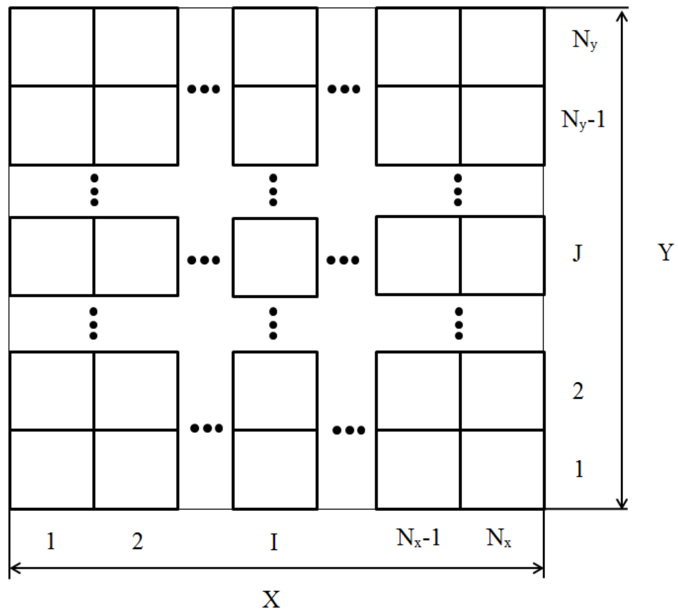
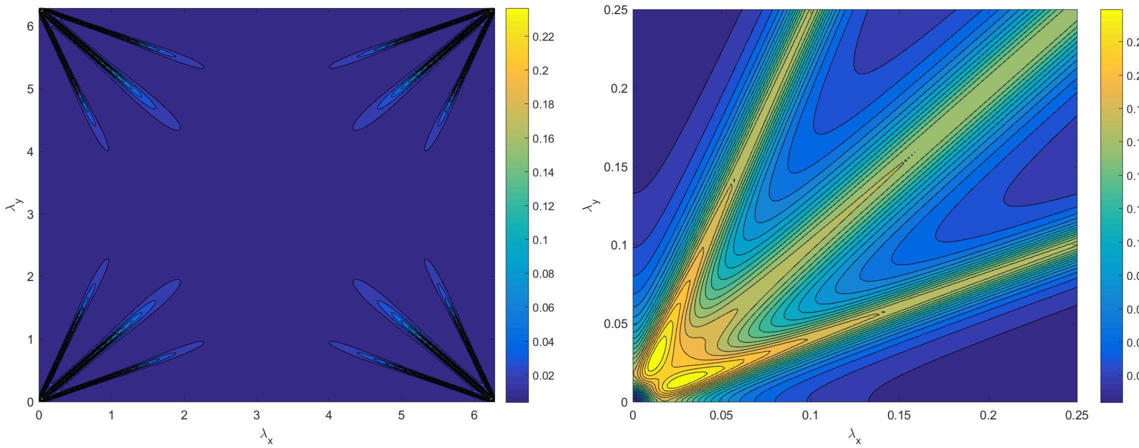


Figure 4.1: Fourier domain for 2D quadrilateral cells or an axial slice of 3D hexahedral cells in a regular grid.

cross section, σ_t, is set to 10^{-2} and the scattering ratio, c, set to 0.9999. We use the $S4$ level-symmetric quadrature set. The left image of Figure 4.2 has the 2D fourier wave number span the full domain space of $[\lambda_x, \lambda_y] = [0, 2\pi]^2$. The right image zooms in on the wave number ranging: $[\lambda_x, \lambda_y] = [0, 1/4]^2$. From the right image, we can see two extremely narrow local maxima. We can qualitatively ascertain that these local maxima would be difficult to find if we had simply laid a grid of wave number points over $[0, 2\pi]^2$. Chang and Adams presented an even more extreme example of a difficult to find global maximum using transport synthetic acceleration [65].

4.7 Numerical Results

We now present all results



(a) (b)

Figure 4.2: 2D fourier wave form for MIP, with 1 square cell with 1e-2 mfp, with the PWL coordinates, with the LS4 quadrature, and where the wave numbers range from: (a) $[\lambda_x, \lambda_y] = [0, 2\pi]^2$ and (b) $[\lambda_x, \lambda_y] = [0, 1/4]^2$.

4.7.1 Transport Solutions in the Thick Diffusive Limit

We present our first numerical example by demonstrating that the various polygonal finite element basis sets satisfy the thick diffusion limit.

$$\vec{\Omega} \cdot \vec{\nabla} \Psi + \sigma_t \Psi = \frac{\sigma_s}{4\pi} \sigma_t + \frac{Q_0}{4\pi} \tag{4.53}$$

As the transport problem becomes more optically thick, the total mean free paths of the neutroncs increases. In the thick diffusion limit, the domain mean free path approaches infinity. If we fix the physical dimensions of the problem to some finite value, then we can scale the cross sections and the source term to reflect the properties of the thick diffusion limit. In the limit the total and scattering cross sections tend to infinity whilc the absorption cross section and the source term tend to zero. If we introduce a scaling parameter, ϵ,

$$
\begin{aligned}
\sigma_t &\rightarrow \frac{\sigma_t}{\epsilon} \\
\sigma_a &\rightarrow \epsilon\sigma_t \\
\sigma_s &\rightarrow \left(\frac{1}{\epsilon} - \epsilon\right)\sigma_t \\
\frac{Q_0}{4\pi} &\rightarrow \epsilon\frac{Q_0}{4\pi}
\end{aligned}
\tag{4.54}
$$

Inserting these scaled cross sections and source term into Eq. (4.53) leads to following scaled transport equation:

$$
\vec{\Omega} \cdot \vec{\nabla}\Psi + \frac{\sigma_t}{\epsilon}\Psi = \sigma_t\left(\frac{1}{\epsilon} - \epsilon\right)\frac{\Phi}{4\pi} + \epsilon\frac{Q_0}{4\pi}. \tag{4.55}
$$

We can also use the scaled terms of Eq. (4.54) to give the corresponding scaled diffusion equation. If we take the 0th and 1st moments of Eq. (4.55) in the usual way and assume that the P1 terms obey Fick's Law, then the scaled diffusion equation is

$$
\epsilon\vec{\nabla} \cdot \frac{1}{3\sigma_t}\vec{\nabla}\Phi + \epsilon\sigma_t\Phi = \epsilon Q_0. \tag{4.56}
$$

One can immediately see that Eq. (4.56) does not truly scale because there is an ϵ for each term. This is the desired behavior we want to see from the diffusion equation because, as $\epsilon \rightarrow 0$, the transport equation will converge to its diffusive limit and satisfy a diffusion equation.

For the sake of analysis, we seek to

$$
\vec{\Omega} \cdot \vec{\nabla}\Psi + \frac{1}{\epsilon}\Psi = \left(\frac{1}{\epsilon} - \epsilon\right)\frac{\Phi}{4\pi} + \frac{\epsilon}{4\pi}, \tag{4.57}
$$

and

$$\frac{\epsilon}{3}\nabla^2\Phi + \epsilon\Phi = \epsilon, \tag{4.58}$$

respectively.

4.7.2 SIP used as a Diffusion Solver

We first wish to know how an interior penalty form of the diffusion equation will perform on unstructured polyhedral grids. The SIP diffusion formulation has previously been analyzed for use as a DFEM diffusion solver for unstructured 2D polygonal grids [61]. The MIP DSA form has also been successfully utilized for unstructured 2D polygonal grids [58]. Here, we first seek to extend the efficacy of the SIP form as a diffusion solver on polyhedral mesh cellls. We will do this by analyzing the following two problem types:

1. An exactly-linear solution to determine if linear basis functions will span the solution space;

2. The Method of Manufactured Solutions (MMS) to test basis function convergence rates.

The polyhedral mesh types that we employ for this analysis are presented in Section 4.7.2.1. Next, the exactly-linear solution analysis is performed in Section 4.7.2.2. Finally, the MMS analysis to confirm the second order convergence rates of the 3D PWL basis functions is presented in Section 4.7.2.3.

4.7.2.1 Geometry Specification for the SIP Problems

To analyze the SIP diffusion form on 3D polyhedral grids, we will utilize many of the 2D polygonal grids that were used in Section 3.4.1. For this analysis we will resuse the cartesian, ordered-triangular, polygonal sinusoidal, and polygonal-z meshes. We will also use purely-randomized polygonal grids formed from Voronoi mesh generation as outlined in Section 2.6.2.2. To form the needed 3D polyhedral grids, we will take these 2D grids and simply extrude the meshes in the z-dimension.

Figure 4.3 provides the 2D mesh types that will be utilized in this analysis. Figure 4.4 then provides the same meshes after they have been extruded in the z-dimension. We note that it will not simply be these exact grids that are employed. For the MMS analysis in Section 4.7.2.3, various refinements of these meshes will be utilized.

4.7.2.2 Exactly-Linear Solution

We first test SIP by enforcing a sytem that yields an exactly-linear diffusion solution. Linear finite elements should then theoretically fully-span the solution space. We can achieve this mathematically by setting the cross-section and right-hand-source terms to zero, $\sigma = Q = 0$. Robin boundary conditions are imposed on opposite faces in 1 dimension, with homogeneous Neumann boundary conditions on all other faces. If the Robin boundaries are chosen in the y-direction, with $y \in (0, L)$, then the analytical solution for the problem will be

$$\Phi(x, y, z) = \frac{4J^{inc}}{L + 4D} \left(L + 2D - y \right), \tag{4.59}$$

with the following boundary conditions in the y-direction:

$$\begin{aligned} \Phi - 2D\partial_y \Phi =& 4J^{inc}, \qquad && \forall (x, z), y = 0 \\ \Phi + 2D\partial_y \Phi =& 0, && \forall (x, z), y = L \end{aligned} . \tag{4.60}$$

In Eqs. (4.59) and (4.60), D is once again the standard diffusion coefficient and J^{inc} is the incoming current on the $y = 0$ boundary. For this analysis, we choose D to be 2, J^{inc} to be 9, and L to be 1. Using Eq. (4.59), our solution has a value of 20 at $y = 0$ and linearly-decreases to 16 at $y = 1$.

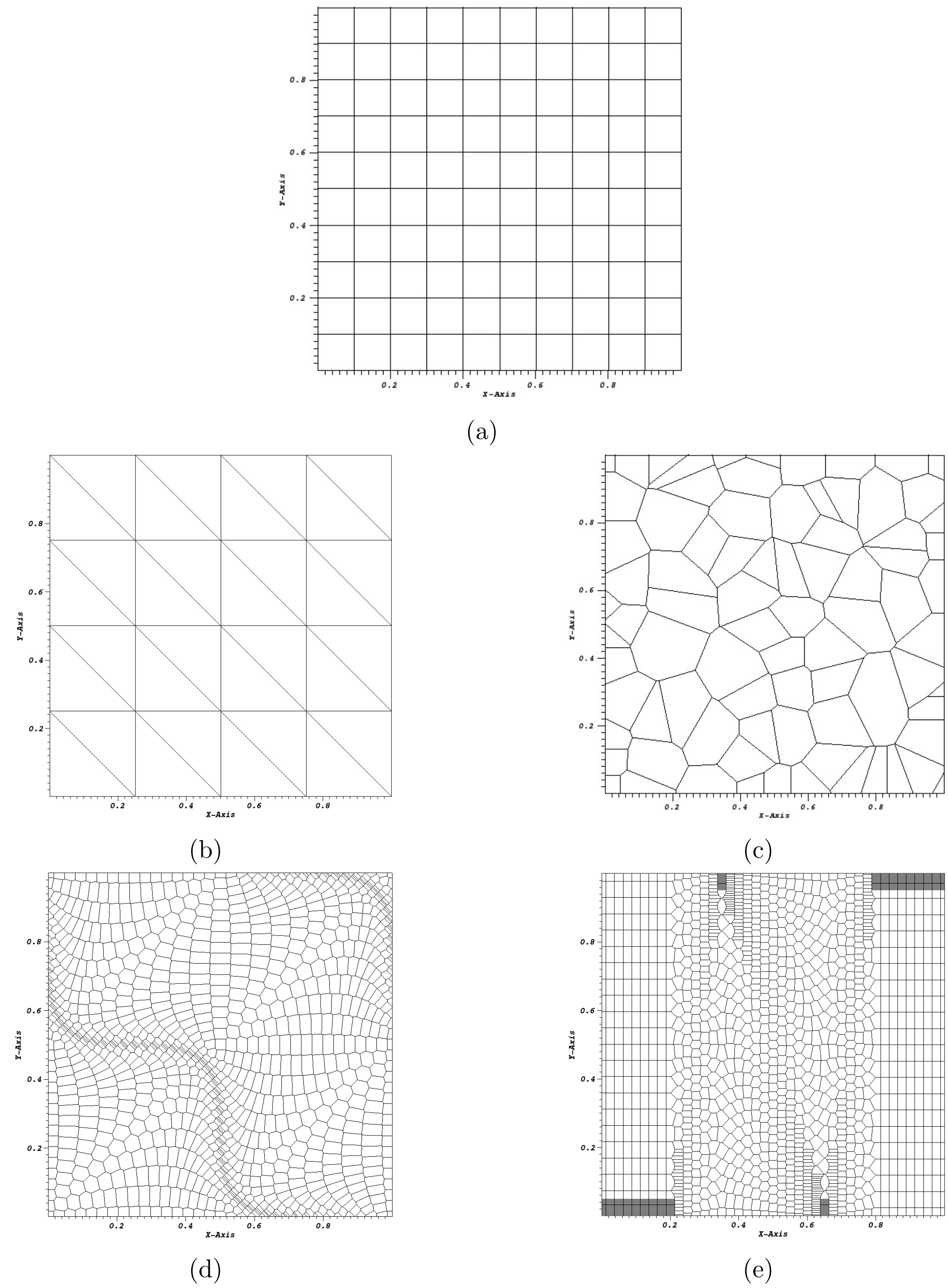
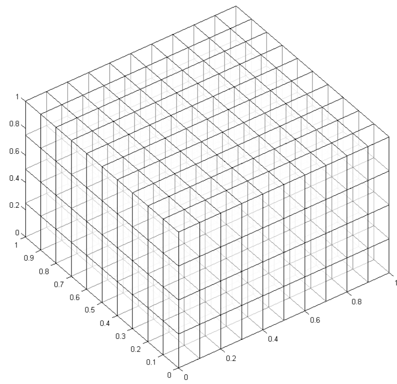
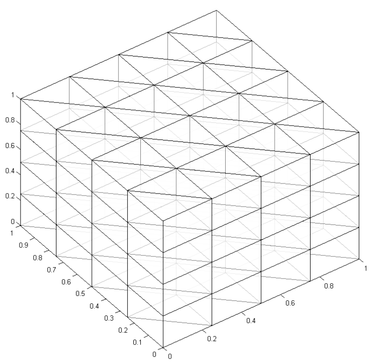


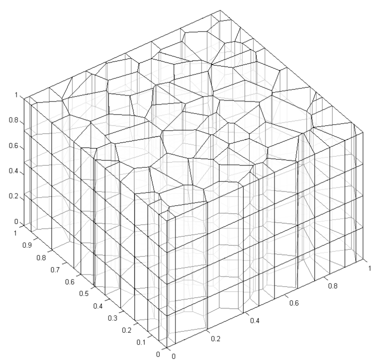
Figure 4.3: 2D grids to be extruded of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.



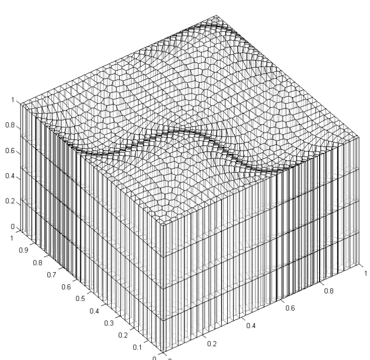
(a)



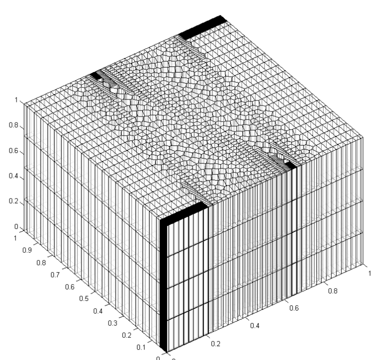
(b)



(c)



(d)



(e)

Figure 4.4: Extrusion of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

Using the 3D PWL basis functions, which were previously used for SIP on 2D polygons [61], the linear solutions are presented in Figure 4.5 at the midplane axial slice of $z = L/2$. We can see from the exact contour lines in the plots that SIP can capture an exactly-linear solution even on some highly distorted polyhedral grids.

4.7.2.3 Method of Manufactured Solutions

We next test to see if the SIP diffusion form can capture the appropriate second order convergencerates with the 3D PWL basis functions. These basis functions were previously shown to capture the appropriate convergence rates for the DGFEM transport equation on 3D hexahedral grids [29].

$$\Phi^{quad}(x, y, z) = x(1 - x)y(1 - y)z(1 - z) \tag{4.61}$$

$$\Phi^{gauss}(x, y, z) = \Phi^{quad}(x, y, z) \exp(-(\vec{r} - \vec{r}_0) \cdot (\vec{r} - \vec{r}_0)^T) \tag{4.62}$$

4.7.3 1 Group DSA Analysis

4.7.3.1 2D Homogeneous Medium Case

4.7.3.2 3D Homogeneous Medium Case

4.7.3.3 Periodic Horizontal Interface Problem

When DSA is applied to multidimensional problems (2D and 3D), the preconditioning of the transport operators can degrade in the presence of heterogeneous configurations with large material discontinuities [66]. The Periodic Horizontal Interface (PHI) problem is considered a litmus test for heterogeneous DSA techniques. This problem consists of horizontal strips of alternating optically thick and optically thin materials that are 1 cell in depth. We define σ_1 as the optically thick total cross

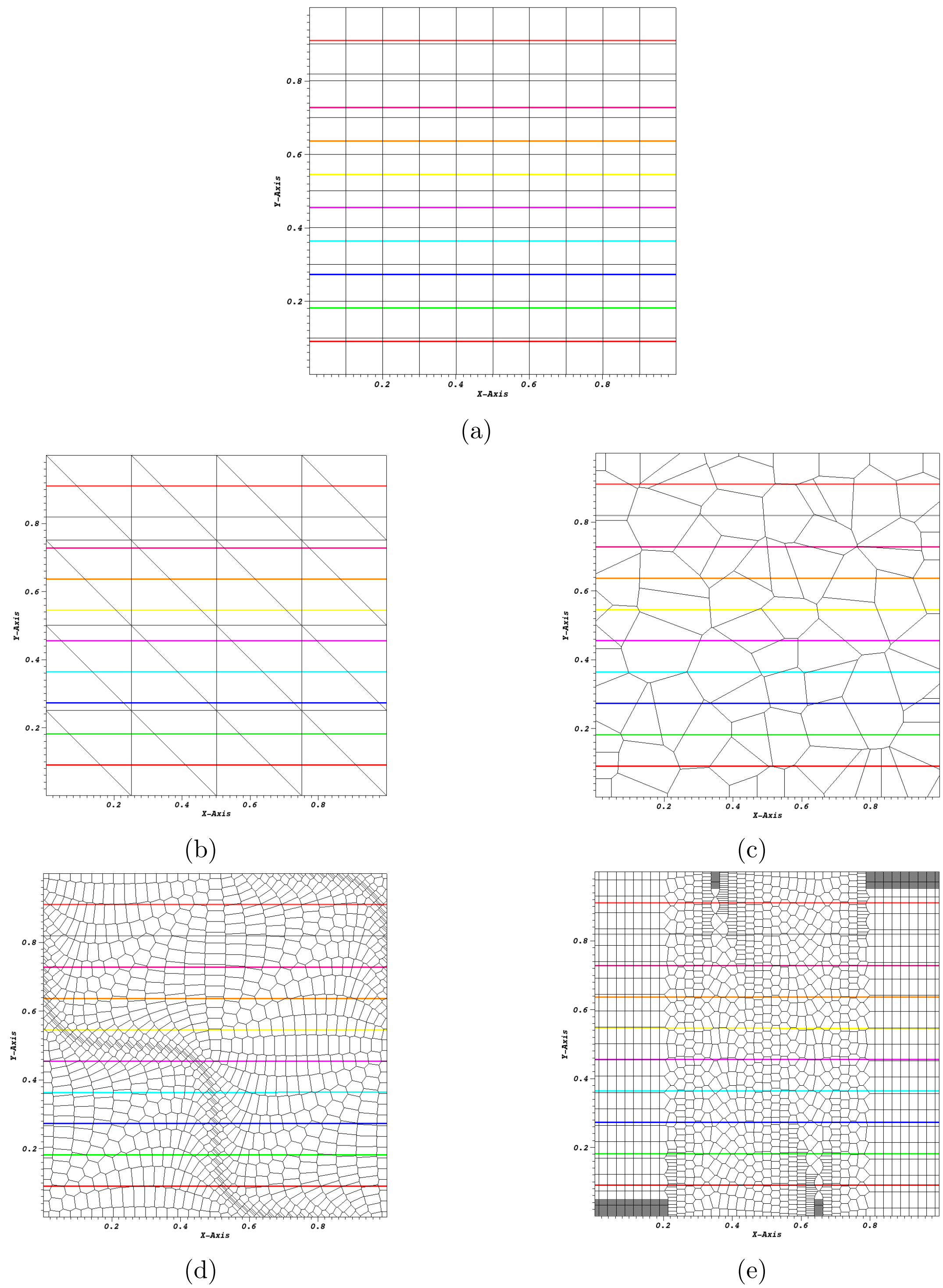


Figure 4.5: Axial slice showing the contours for the linear solution of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

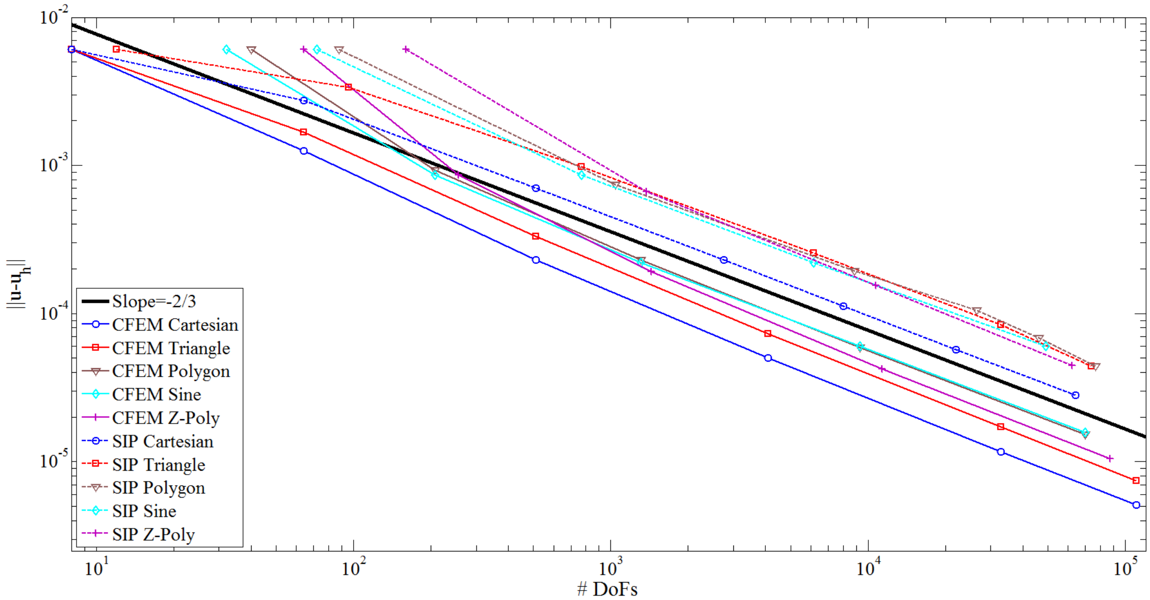


Figure 4.6: blah

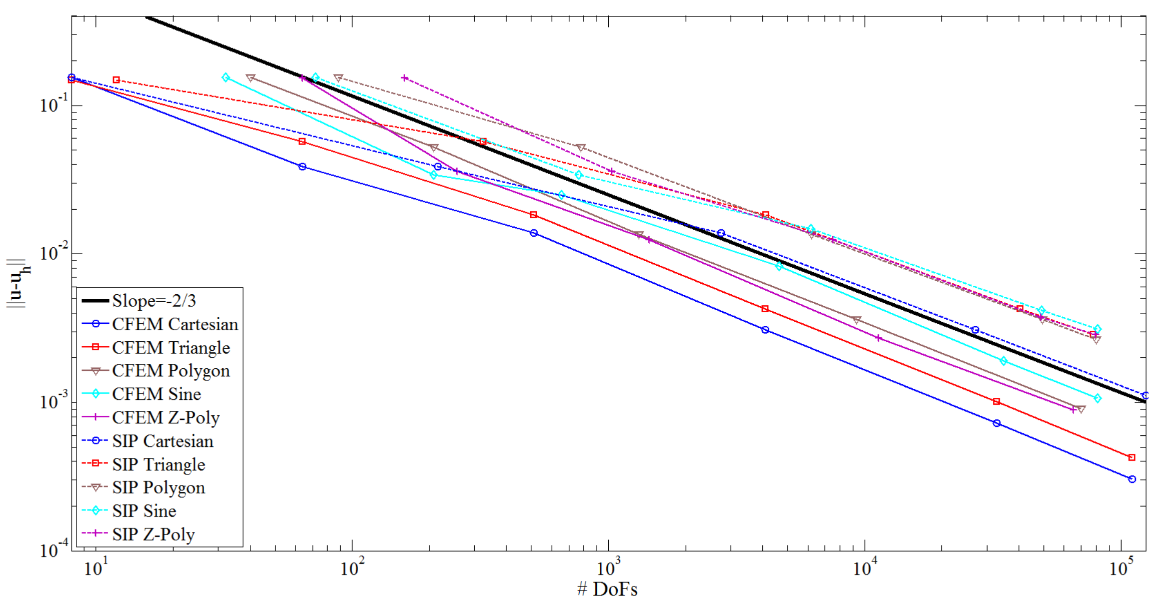


Figure 4.7: blah

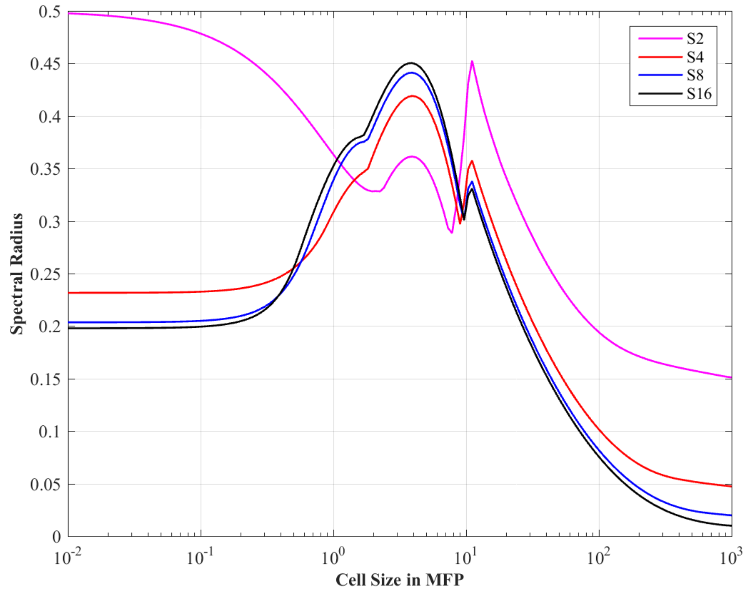


Figure 4.8: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear Wachspress coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

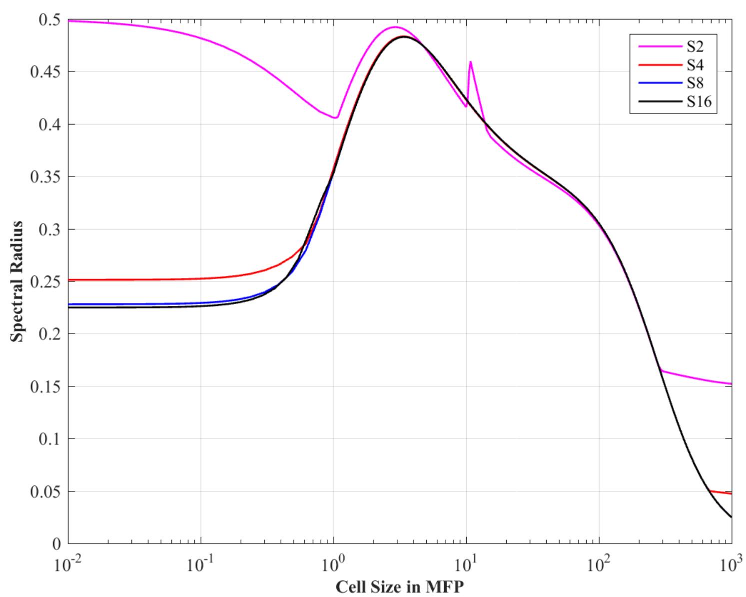


Figure 4.9: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear PWL coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

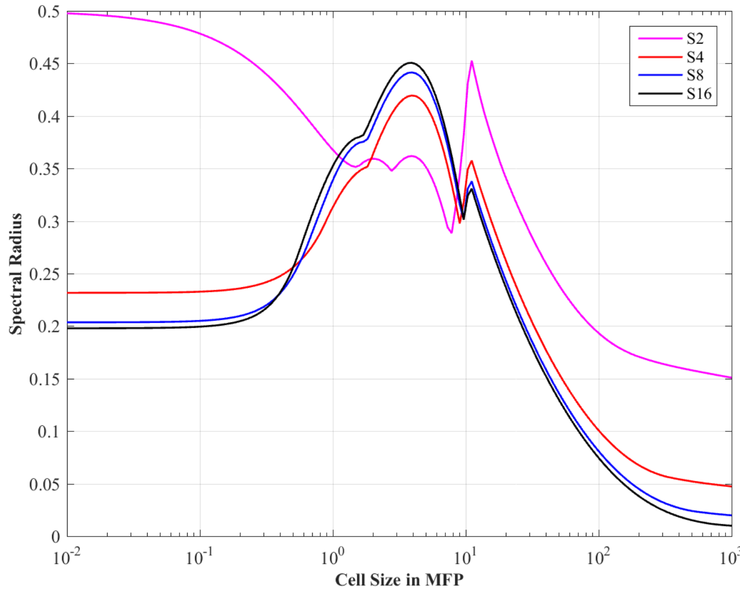


Figure 4.10: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear mean value coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

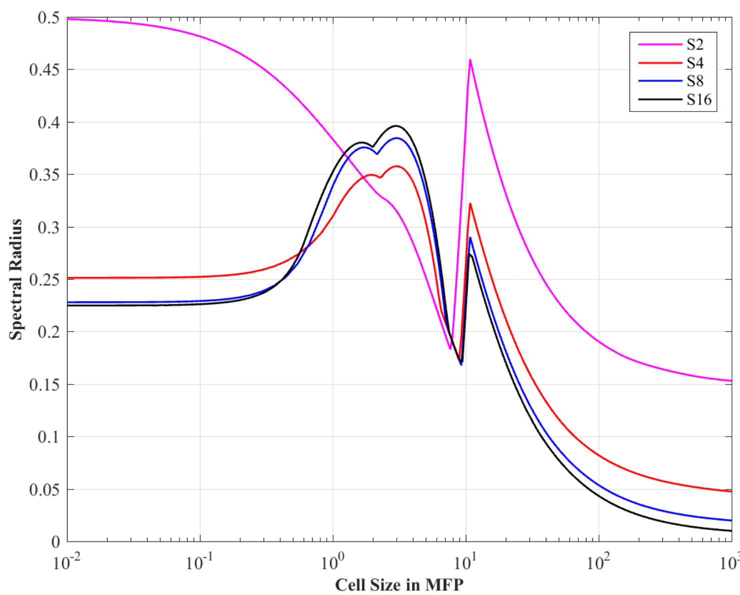
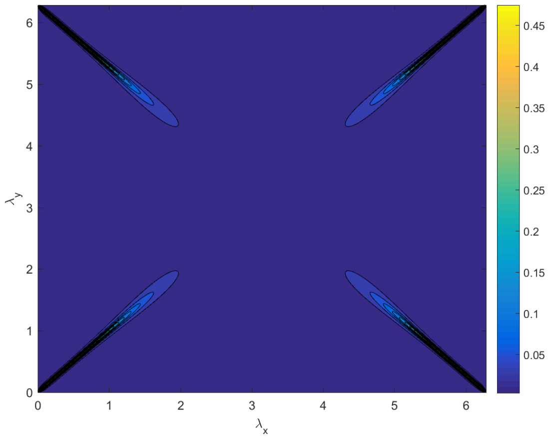
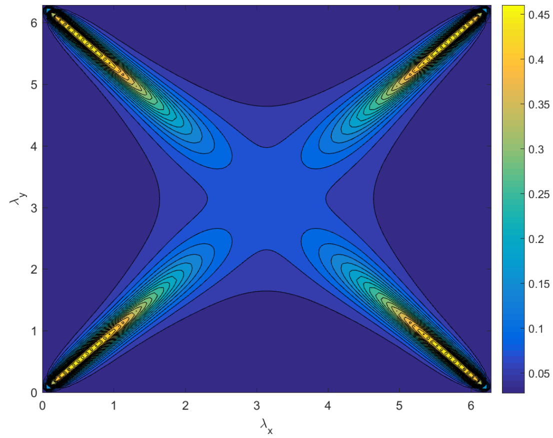


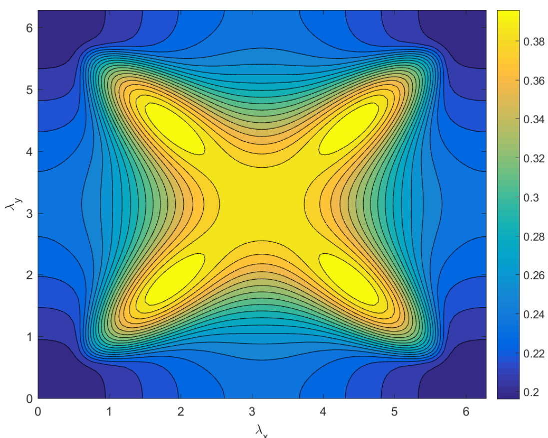
Figure 4.11: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear maximum entropy coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.



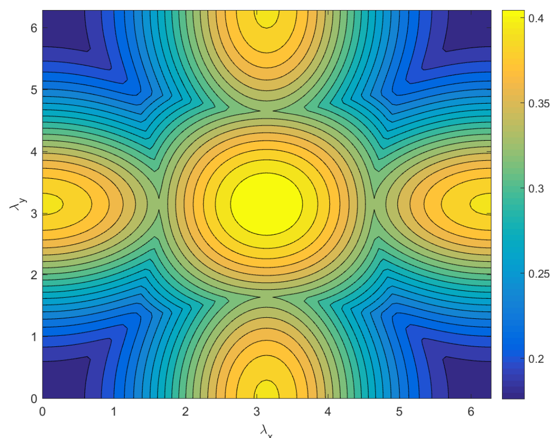
(a) 10^{-2} mfp



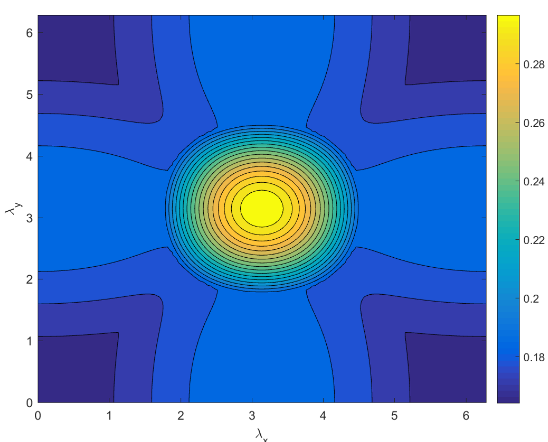
(b) 10^{-1} mfp



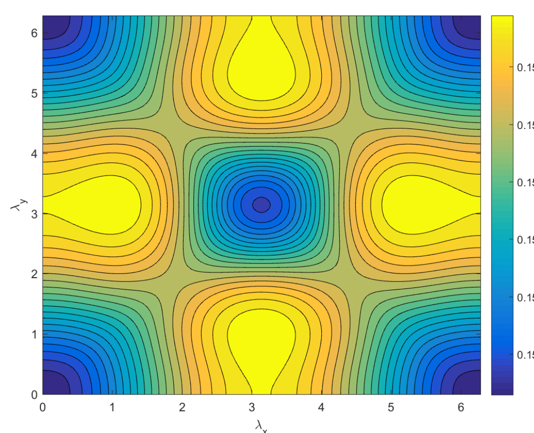
(c) 10^{0} mfp



(d) 10^{1} mfp

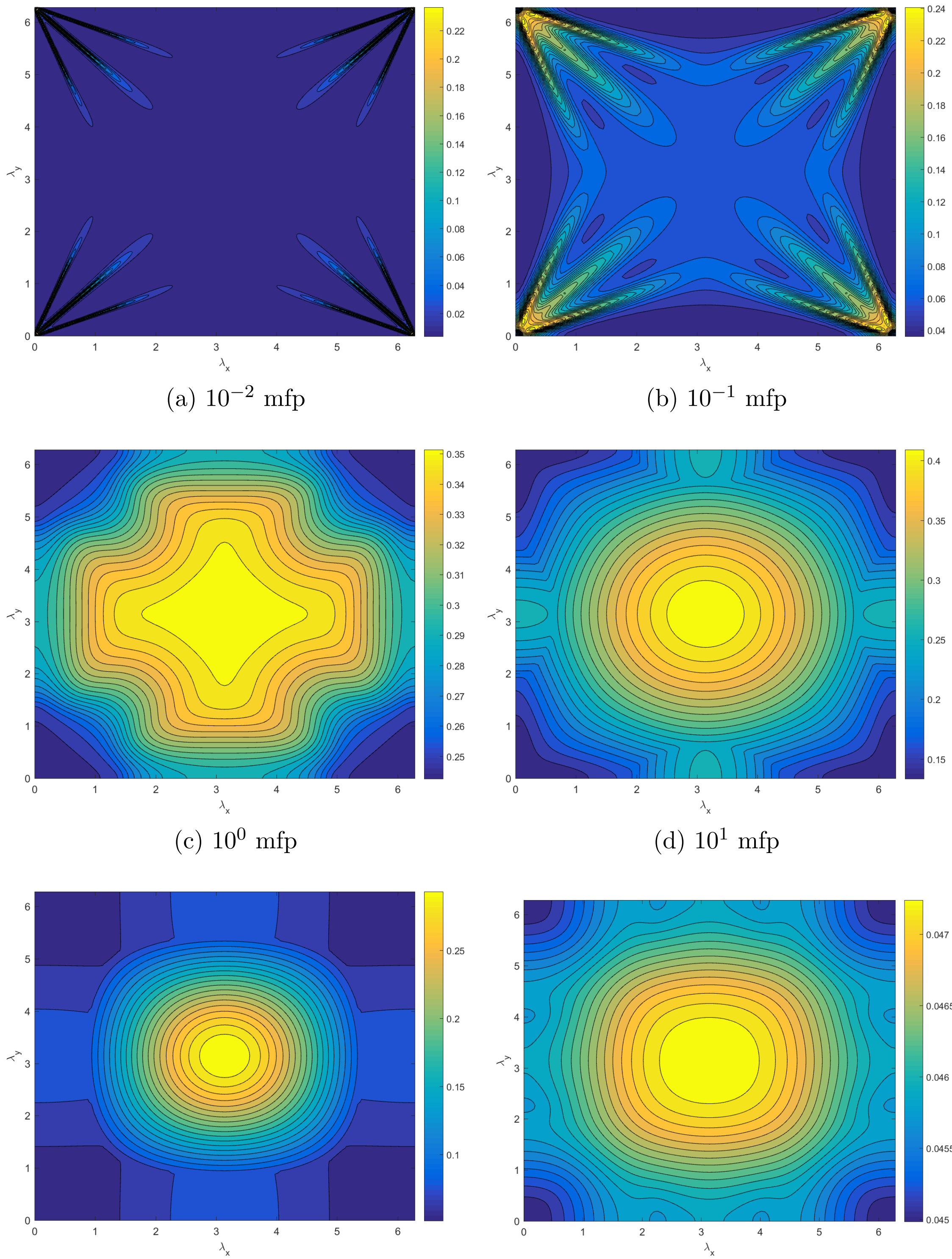


(e) 10^{2} mfp



(f) 10^{3} mfp

Figure 4.12: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS2 quadrature.



(a) 10^{-2} mfp

(b) 10^{-1} mfp

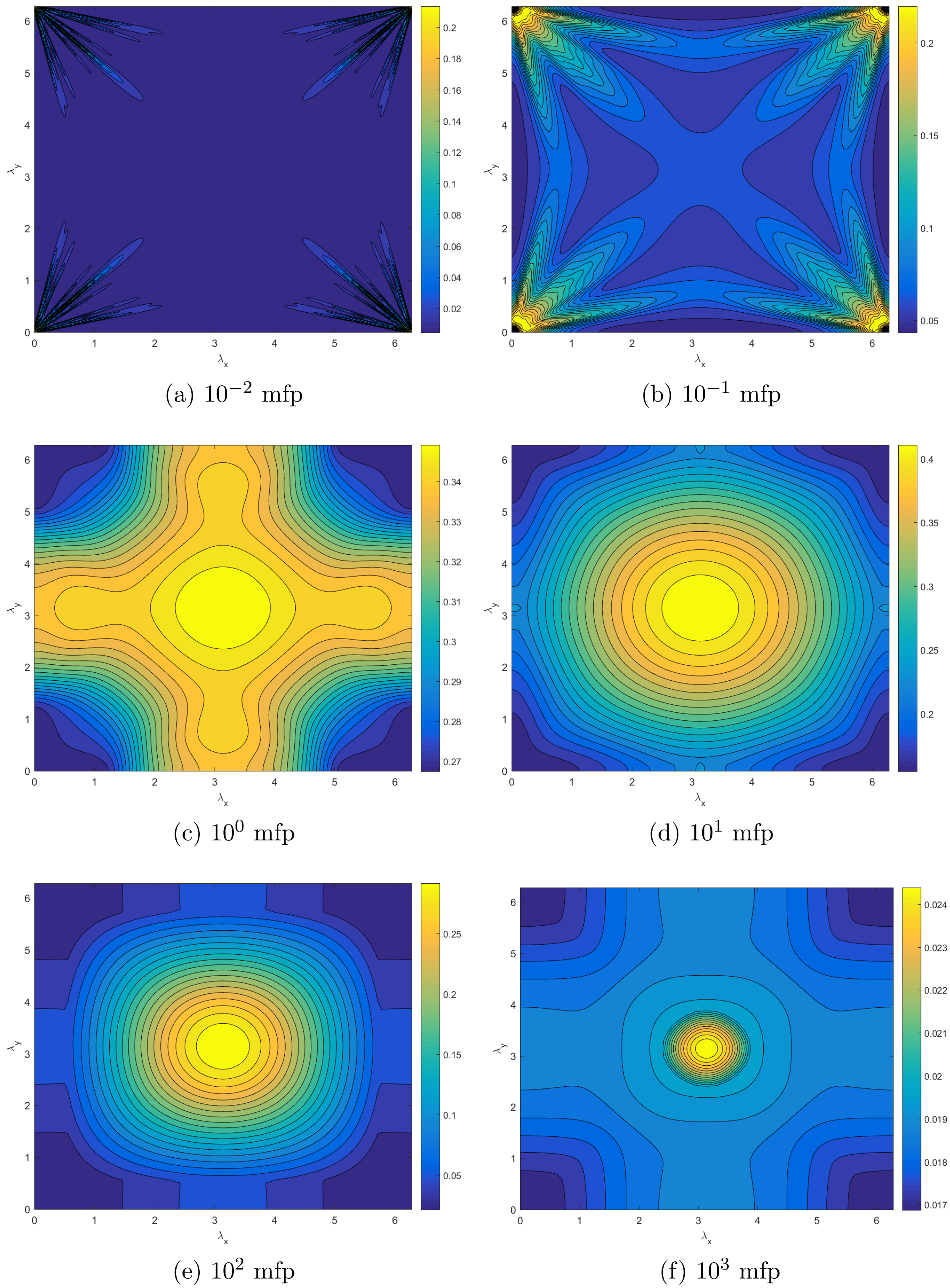
(c) 10^{0} mfp

(d) 10^{1} mfp

(e) 10^{2} mfp

(f) 10^{3} mfp

Figure 4.13: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS4 quadrature.



(a) 10^{-2} mfp (b) 10^{-1} mfp

(c) 10^{0} mfp (d) 10^{1} mfp

(e) 10^{2} mfp (f) 10^{3} mfp

Figure 4.14: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS8 quadrature.

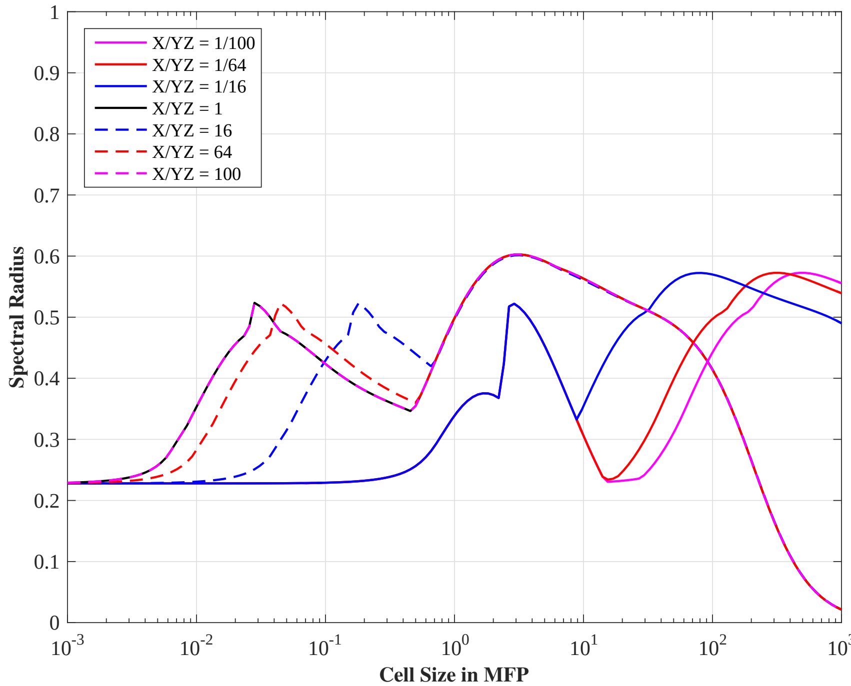


Figure 4.15: Fourier spectral radii for MIP with 3D PWL coordinates with different aspect ratios and $c = 1$.

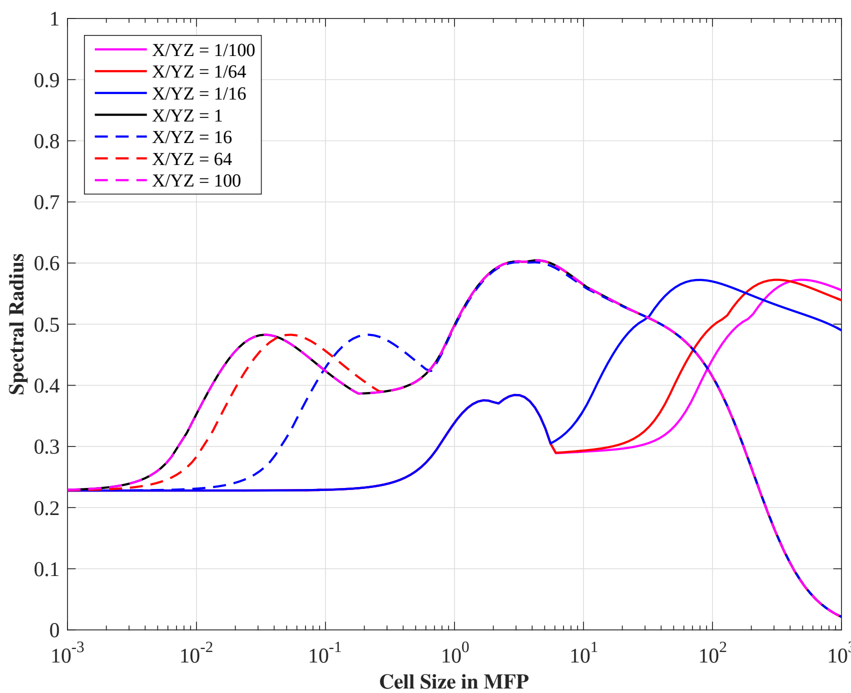


Figure 4.16: Fourier spectral radii for MIP with 3D PWL coordinates with different aspect ratios and $c = 4$.

section and σ_2 as the optically thin total cross section. We then define a tuning parameter, σ, and allow the region cross sections to have the following form:

$$\begin{aligned} \sigma_1 &= \sigma \\ \sigma_2 &= \frac{1}{\sigma} \end{aligned} . \tag{4.63}$$

We can see that increasing the value of σ will increase the magnitude of difference between the two region cross sections. Therefore, as σ grows large, the material discontinuities will grow which could potentially reduce the performance of our DSA scheme.

From the analysis presented in Section 4.7.3.1, we showed that our different 2D linear and quadratic basis functions were robust and stable, even for mesh cells with large aspect ratios. For this PHI analysis, we concentrate on analyzing just the linear PWL coordinates as our basis functions. Just like before, we will also examine different level-symmetric quadrature sets and their effects problems with varying optical thickness. The optical thickness and diffusivity of the problem is increased by varying both σ and the scattering ratio, c. This study was conducted with the sequence, $\sigma = [10, 20, 40, 80, 160, 320, 640]$, and with following scattering ratios: $c = [0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999]$.

The full results of this PHI analysis are presented in Tables 4.3 - 4.6 for the LS2, LS4, LS8, and LS16 quadratures, respectively. From these tables, we can see that MIP DSA loses its effectiveness as the heterogeneity and overall diffusivity of the problem increases. The theoretical spectral radii are greater than any of the values presented for the homogeneous case from Figure 4.9. This result is true even for the example with the smallest heterogeneity and smallest diffusivity ($\sigma = 10$ and $c = 0.9$). We can gain some more knowledge of the DSA degradation by observing the eigenvalue dependency on the fourier wave numbers for our problems. Figure

Table 4.3: Spectral radius for the 2D PHI problem with the PWL basis functions and LS2 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.77091	0.89908	0.91279	0.91417	0.91431	0.91432
20	0.82038	0.94425	0.95859	0.96003	0.96018	0.96019
40	0.84803	0.96525	0.97982	0.98129	0.98144	0.98145
80	0.86188	0.97491	0.98955	0.99102	0.99117	0.99119
160	0.87134	0.98003	0.99409	0.99557	0.99571	0.99573
320	0.87833	0.98353	0.99626	0.99774	0.99789	0.99790
640	0.88187	0.98533	0.99732	0.99880	0.99894	0.99896

4.17 provides the eigenvalue distribution based off the fourier wave numbers for the different quadrature sets for $\sigma = 10$ and $c = 0.9$. Figure 4.18 then provides the same information for $\sigma = 640$ and $c = 0.9999$.

4.7.3.4 Performance of MIP DSA with Adaptive Mesh Refinement

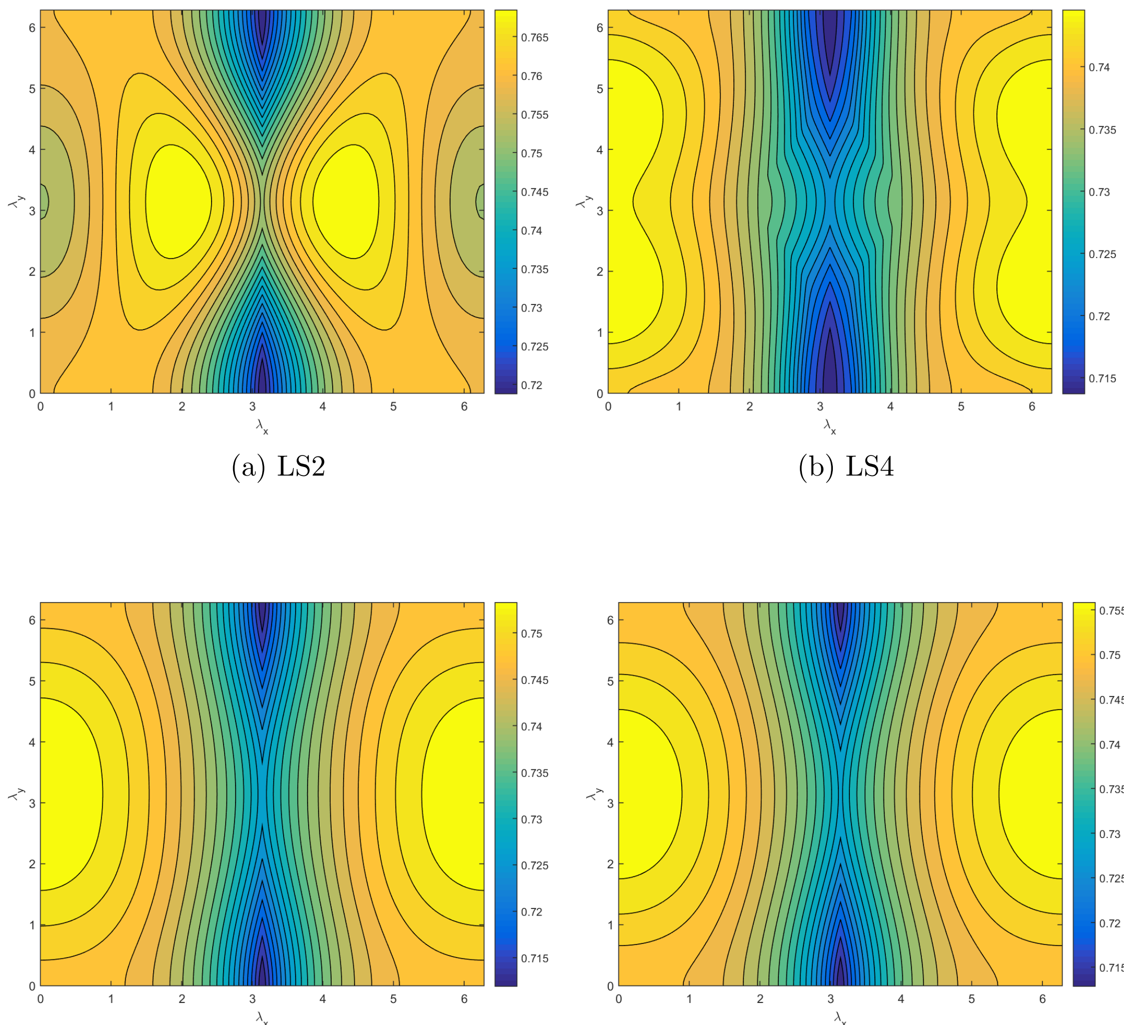
For the final theoretical analysis of DSA with the MIP form, we analyze the acceleration performance when AMR is utilized on a sufficiently optically thick transport with material discontinuities. The 2D transport problem that will be examined is similar to the Iron-Water problem [67]. It was modified by Wang and Ragusa for use with higher-order basis functions on triangular meshes with hanging nodes [56]. We will reexamine their work on degenarate polygonal grids and not use hanging nodes. The complete geometric desciption of our problem including boundary conditions and material distributions is given in Figure 4.19. The material properties for each

Table 4.4: Spectral radius for the 2D PHI problem with the PWL basis functions and LS4 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.74609	0.87284	0.88970	0.89148	0.89166	0.89167
20	0.81135	0.93358	0.95001	0.95179	0.95197	0.95199
40	0.84353	0.96104	0.97612	0.97781	0.97799	0.97800
80	0.85865	0.97342	0.98785	0.98943	0.98960	0.98961
160	0.86998	0.97913	0.99335	0.99482	0.99497	0.99499
320	0.87767	0.98307	0.99593	0.99739	0.99753	0.99755
640	0.88166	0.98500	0.99717	0.99863	0.99878	0.99879

Table 4.5: Spectral radius for the 2D PHI problem with the PWL basis functions and LS8 quadrature.

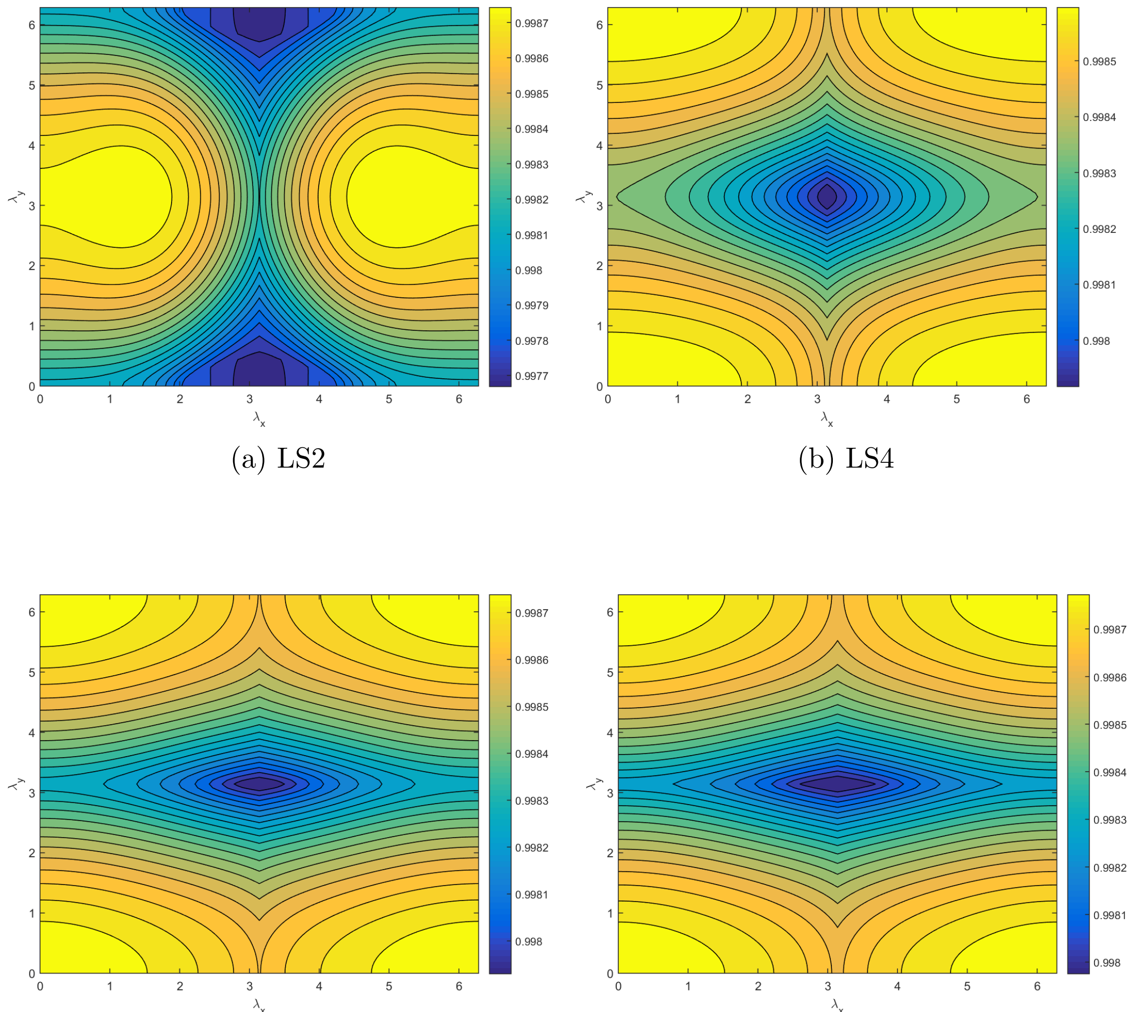
	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.75532	0.87917	0.89780	0.89989	0.90010	0.90012
20	0.81754	0.93694	0.95380	0.95581	0.95602	0.95604
40	0.84681	0.96337	0.97793	0.97969	0.97988	0.97990
80	0.86032	0.97459	0.98898	0.99042	0.99057	0.99058
160	0.86980	0.98007	0.99394	0.99540	0.99554	0.99556
320	0.87795	0.98354	0.99623	0.99768	0.99783	0.99784
640	0.88204	0.98524	0.99732	0.99878	0.99892	0.99894



(a) LS2 (b) LS4

(c) LS8 (d) LS16

Figure 4.17: Fourier wave number distribution for the 2D PHI problem with $\sigma = 10$ and $c = 0.9$ and different level-symmetric quadratures.



(a) LS2

(b) LS4

(c) LS8

(d) LS16

Figure 4.18: Fourier wave number distribution for the 2D PHI problem with $\sigma = 640$ and $c = 0.9999$ and different level-symmetric quadratures.

Table 4.6: Spectral radius for the 2D PHI problem with the PWL basis functions and LS16 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.75796	0.88052	0.89888	0.90097	0.90119	0.90121
20	0.81903	0.93790	0.95455	0.95651	0.95671	0.95673
40	0.84758	0.96387	0.97842	0.98015	0.98033	0.98035
80	0.86070	0.97485	0.98924	0.99069	0.99083	0.99085
160	0.87025	0.98029	0.99407	0.99553	0.99567	0.99569
320	0.87817	0.98365	0.99629	0.99775	0.99790	0.99791
640	0.88216	0.98530	0.99735	0.99881	0.99896	0.99897

Table 4.7: Material definitions and physical properties for the Iron-Water problem.

Region	σ_t (cm^{-1})	c	S (cm^{-3}sec^{-1})
I	1.0	0.90	1.0
II	1.5	0.96	0.0
III	1.0	0.30	0.0

region which include the total cross section, scattering ratio, and source strength are given in Table 4.7. Scattering is isotropic.

4.7.4 Scalability of the MIP DSA Preconditioner

So far, we have presented a detailed theoretical analysis of DSA preconditioning with the MIP diffusion form. We have shown that the different 2D and 3D basis functions provided in Chapter 3 are robust and stable, even on mesh cells with

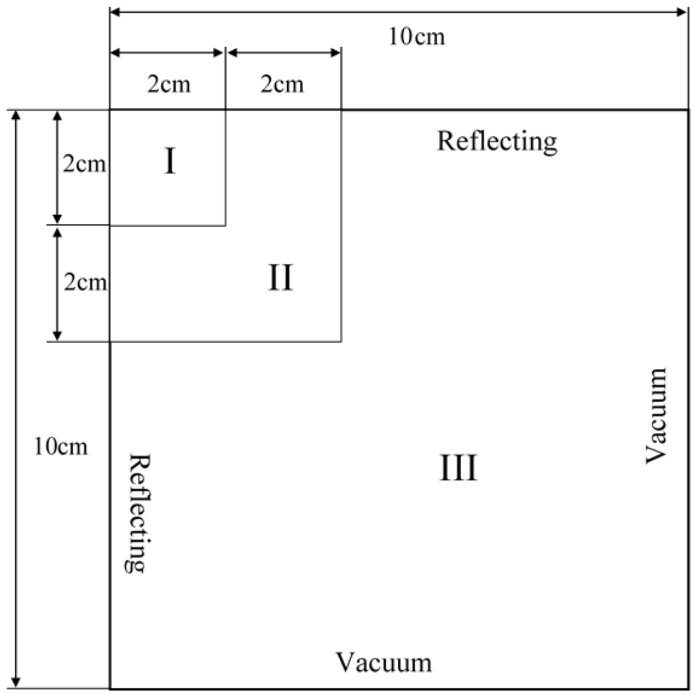


Figure 4.19: Geometry description for the Iron-Water problem.

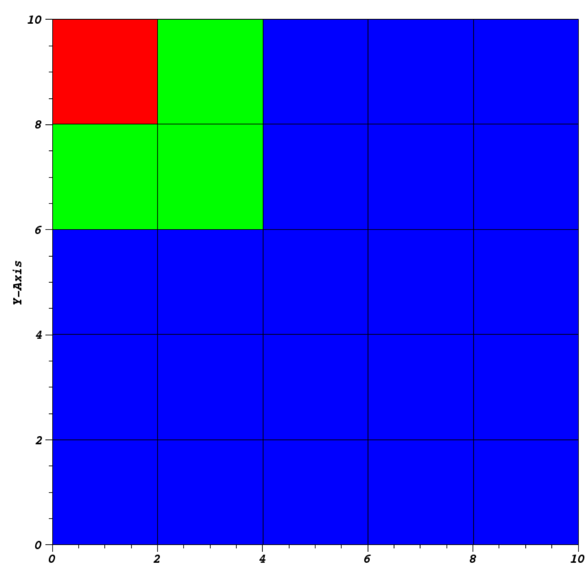
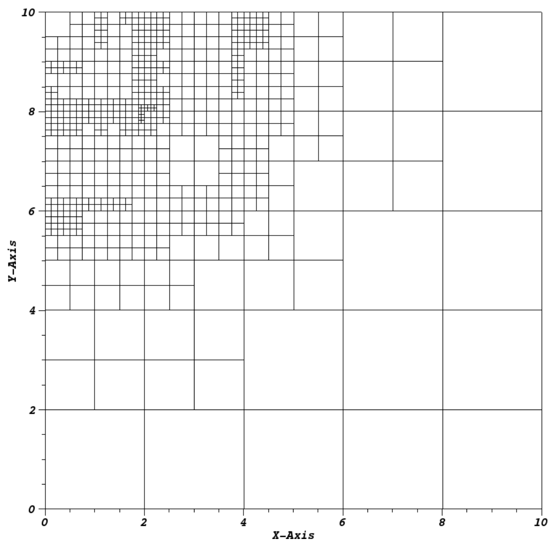
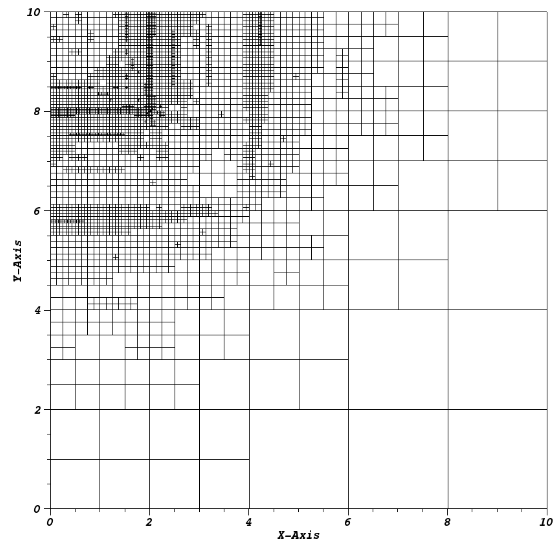


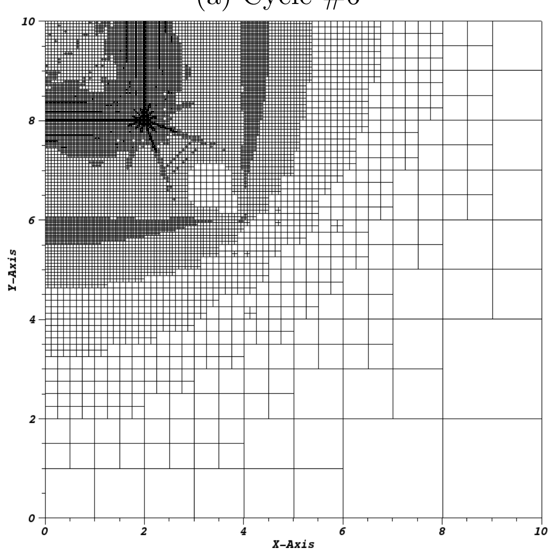
Figure 4.20: Initial mesh for the Iron-Water problem.



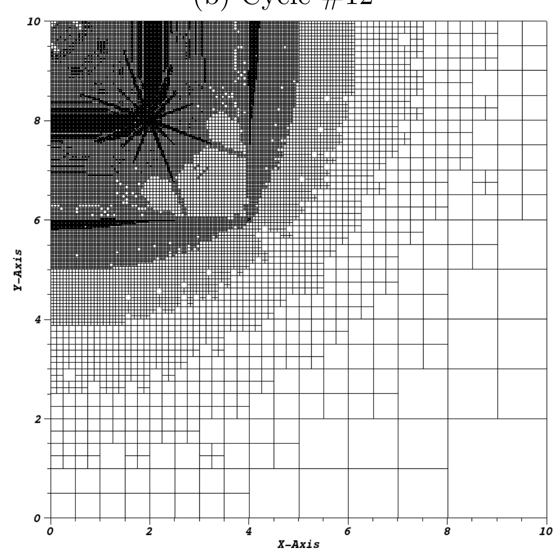
(a) Cycle #6



(b) Cycle #12

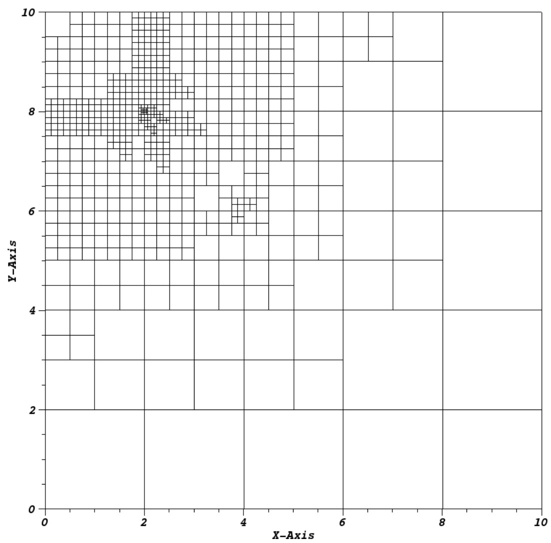


(c) Cycle #18

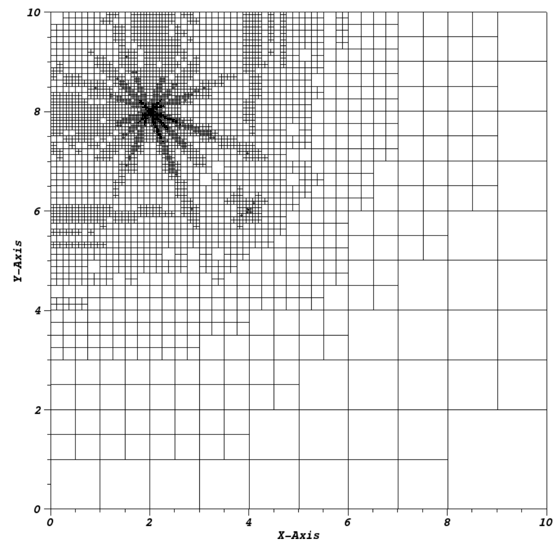


(d) Cycle #24

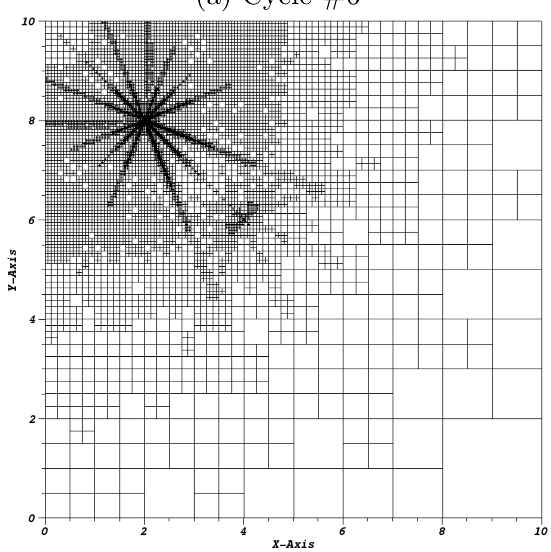
Figure 4.21: Meshes for the Iron-Water problem using the linear PWL coordinates and LS4 quadrature.



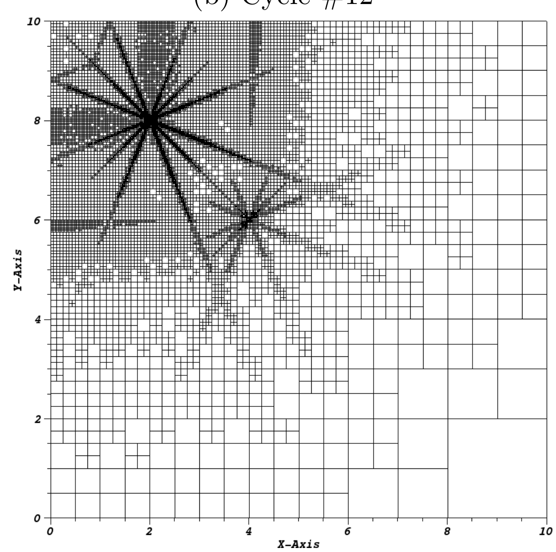
(a) Cycle #6



(b) Cycle #12

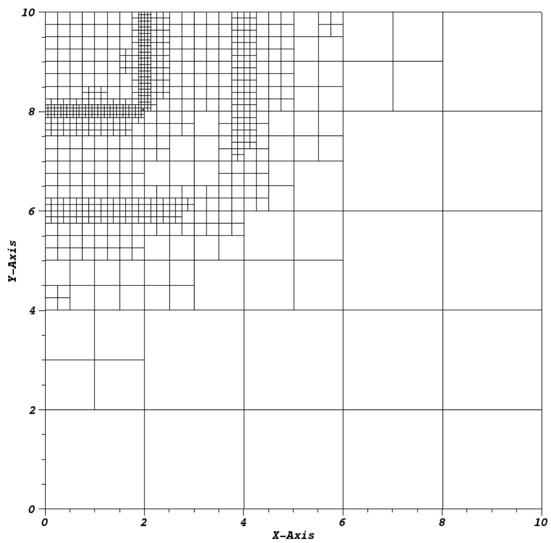


(c) Cycle #18

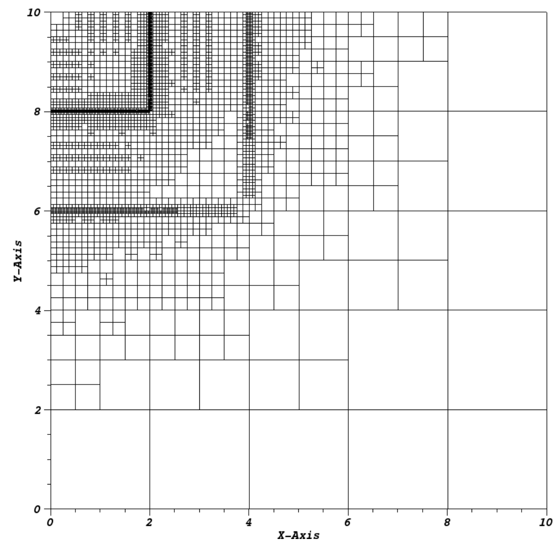


(d) Cycle #24

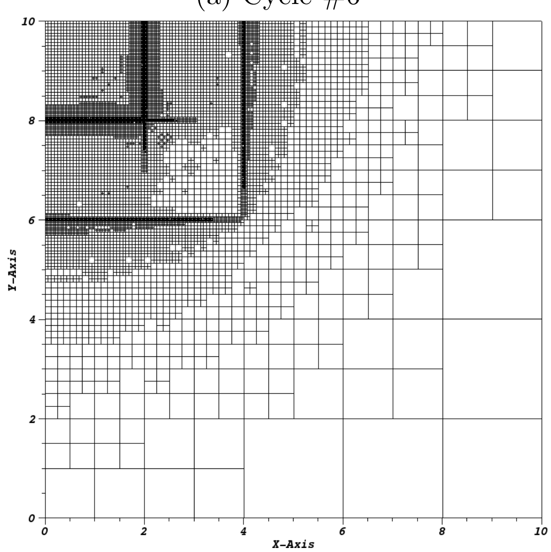
Figure 4.22: Meshes for the Iron-Water problem using the quadratic PWL coordinates and LS4 quadrature.



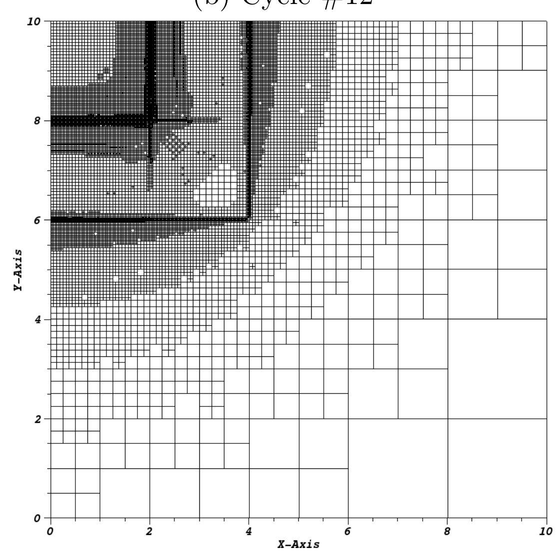
(a) Cycle #6



(b) Cycle #12

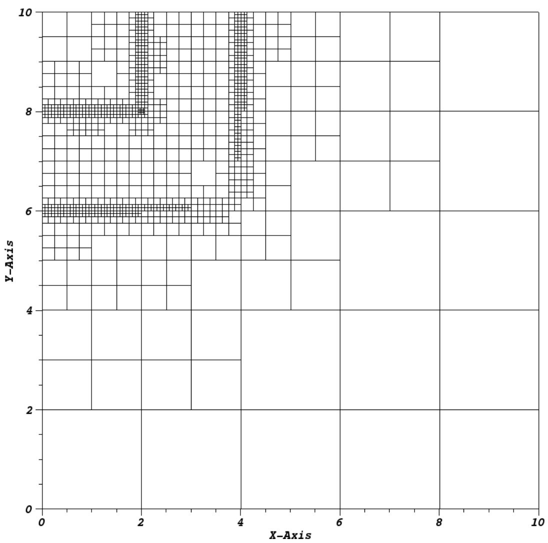


(c) Cycle #18

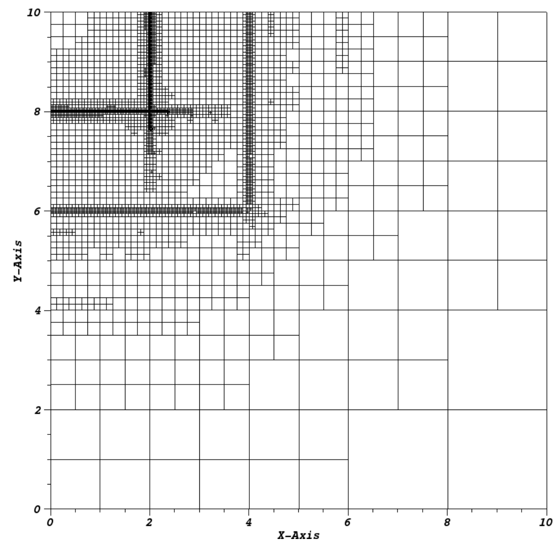


(d) Cycle #24

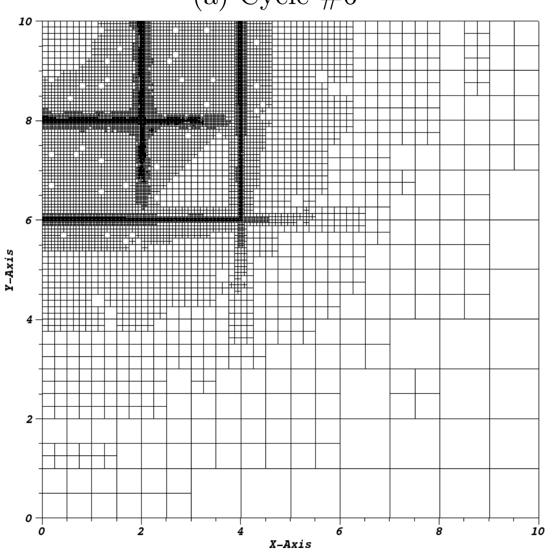
Figure 4.23: Meshes for the Iron-Water problem using the linear PWL coordinates and S_{24}^2 PGLC quadrature.



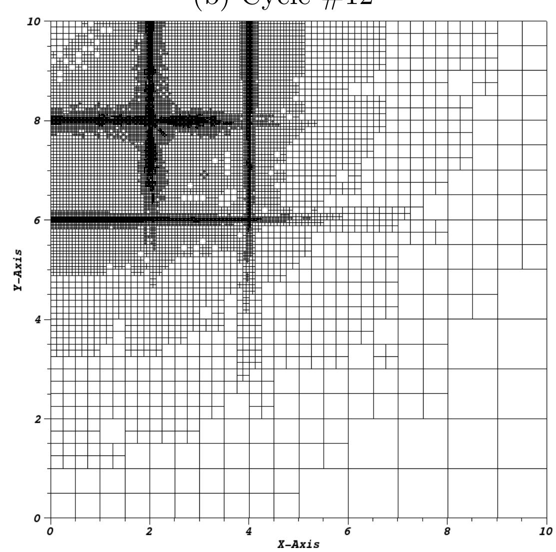
(a) Cycle #6



(b) Cycle #12



(c) Cycle #18



(d) Cycle #24

Figure 4.24: Meshes for the Iron-Water problem using the quadratic PWL coordinates and S_{24}^{2} PGLC quadrature.

high aspect ratios. We also demonstrated that problems with large heterogeneous configurations can diminish the effectiveness of MIP DSA.

Now, we need to demonstrate the scalability of MIP DSA preconditioning onto large massively-parallel computer architectures. Our ability to utilize this transport acceleration form would be greatly minimized if the DSA solve times scaled at a much worse rate compared to the transport sweep times. We specifically use the low-order diffusion operator because it is supposed to be easier to invert than the full transport operator.

From the results of the Iron-Water problem in Section 4.7.3.4, we observed that the PCG iteration counts did not appreciably grow when utilizing AMG as the preconditioner for the diffusion solve. This was in direct contrast to the simpler preconditioners (Jacobi, GS, and ILU) that had their iteration counts grow rapidly as the number of unknowns increased. Motivated by the efficiency of AMG methods with the MIP diffusion form for the simple Iron-Water AMR problem, we will continue to use AMG as the diffusion preconditioner for our massively-parallel calculations.

We have implemented the 1-group and thermal upscattering DSA methodologies of Section 4.2 with the MIP form into Texas A&M University's PDT code. It is a massively-parallel DGFEM S_N transport code that has had good sweep scaling efficiency out to $O(10^5) - O(10^6)$ processes [68, 69]. BoomerAMG of the HYPRE library is used as the AMG diffusion preconditioner [70, 71]. We next analyze the scalability of HYPRE's PCG solver with BoomerAMG preconditioning and how this performs in comparison to PDT's transport sweeping.

4.7.4.1 Weak Scaling with a Homogeneous Zerr Problem

We first test MIP's scaling with HYPRE in PDT by analyzing a simple homogenized version of the Zerr scaling problem [21]. The problem configuration is a 3D cube that spans $[0, 16]^3$ in dimension and uses strictly orthogonal hexahedral mesh cells. The total cross section is set to $\sigma_t = 10$ with a scattering ratio of $c = 0.9999$ and a distributed source of 1 $\frac{n}{cm^3\, s}$. LS8 quadrature is employed and vacuum boundary conditions are used for all boundaries. A residual tolerance of 10^{-8} is used for the richardson transport iterations, and a residual tolerance of 10^{-3} is used for the HYPRE PCG iterations.

Our weak scaling study is performed with a fixed 4096 spatial cells per processor as we increase the number of processors. We do not change the physical dimensions of the problem. Instead, we simply increase the number of cells in a given coordinate direction whenever we allocate more processors in that dimension. The partitioning of the processor layout (P_x, P_y, and P_z), the task aggregation (A_x, A_y, and A_z), and the number of mesh cells per dimension (N_x, N_y, and N_z) are selected to minimize the time per sweep in PDT. These parallel parameters are given for the full scaling suite in Table 4.8.

Table 4.8: Partitioning factors, aggregation factors, and cell counts per dimension for the 3D homogeneous Zerr problem run on Vulcan.

P_{tot}	P_x	P_y	P_z	A_x	A_y	A_z	N_x	N_y	N_z
1	1	1	1	16	16	1	16	16	16
8	2	2	2	16	16	1	32	32	32
64	8	4	2	8	16	1	64	64	64
512	16	16	2	8	8	1	128	128	128
1024	32	16	2	4	8	1	128	128	256
2048	32	32	2	8	4	1	256	128	256
4096	64	32	2	4	8	1	256	256	256
8192	64	64	2	4	4	1	256	256	512
16384	128	64	2	4	4	1	512	256	512
32768	128	128	2	4	4	1	512	512	512
65536	256	128	2	2	4	1	512	512	1024
131072	256	256	2	4	2	1	1024	512	1024

The timing data for this weak scaling study is given in Figure 4.25. As a function of processors used, we plot the overall solve time, the overall sweep time, the overall MIP DSA time, the time to build the MIP system matrix, the time to perform the HYPRE BoomerAMG setup call, and the time per HYPRE PCG iteration. The sweep time only constitutes the time taken to perform the actual sweep. This discounts the time needed to build the transport sources and prepare for each sweep. There are a lot of results that can be observed from this plot. We first note that all the solve times increase at the low processor counts because the number of sweeps increases from about 20 to 35 as the mesh optical thicknesses move into the intermediate range. The second note is that while the sweep times remain about the same, the various HYPRE times begin to scale more poorly as we get to $O(10^4)$ processors and higher. We can see that the HYPRE setup time especially begins to increase at a rapid pace for the highest core counts. Finally we observe that at certain processor values, the HYPRE times seem to 'spike'. This occurs at 1024, 8192, and 65536 processor counts. By looking at the problem's parallel characteristics from Table 4.8, we can see that these processor counts occur when the number of mesh cells in the z-dimension is doubled. This means that the distribution of the cell sets on a processor look like an even longer and skinnier column. This behavior is analyzed in greater detail shortly in Section 4.7.4.2.

From the timing results of Figure 4.25, one must wonder about the scalability of solving the MIP equation at even higher processor counts out to $O(10^6)$ or $O(10^7)$. For this simple problem, our diffusion solver was taking about 50%-60% of the solve time. However, our transport problem was extremely coarse in angle and only 1 energy group. This leads to low parallel concurrency for the phase-space that we solve for in one transport sweep. This means that for a much larger 3D transport problem with many energy groups ($O(10^2)$) and many quadrature angles ($O(10^3)$),

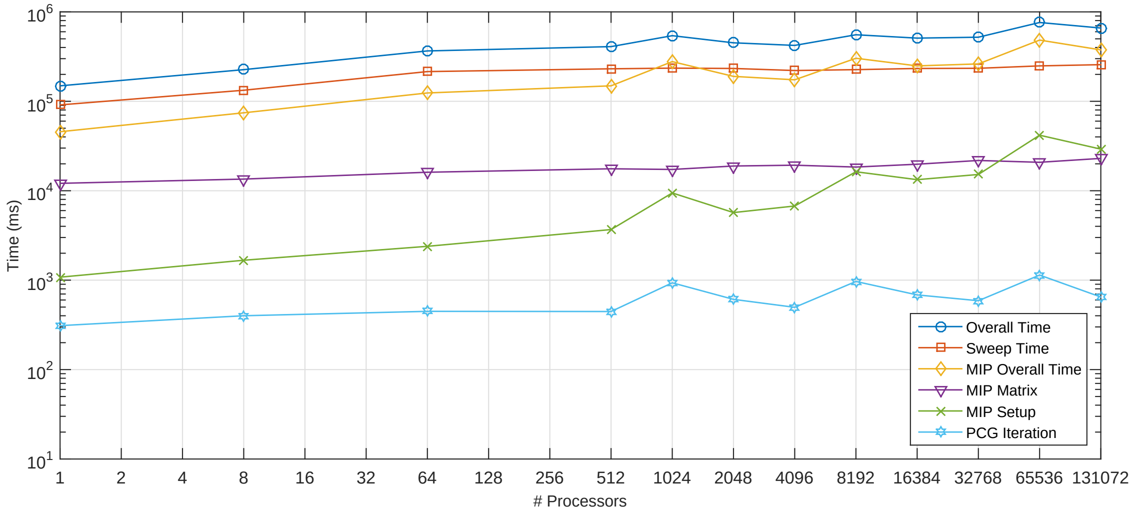


Figure 4.25: blah

the fraction of time that is spent performing DSA calculations will become negligibly small. In fact, we will show a little later that our energy-collapsed thermal neutron upscattering methodologies yield DSA solve times that are not noticible.

4.7.4.2 Aggregation and Partitioning Effects on the HYPRE PCG Algorithm

4.7.5 Thermal Neutron Upscattering Acceleration

We conclude the results of this chapter by presenting analysis on the ability to use DSA to accelerate the convergence of multigroup transport problems that are dominated by thermal neutron upscattering.

4.8 Conclusions

In this chapter, we analyzed the Modified Interior Penalty form of the diffusion equation for use as the diffusion solver for DSA preconditioning of the DGFEM S_N transport equation.

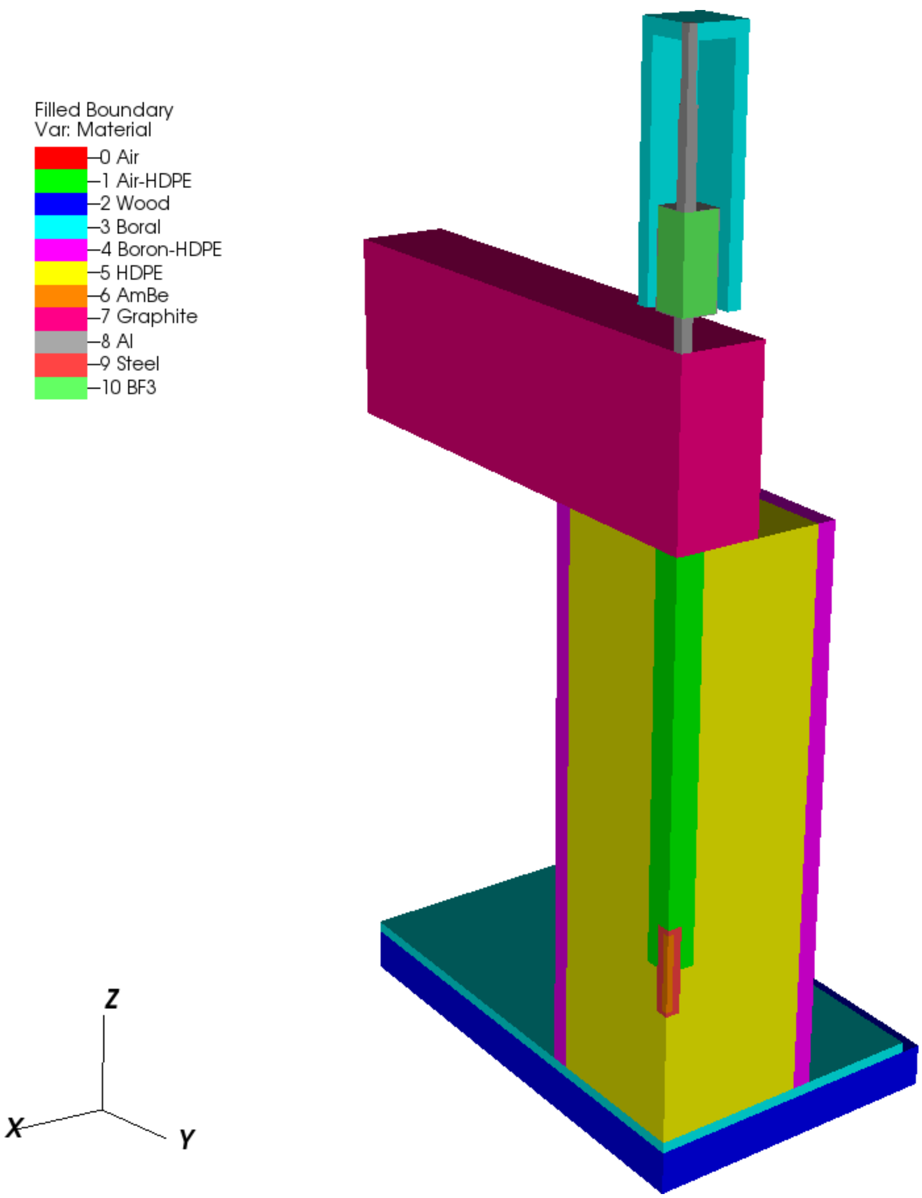
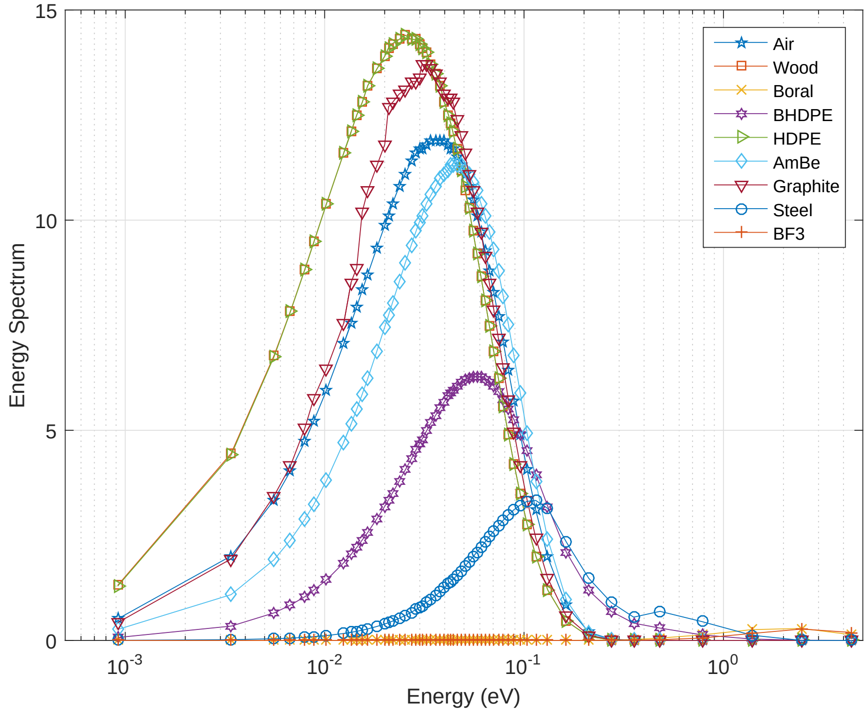
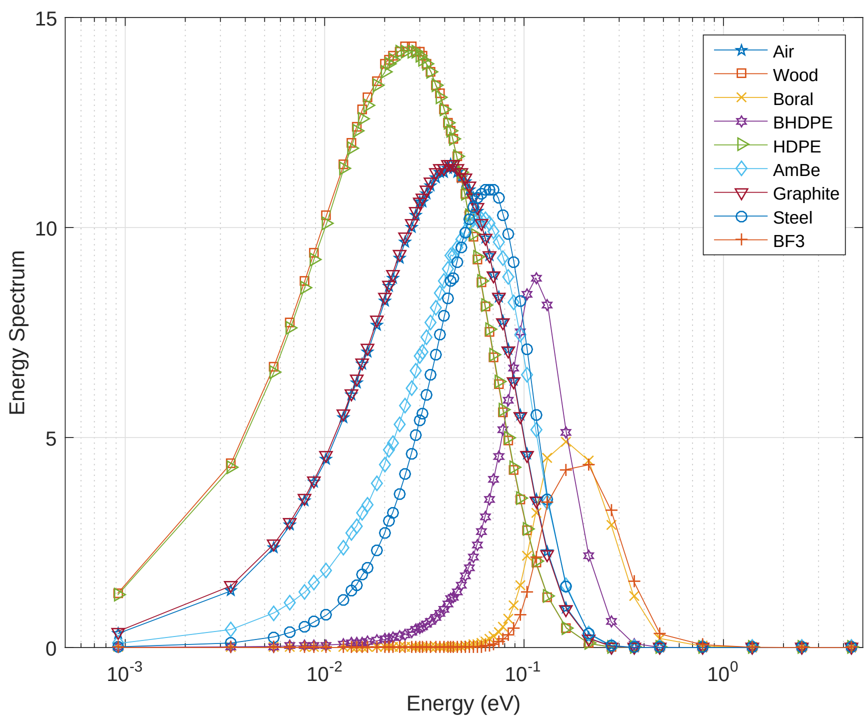


Figure 4.26: Configuration of the IM1 problem with the outer layers of air removed.

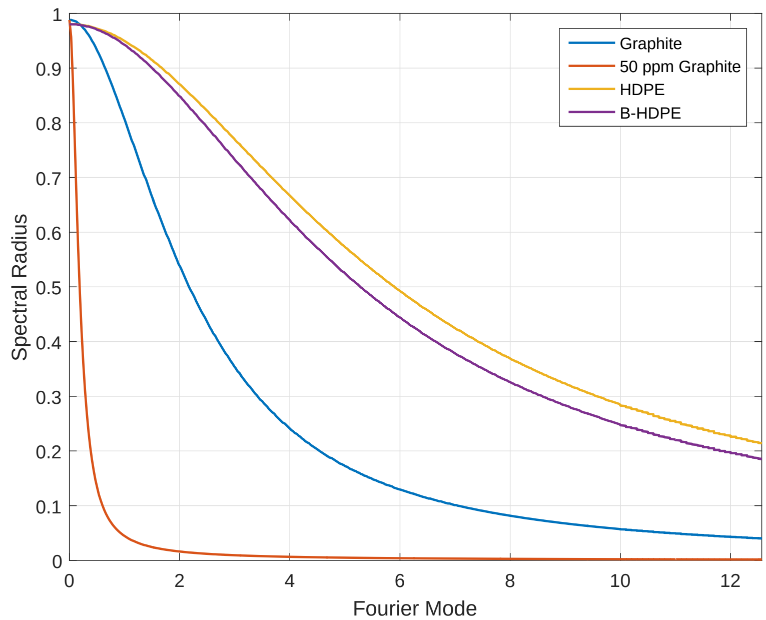


(a) Richardson

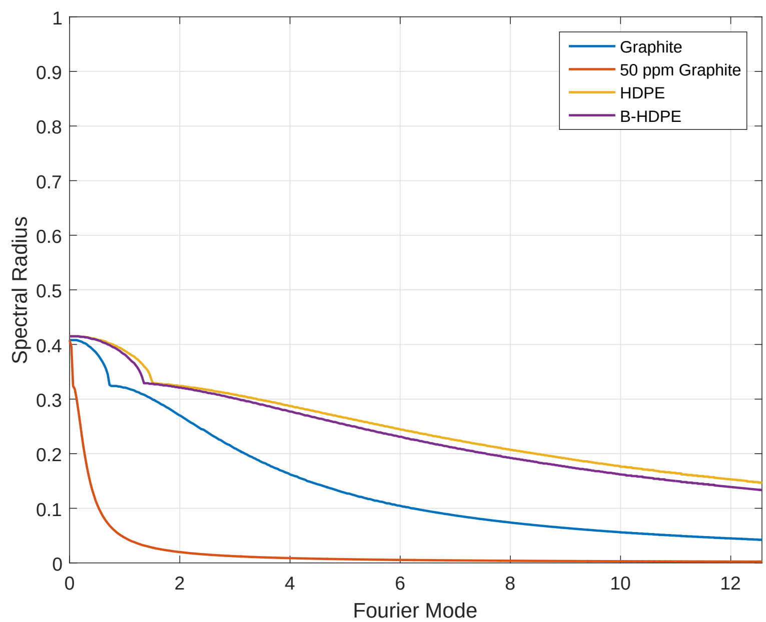


(b)

Figure 4.27: Spectral shape of the infinite medium iteration matrices for the IM1 problem materials.



(a) Gauss-Seidel



(b)

Figure 4.28: 1D fourier analysis for some IM1 materials of interest. (a)

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

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

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

[17] 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).

[18] 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).

[19] 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).

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

[21] 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).

[22] M. S. FLOATER, "Mean value coordinates," *Computer aided geometric design*, **20**, *1*, 19–27 (2003).

[23] K. HORMANN and M. S. FLOATER, "Mean value coordinates for arbitrary planar polygons," *ACM Transactions on Graphics (TOG)*, **25**, *4*, 1424–1441 (2006).

[24] N. SUKUMAR, "Construction of polygonal interpolants: a maximum entropy approach," *International Journal for Numerical Methods in Engineering*, **61**, *12*, 2159–2181 (2004).

[25] M. ARROYO and M. ORTIZ, "Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods," *International journal for numerical methods in engineering*, **65**, *13*, 2167–2202 (2006).

[26] K. HORMANN and N. SUKUMAR, "Maximum entropy coordinates for arbitrary polytopes," in "Computer Graphics Forum," Wiley Online Library (2008), vol. 27, pp. 1513–1520.

[27] S. KULLBACK and R. A. LEIBLER, "On information and sufficiency," *The annals of mathematical statistics*, pp. 79–86 (1951).

[28] E. T. JAYNES, "Information theory and statistical mechanics," in "Statistical Physics," (1963), vol. 3, pp. 181–218.

[29] T. S. BAILEY, *The piecewise linear discontinuous finite element method applied to the RZ and XYZ transport equations*, Ph.D. thesis, Texas A&M University (2008).

[30] A. SHESTAKOV, J. HARTE, and D. KERSHAW, "Solution of the diffusion equation by finite elements in lagrangian hydrodynamic codes," *Journal of Com-*

putational Physics, **76**, *2*, 385–413 (1988).

[31] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, "Test problems in radiative transfer calculations," *Nuclear science and engineering*, **105**, *1*, 88–104 (1990).

[32] D. S. KERSHAW, "Differencing of the diffusion equation in Lagrangian hydrodynamic codes," *Journal of Computational Physics*, **39**, *2*, 375–395 (1981).

[33] J. H. BRAMBLE and S. HILBERT, "Estimation of linear functionals on Sobolev spaces with application to Fourier transforms and spline interpolation," *SIAM Journal on Numerical Analysis*, **7**, *1*, 112–124 (1970).

[34] 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).

[35] 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).

[36] 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).

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

[38] H. KOPP, "Synthetic method solution of the transport equation," *Nuclear Science and Engineering*, **17**, 65 (1963).

[39] V. LEBEDEV, "The Iterative *KP* Method for the Kinetic Equation," in "Proc. Conf. on Mathematical Methods for Solution of Nuclear Physics Problems," (1964).

[40] V. LEBEDEV, "The KP-method of accelerating the convergence of iterations in the solution of the kinetic equation," *Numerical methods of solving problems of mathematical physics, Nauka, Moscow*, pp. 154–176 (1966).

[41] V. LEBEDEV, "On Finding Solutions of Kinetic Problems," *USSR Comp. Math. and Math. Phys.*, **6**, 895 (1966).

[42] V. LEBEDEV, "An Iterative *KP* Method," *USSR Comp. Math. and Math. Phys.*, **7**, 1250 (1967).

[43] V. LEBEDEV, "Problem of the Convergence of a Method of Estimating Iteration Deviations," *USSR Comp. Math. and Math. Phys.*, **8**, 1377 (1968).

[44] V. LEBEDEV, "Convergence of the *KP* Method for Some Neutron Transfer Problems," *USSR Comp. Math. and Math. Phys.*, **9**, 226 (1969).

[45] V. LEBEDEV, "Construction of the *P* Operation in the *KP* Method," *USSR Comp. Math. and Math. Phys.*, **9**, 762 (1969).

[46] E. GELBARD and L. HAGEMAN, "The synthetic method as applied to the SN equations," *Nucl. Sci. Eng*, **37**, *2*, 288 (1969).

[47] W. H. REED, "The effectiveness of acceleration techniques for iterative methods in transport theory," *Nucl. Sci. Eng*, **45**, *3*, 245 (1971).

[48] R. ALCOUFFE, "Stable diffusion synthetic acceleration method for neutron transport iterations," Los Alamos Scientific Lab., NM (1976), vol. 23.

[49] R. ALCOUFFE, "The Diffusion Synthetic Acceleration Method Applied to Two-Dimenaionsl Neutron Transport Problems," Los Alamos Scientific Lab., NM (1977), vol. 27.

[50] R. E. ALCOUFFE, "Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations," *Nuclear Science and Engineering*, **64**, *2*, 344–355 (1977).

[51] E. W. LARSEN, "Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part I: Theory," *Nucl. Sci. Eng*, **82**, 47 (1982).

[52] D. R. MCCOY and E. W. LARSEN, "Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part II: Numerical results," *Nucl. Sci. Eng*, **82**, 64 (1982).

[53] M. L. ADAMS and W. R. MARTIN, "Diffusion synthetic acceleration of discontinuous finite element transport iterations," *Nucl. Sci. Eng*, **111**, 145–167 (1992).

[54] J. S. WARSA, T. A. WAREING, and J. E. MOREL, "Fully consistent diffusion synthetic acceleration of linear discontinuous S n transport discretizations on unstructured tetrahedral meshes," *Nuclear Science and Engineering*, **141**, *3*, 236–251 (2002).

[55] E. L. T.A. WAREING and M. ADAMS, "Diffusion Accelerated Discontinuous Finite Element Schemes for the S_N Equations in Slab and X-Y Geometries," in "Advances in Mathematics, Computations, and Reactor Physics," (1991), p. 245.

[56] Y. WANG and J. RAGUSA, “Diffusion Synthetic Acceleration for High-Order Discontinuous Finite Element S_n Transport Schemes and Application to Locally Refined Unstructured Meshes,” *Nuclear Science and Engineering*, **166**, 145–166 (2010).

[57] Y. WANG, *Adaptive mesh refinement solution techniques for the multigroup SN transport equation using a higher-order discontinuous finite element method*, Ph.D. thesis, Texas A&M University (2009).

[58] B. TURCKSIN and J. C. RAGUSA, “Discontinuous diffusion synthetic acceleration for Sn transport on 2D arbitrary polygonal meshes,” *Journal of Computational Physics*, **274**, 356–369 (2014).

[59] B. ADAMS and J. MOREL, “A two-grid acceleration scheme for the multigroup Sn equations with neutron upscattering,” *Nuclear science and engineering*, **115**, *3*, 253–264 (1993).

[60] D. N. ARNOLD, F. BREZZI, B. COCKBURN, and L. D. MARINI, “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM journal on numerical analysis*, **39**, *5*, 1749–1779 (2002).

[61] J. C. RAGUSA, “Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids,” *Journal of Computational Physics*, **280**, 195–213 (2015).

[62] M. HACKEMACK and J. RAGUSA, “A DFEM Formulation of the Diffusion Equation on Arbitrary Polyhedral Grids,” in “Transactions of the American Nuclear Society,” (2014).

[63] S. C. EISENSTAT, “Efficient implementation of a class of preconditioned conjugate gradient methods,” *SIAM Journal on Scientific and Statistical Computing*,

2, *1*, 1–4 (1981).

[64] J. A. NELDER and R. MEAD, “A simplex method for function minimization,” *The computer journal*, **7**, *4*, 308–313 (1965).

[65] J. CHANG and M. ADAMS, “Analysis of transport synthetic acceleration for highly heterogeneous problems,” in “Proc. of M&C Topical Meeting,” (2003).

[66] Y. AZMY, “Unconditionally stable and robust adjacent-cell diffusive preconditioning of weighted-difference particle transport methods is impossible,” *Journal of Computational Physics*, **182**, *1*, 213–233 (2002).

[67] H. KHALIL, “A nodal diffusion technique for synthetic acceleration of nodal S_n calculations,” *Nuclear Science and Engineering*, **90**, *3*, 263–280 (1985).

[68] M. A. L. R. N. A. WILLIAM HAWKINS, TIMMIE SMITH and M. ADAMS, “Efficient Massively Parallel Transport Sweeps,” in “Transactions of the American Nuclear Society,” (2012), vol. 107, pp. 477–481.

[69] M. P. ADAMS, M. L. ADAMS, W. D. HAWKINS, T. SMITH, L. RAUCHWERGER, N. M. AMATO, T. S. BAILEY, and R. D. FALGOUT, “Provably Optimal Parallel Transport Sweeps on Regular Grids,” in “Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, Idaho,” (2013).

[70] R. D. FALGOUT and U. M. YANG, “hypre: A Library of High Performance Preconditioners,” in P. SLOOT, A. HOEKSTRA, C. TAN, and J. DONGARRA, editors, “Computational Science ICCS 2002,” Springer Berlin Heidelberg, *Lecture Notes in Computer Science*, vol. 2331, pp. 632–641 (2002).

[71] V. HENSON and U. YANG, “BoomerAMG: a parallel algebraic multigrid solver and preconditioner,” *Applied Numerical Mathematics*, **41**, *5*, 155–177 (2002).