

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

May 2016

Major Subject: Nuclear Engineering

Copyright 2016 Michael Wayne Hackemack

# TABLE OF CONTENTS

	Page
TABLE OF CONTENTS . . . . .	1
LIST OF FIGURES . . . . .	3
LIST OF TABLES . . . . .	5
1. INTRODUCTION . . . . .	1
1.1 Current State of the Problem . . . . .	1
1.2 Motivation and Purpose of the Dissertation . . . . .	1
1.3 Organization of the Dissertation . . . . .	1
2. THE DGFEM FORMULATION OF THE MULTIGROUP $S_N$ EQUATIONS	3
2.1 The Neutron Transport Equation . . . . .	4
2.2 Energy Discretization . . . . .	7
2.3 Angular Discretization . . . . .	10
2.3.1 Level Symmetric Quadrature Set . . . . .	17
2.3.2 Product Gauss-Legendre-Chebyshev Quadrature Set . . . . .	22
2.4 Boundary Conditions . . . . .	24
2.5 Spatial Discretization . . . . .	28
2.5.1 Convergence Rates of the DGFEM $S_N$ Equation . . . . .	32
2.5.2 Elementary Matrices on an Arbitrary Spatial Cell . . . . .	33
2.5.2.1 Elementary Mass Matrices . . . . .	33
2.5.2.2 Elementary Streaming Matrices . . . . .	34
2.5.2.3 Elementary Surface Matrices . . . . .	35
2.6 Solution Procedures . . . . .	37
2.6.1 Angle and Energy Iteration Procedures . . . . .	38
2.6.1.1 Source Iteration . . . . .	41
2.6.1.2 Krylov Subspace Methods - GMRES . . . . .	41
2.6.2 Spatial Solution Procedures . . . . .	41
2.6.2.1 Transport Sweeping . . . . .	41
2.6.2.2 Polytope Grids Formed from Voronoi Mesh Generation	43
2.6.2.3 Spatial Adaptive Mesh Refinement . . . . .	43
2.7 Conclusions . . . . .	45
3. FEM BASIS FUNCTIONS FOR UNSTRUCTURED POLYTOPES . . . . .	46

3.1	Linear Basis Functions on 2D Polygons . . . . .	46
3.1.1	Traditional Linear Basis Functions - $\mathbb{P}_1$ and $\mathbb{Q}_1$ Spaces . . . .	48
3.1.2	Wachspress Rational Basis Functions . . . . .	49
3.1.3	Piecewise Linear (PWL) Basis Functions . . . . .	49
3.1.4	Mean Value Basis Functions . . . . .	50
3.1.5	Maximum Entropy Basis Functions . . . . .	51
3.1.6	Summary of 2D Linear Basis Functions on Polygons . . . . .	54
3.2	Quadratic Serendipity Basis Functions on 2D Polygons . . . . .	54
3.2.1	Traditional Quadratic Basis Functions - $\mathbb{P}_2$ and $\mathbb{S}_2$ Spaces . . .	56
3.2.2	Analytical form of the quadratic serendipity extension of the PWL functions . . . . .	56
3.3	Linear Basis Functions on 3D Polyhedra . . . . .	56
3.3.1	3D Linear and TriLinear Basis Functions . . . . .	56
3.3.2	3D Piecewise Linear (PWL) Basis Functions . . . . .	56
3.4	Numerical Results . . . . .	57
3.4.1	Two-Dimensional Exactly-Linear Transport Solutions . . . . .	58
3.4.2	Convergence Rate Analysis by the Method of Manufactured Solutions . . . . .	66
3.4.3	Searchlight Problem . . . . .	68
3.5	Conclusions . . . . .	69
	REFERENCES . . . . .	71

## LIST OF FIGURES

FIGURE		Page
2.1	Interval structure of the multigroup methodology. . . . .	8
2.2	Number of spherical harmonics moments . . . . .	13
2.3	Angular Coordinate System . . . . .	18
2.4	3D Level-Symmetric angular quadrature set . . . . .	20
2.5	2D Level-Symmetric angular quadrature set . . . . .	21
2.6	3D Product Gauss-Legendre-Chebyshev angular quadrature set . . .	25
2.7	2D Product Gauss-Legendre-Chebyshev angular quadrature set . . .	26
2.8	Two cells of the spatial discretization . . . . .	29
2.9	Scattering matrices with and without upscattering . . . . .	40
2.10	blah . . . . .	42
2.11	blah . . . . .	43
3.1	Arbitrary polygon with geometric properties used for 2D basis function generation. . . . .	47
3.2	Contour plots of the PWL basis functions on the unit square for the vertices located at: (a) (0,0), (b) (1,0), (c) (1,1), and (d) (0,1). . . . .	50
3.3	Vertex structure for a (a) regular pentagonal cell and a (b) degenerate pentagonal cell. . . . .	51
3.4	Contour plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d). . . . .	52
3.5	Plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d). . . . .	53
3.6	Vertex structure for the (a) unit square and (b) unit cube. . . . .	55

3.7	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. . . . .	61
3.8	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. . . . .	62
3.9	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. . . . .	63
3.10	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. . . . .	64
3.11	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. . . . .	65
3.12	Initial mesh configuration for the searchlight problem before any refinement cycles. . . . .	69
3.13	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. . . . .	70

## LIST OF TABLES

TABLE		Page
2.1	2D angle mapping from the first quadrant into the other 3 quadrants.	16
2.2	3D angle mapping from the first octant into the other 7 octants. . . .	17
3.1	Summary of the 2D polygonal basis functions . . . . .	55

# 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 introduction 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

In Chapter ??, we present

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 ??, 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:

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

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:

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

with the following, general boundary condition:

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

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 neutrons emitted per fission,  $\sigma_s(E' \rightarrow E, \Omega' \rightarrow \Omega, t)$  is the scattering cross section, and  $Q_{ext}(\vec{r}, E, \vec{\Omega}, t)$  is a distributed external source.

We can simplify Eq. (2.2) to:

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

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

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

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

$$\mathbf{L}\Psi = \mathbf{F}\Psi + \mathbf{S}\Psi + \mathbf{Q}, \quad (2.6)$$

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:

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

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

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

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:  $E_G < E_{G-1} < \dots < E_1 < E_0$ .

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

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

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

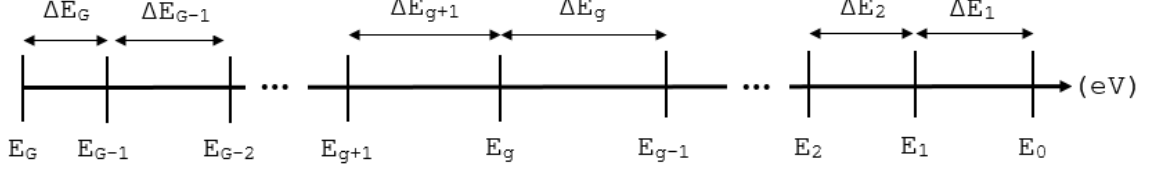


Figure 2.1: Interval structure of the multigroup methodology.

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

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

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

where

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

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 distribution 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  $s$ :

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

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):

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

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 steady-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:

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

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,  $\mu_0$ ,

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



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,

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

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:

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

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

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

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

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

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:

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

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,

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

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

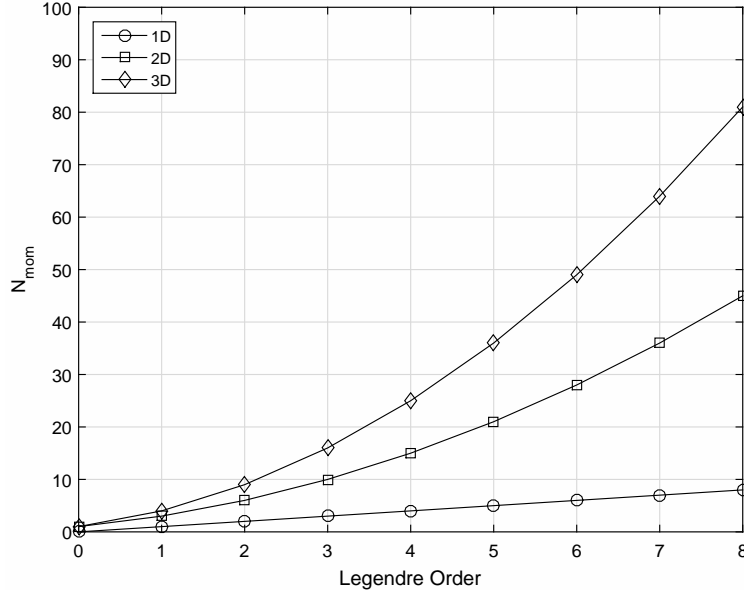


Figure 2.2: Number of spherical harmonics moments,  $N_{mom}$ , in 1D, 2D, and 3D as a function of the expansion order,  $p$ .

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,

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

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

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

as well as the *Lagrange* property,

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

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

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

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

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

$k = 1, \dots, M$

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

angular flux moments:

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

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  $m$ ):

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

$m = 1, \dots, M$

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  $\vec{0}$ :  $\sum_m w_m \left( \vec{\Omega}_m \right)^n = \vec{0}$  ( $n = 1, 3, 5, \dots$ ).
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	$\mu$	$\eta$
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	$\mu$	$\eta$	$\xi$
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

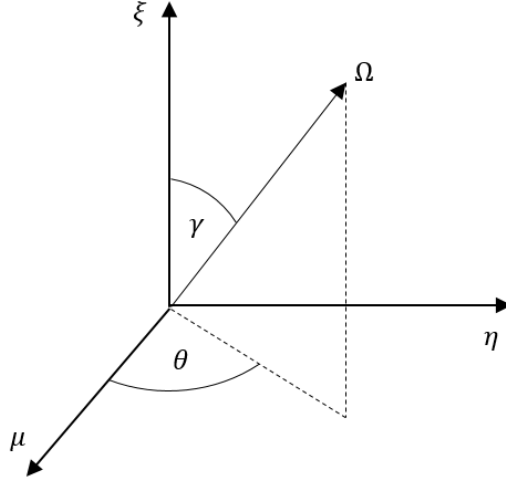


Figure 2.3: Angular coordinate system for the direction  $\vec{\Omega}$ .

angular direction,  $\vec{\Omega} = [\vec{\Omega}_x, \vec{\Omega}_y, \vec{\Omega}_z]$ , is typically described with its directional cosines:  $\mu$ ,  $\eta$ , and  $\xi$ . These are described by the angles  $\theta$  and  $\gamma$  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 ( $\mu$ ,  $\eta$ , and  $\xi$ ) 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:

$$\mu_i = \eta_i = \xi_i, \quad i \in (1, N/2) \quad (2.32)$$

As previously stated, once the location of the first ordinate,  $\mu_1$ , 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  $\mu_1$  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):

$$\mu_i^2 = \mu_{i-1}^2 + \Delta \quad (2.33)$$

where the spacing constant,  $\Delta$ , has the form:

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

Based on this recursion form, we can see that if  $\mu_1^2$  is close to 0, then the ordinates will be clustered around the poles of the primary octant. Likewise, if  $\mu_1^2$  is close to 1/3, then the ordinates will be clustered away from the poles. For this work, we choose to select values of  $\mu_1$  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  $N \geq 20$ .

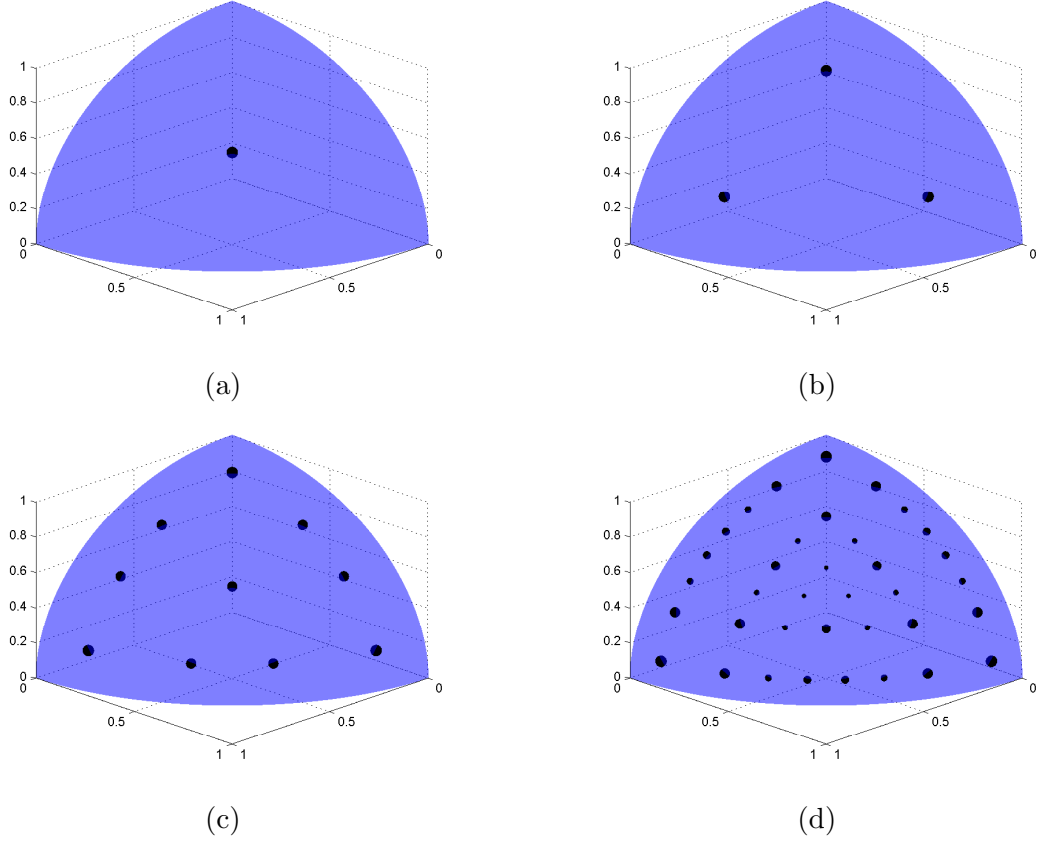


Figure 2.4: Level-Symmetric angular quadrature sets of order (a) 2, (b) 4, (c) 8, and (d) 16.

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 are 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.

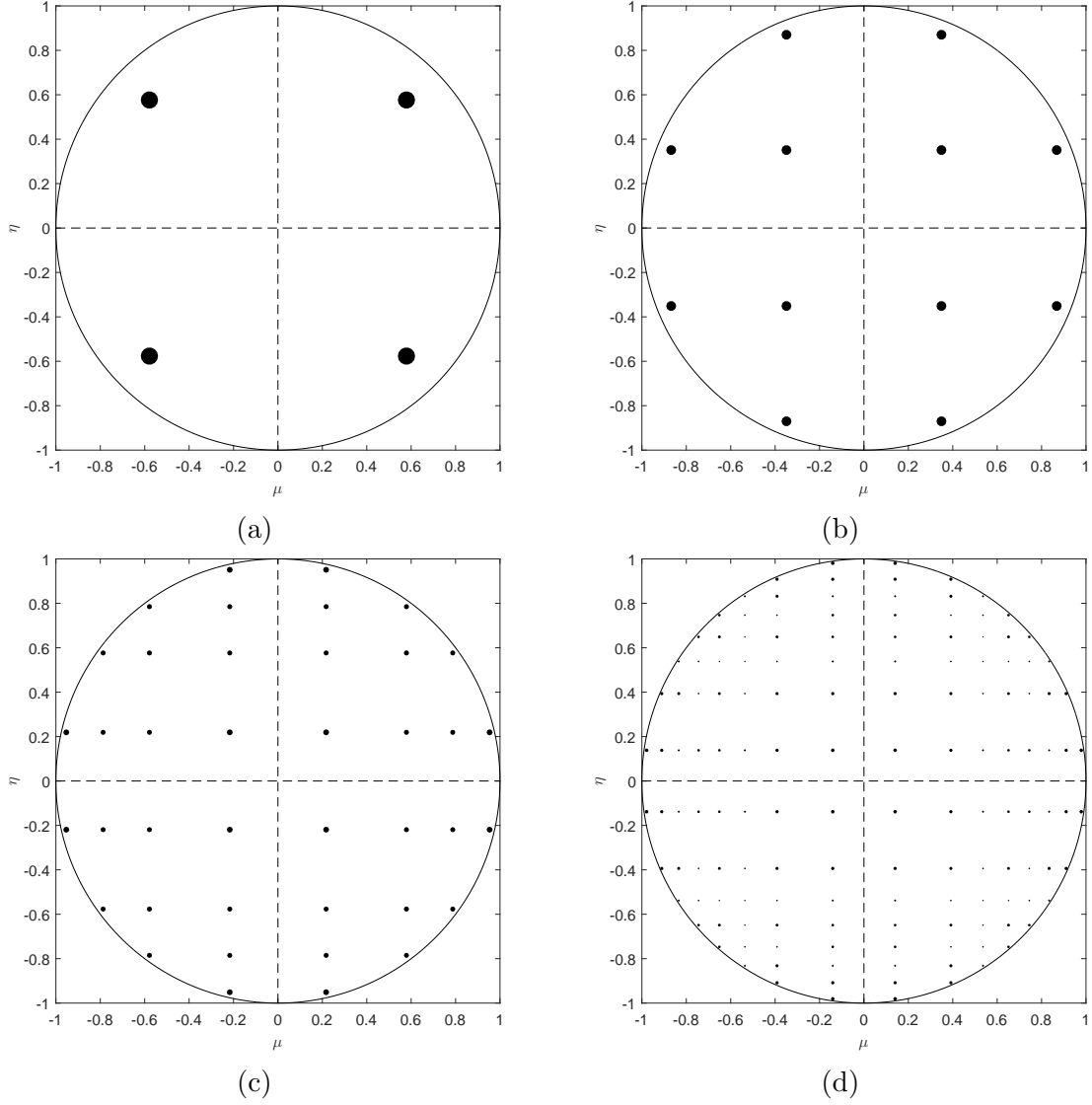


Figure 2.5: Projection of the 3D Level-Symmetric angular quadrature set with orders (a) 2, (b) 4, (c) 8, and (d) 16 onto the x-y space on the unit circle.

### 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  $90^\circ$  rotational invariance within the primary octant; we still maintain octant-to-octant symmetry via mapping however;
- Has more control over the placement of the angular 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,  $\theta$ , 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

$$\theta_m = \frac{2m-1}{4A}\pi \quad \text{and} \quad w_m = \frac{\pi}{2A}, \quad (2.35)$$

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,  $\xi$ , of our quadrature set are the positive roots of the  $2P$ -order Legendre polynomials taken over the interval  $[-1, 1]$ . In this case, we simply discard the negative roots. The corresponding Legendre weights are given by the following formula,

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

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,

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

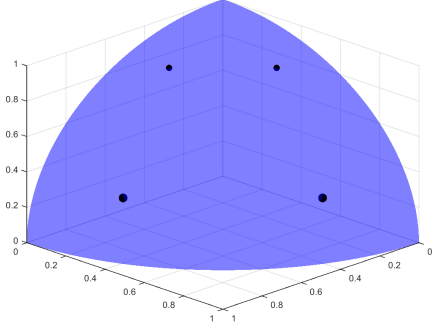
For this dissertation, we will use the following notation to define the product nature of the PGLC quadrature points:  $S_A^P$ . Here,  $A$  and  $P$  correspond to the number of azimuthal and polar directions in the primary octant, respectively. We demonstrate this definition in Figure 2.6 for the primary 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

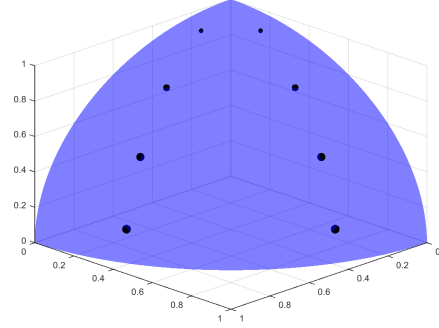
Using the energy and angular discretizations presented in Sections 2.2 and 2.3, respectively, we write the standard, steady-state, multigroup  $S_N$  transport equation for one angular direction,  $m$ , and one energy group,  $g$ :

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

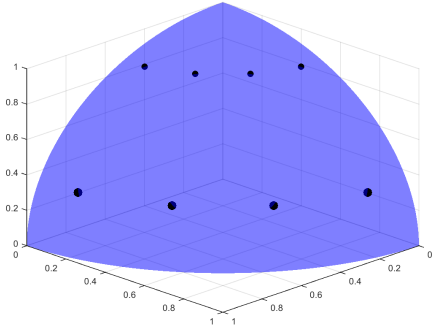
where we have dropped the spatial parameter for clarity and is beholden to the



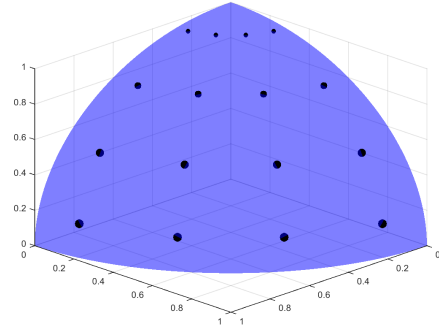
(a)



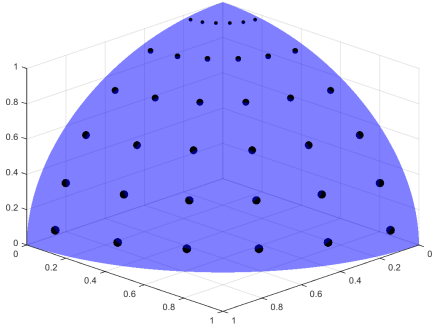
(b)



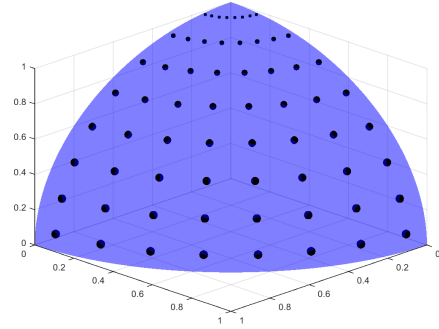
(c)



(d)



(e)



(f)

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)  $S_6^2$ , and (f)  $S_8^2$ .

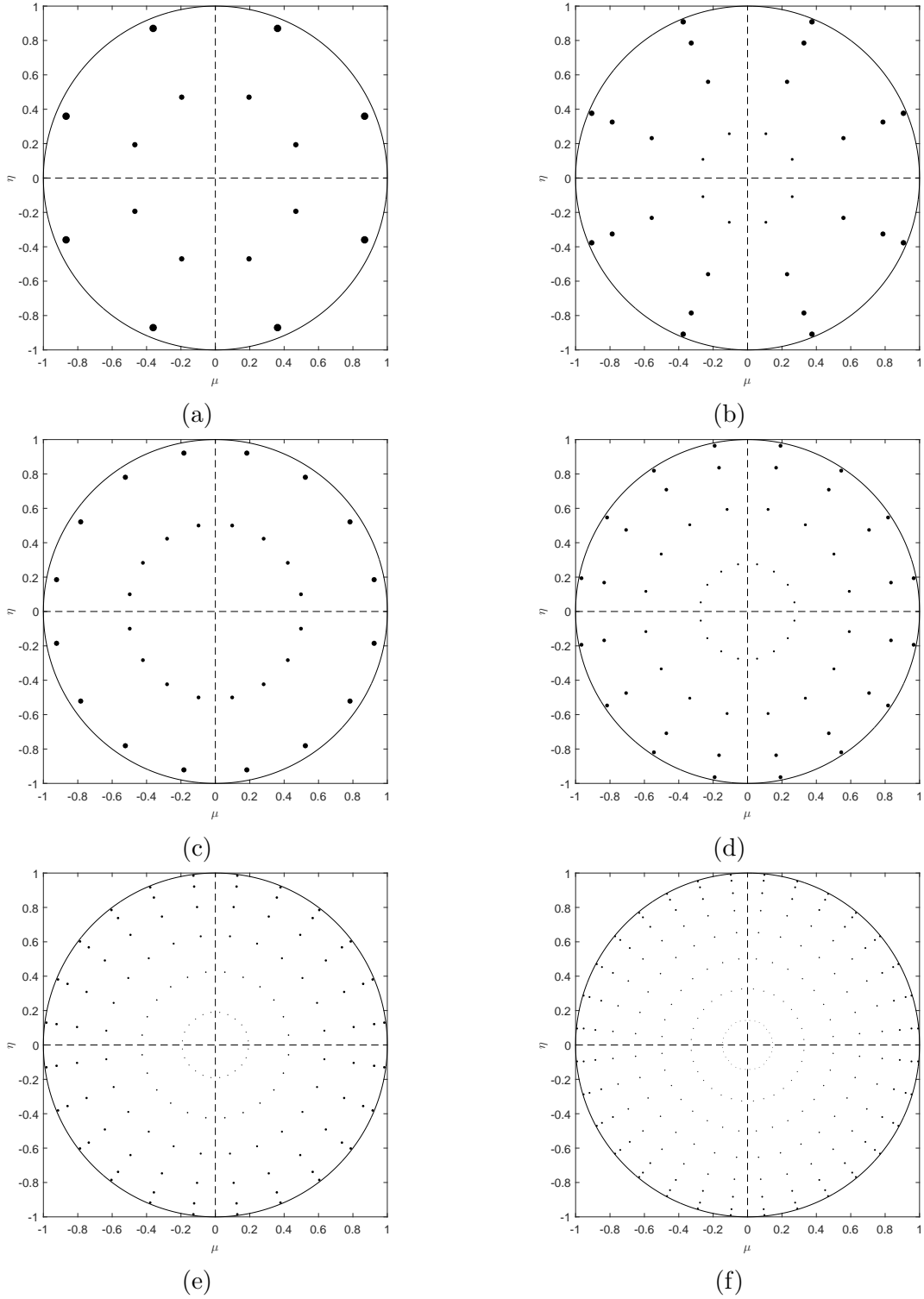


Figure 2.7: Projection of the 3D Product Gauss-Legendre-Chebyshev angular quadrature set with orders: (a)  $S_2^2$ , (b)  $S_2^2$ , (c)  $S_4^2$ , (d)  $S_4^2$ , (e)  $S_6^2$ , and (f)  $S_8^2$  onto the x-y space on the unit circle.



following general, discretized boundary condition:

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

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

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

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

where the reflecting angle is  $\vec{\Omega}_{m'} = \vec{\Omega}_m - 2(\vec{\Omega}_m \cdot \vec{n})\vec{n}$  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)

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

to form  $M$  ( $m = 1, \dots, M$ ) 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 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.*  $\vec{n}' = -\vec{n}$ ).

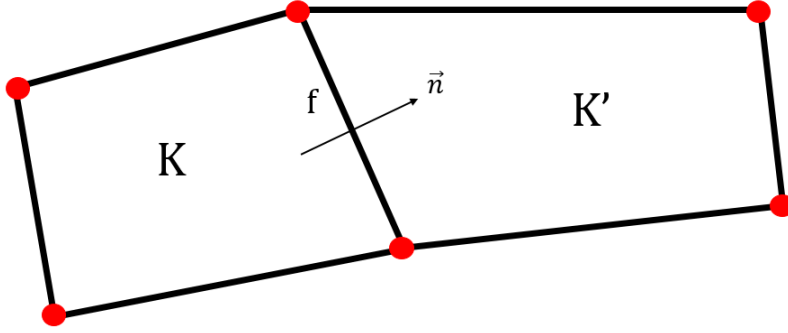


Figure 2.8: Two cells of the spatial discretization with the connecting face,  $f$ , with normal direction,  $\vec{n}$ , oriented from cell  $K$  to cell  $K'$ .

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  $\vec{\Omega}_m$ :

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

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:  $\partial K^- \setminus \partial \mathcal{D}$ ; 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. Therefore, with the cell boundary terminology as proposed, Eq. (2.42) can be written into the following form:

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

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,

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

when the following trace is applied to the angular fluxes :

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

This trace has the notation, with  $\vec{n}$  pointing outwards from cell  $K$ , of  $\Psi_m^-$  corresponding to fluxes within the cell and  $\Psi_m^+$  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

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

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 . We do this by taking Eq. (2.48), performing another integration-by-parts on the streaming term, multiplying by the angular weight,  $w_m$ , and summing over all elements and all angular directions. This leads to the variational form for the 1-group  $S_N$  equation:

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

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

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

and

$$\left\langle u, v \right\rangle_{E_h^i} \equiv \sum_{f \in E_h^i} \left\langle u, v \right\rangle_f, \quad (2.51)$$

respectively. The interior faces are designated as the non-repeating set:  $E_h^i \in \bigcup_{K \in \mathbb{T}_h} \partial K \setminus \partial\mathcal{D}$ .

### 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.

#### 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

$$\mathbf{M}_K = \int_K \mathbf{b}_K \mathbf{b}_K^T dr, \quad (2.52)$$

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. (2.52) 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

$$\mathbf{M}_K = \sum_{q=1}^{N_q} w_q^K \mathbf{b}_K(\vec{x}_q) \mathbf{b}_K^T(\vec{x}_q). \quad (2.53)$$

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  $(N_K \times N_K)$  matrix. This matrix is written for completeness of this discussion on the mass matrix:

$$\mathbf{M}_K = \begin{bmatrix} \int_K b_1 b_1 & \dots & \int_K b_1 b_j & \dots & \int_K b_1 b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_i b_1 & \dots & \int_K b_i b_j & \dots & \int_K b_i b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_{N_K} b_1 & \dots & \int_K b_{N_K} b_j & \dots & \int_K b_{N_K} b_{N_K} \end{bmatrix}, \quad (2.54)$$

where an individual matrix entry is of the form:

$$M_{i,j,K} = \int_K b_i b_j. \quad (2.55)$$

#### 2.5.2.2 Elementary Streaming Matrices

Next, we will consider the streaming term that has the form:  $(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m)_K$  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:

$$\vec{\mathbf{G}}_K = \int_K \vec{\nabla} \mathbf{b}_K \mathbf{b}_K^T dr, \quad (2.56)$$

which has dimensionality  $(N_K \times N_K \times d)$ . We choose to store the elementary streaming matrix in this form and not store  $M$  separate  $(N_K \times N_K)$  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,  $\{\vec{x}_q, w_q^K\}_{q=1}^{N_q}$ , for cell  $K$  to numerically calculate the streaming matrix:

$$\vec{\mathbf{G}}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla} \mathbf{b}_K(\vec{x}_q) \mathbf{b}_K^T(\vec{x}_q). \quad (2.57)$$

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

$$\vec{\mathbf{G}}_K = \begin{bmatrix} \int_K \vec{\nabla} b_1 b_1 & \dots & \int_K \vec{\nabla} b_1 b_j & \dots & \int_K \vec{\nabla} b_1 b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_i b_1 & \dots & \int_K \vec{\nabla} b_i b_j & \dots & \int_K \vec{\nabla} b_i b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_{N_K} b_1 & \dots & \int_K \vec{\nabla} b_{N_K} b_j & \dots & \int_K \vec{\nabla} b_{N_K} b_{N_K} \end{bmatrix}, \quad (2.58)$$

where an individual matrix entry is of the form:

$$\vec{G}_{i,j,K} = \int_K \vec{\nabla} b_i b_j. \quad (2.59)$$

### 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:  $\vec{\Omega}_m \cdot \langle \vec{n} b_m, \Psi_m \rangle_{\partial K}$ . 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,

$$\vec{\mathbf{F}}_{f,K} = \int_f \vec{n}(\vec{r}) \mathbf{b}_K \mathbf{b}_K^T ds, \quad (2.60)$$

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. (2.60), we can see that they have dimensionality,  $(N_K \times N_K \times d)$ . 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  $(N_b^{f,K} \times N_b^{f,K})$ .

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  $K$ :

$$\vec{\mathbf{F}}_{f,K} = \sum_{q=1}^{N_q^f} w_q^f \vec{n}(\vec{x}_q) \mathbf{b}_K(\vec{x}_q) \mathbf{b}_K^T(\vec{x}_q). \quad (2.61)$$

Similar to the cell-wise spatial quadrature sets, the sum of the weights of these face-wise quadrature sets need to exactly equal the geometric measure of face  $f$ . This means that  $\sum_{q=1}^{N_q^f}$  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,

$$\vec{\mathbf{F}}_{f,K} = \begin{bmatrix} \int_f \vec{n} b_1 b_1 & \dots & \int_f \vec{n} b_1 b_j & \dots & \int_f \vec{n} b_1 b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n} b_i b_1 & \dots & \int_f \vec{n} b_i b_j & \dots & \int_f \vec{n} b_i b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n} b_{N_K} b_1 & \dots & \int_f \vec{n} b_{N_K} b_j & \dots & \int_f \vec{n} b_{N_K} b_{N_K} \end{bmatrix}, \quad (2.62)$$

where an individual matrix entry is of the form:

$$\vec{F}_{i,j,f,K} = \int_f \vec{n} b_i b_j. \quad (2.63)$$

## 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:

$$\begin{aligned}
& -\left(\vec{\Omega}_m \cdot \vec{\nabla} b_{m,g}, \Psi_{m,g}\right)_K + \left(\sigma_{t,g} b_{m,g}, \Psi_{m,g}\right)_K + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_{m,g}, \Psi_{m,g}^- \right\rangle_{\partial K^+} \\
& + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_{m,g}, \Psi_{m,g}^+ \right\rangle_{\partial K^- \setminus \partial \mathcal{D}} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_{m,g}, \Psi_{m',g}^- \right\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\
& = \sum_{g'=1}^G \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^p Y_{p,n}(\vec{\Omega}_m) \left( \sigma_{s,p}^{g' \rightarrow g} b_{m,g}, \Phi_{p,n,g'} \right)_K \\
& + (b_{m,g}, Q_{m,g})_K + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_{m,g}, \Psi_{m,g}^{inc} \right\rangle_{\partial K^- \cap \partial \mathcal{D}^d}. \tag{2.64}
\end{aligned}$$

All of the notations used in Eq. (2.64) 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,  $\Psi$ , with dimensionality of  $(G \times M \times N_{dof})$ . The angular flux moments,  $\Phi$ , have dimensionality of  $(G \times N_{mom} \times N_{dof})$ . 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  $(G \times M \times N_{dof}) \times (G \times M \times N_{dof})$  with a corresponding right-hand-side vector of dimensionality  $(G \times M \times N_{dof}) \times 1$ , 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.

$$\begin{aligned}\mathbf{L}\Psi - \mathbf{M}\Sigma\Phi &= \mathbf{Q} \\ \Phi &= \mathbf{D}\Psi\end{aligned}\tag{2.65}$$

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,  $\Sigma$  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.

$$(\mathbf{I} - \mathbf{T})\Phi = \mathbf{D}\mathbf{L}^{-1}\mathbf{Q}\tag{2.66}$$

where we define,

$$\mathbf{T} \equiv \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\Sigma,\tag{2.67}$$

for further brevity.

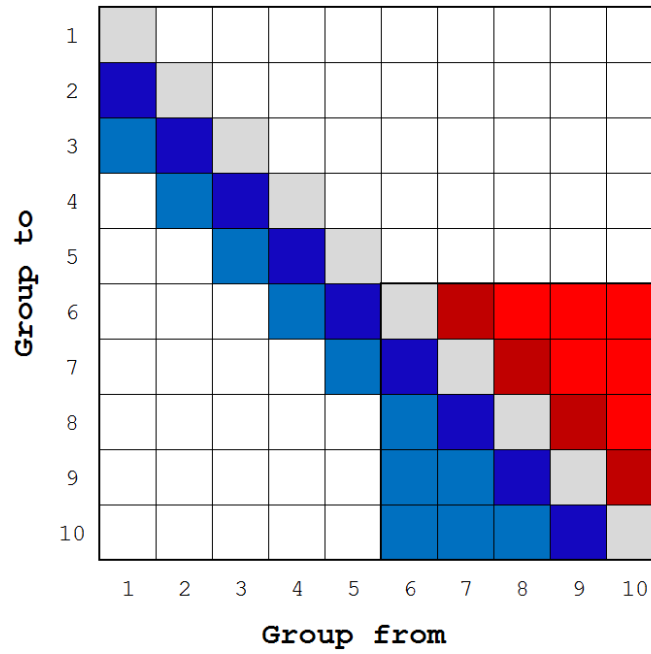
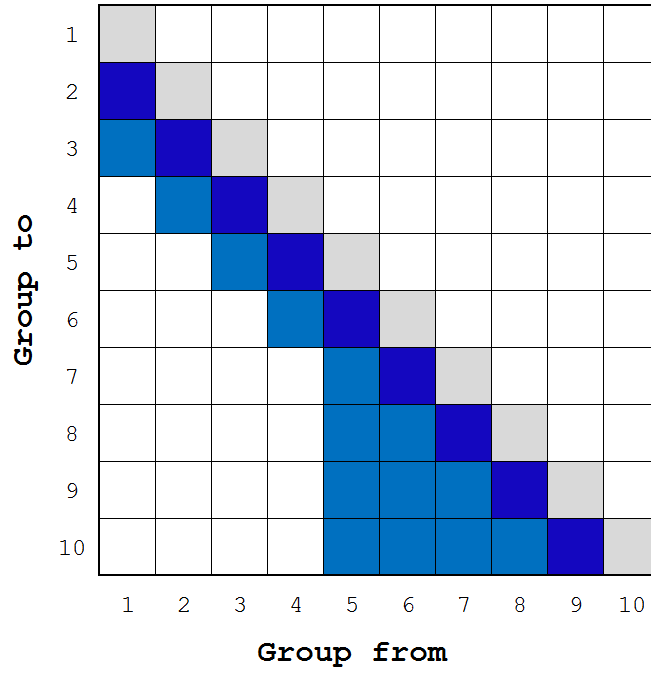


Figure 2.9: Scattering matrices (top) without and (bottom) with upscattering. The gray corresponds to within-group scattering; the blue corresponds to down-scattering in energy; and the red corresponds to up-scattering in energy.

### 2.6.1.1 Source Iteration

One simple method to invert  $(\mathbf{I}-\mathbf{T})$  of Eq. (2.66) is the *source iteration* technique, also known as *richardson iteration*.

$$\begin{aligned}\Psi^{(\ell+1)} &= \mathbf{L}^{-1} \left( \mathbf{M}\Sigma\Phi^{(\ell)} + \mathbf{Q} \right) \\ \Phi^{(\ell+1)} &= \mathbf{D}\Psi^{(\ell+1)}\end{aligned}\tag{2.68}$$

### 2.6.1.2 Krylov Subspace Methods - GMRES

## 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 GMRES 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 serveral 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*

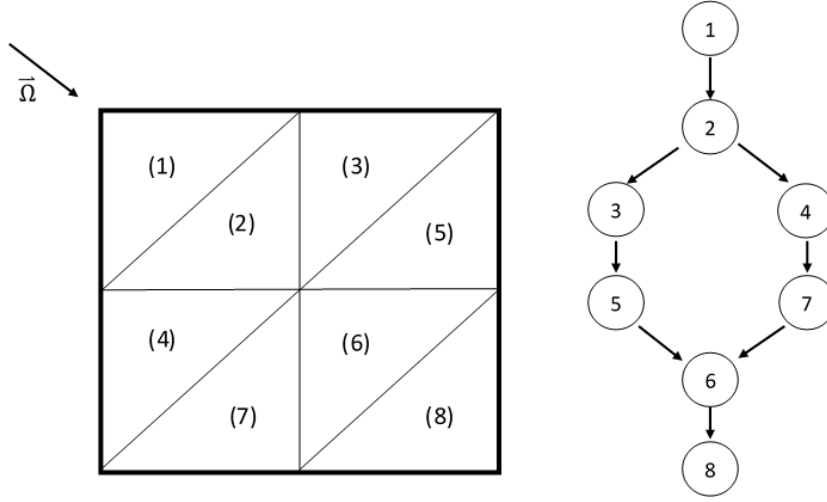


Figure 2.10: blah.

*block jacobi* (PBJ) [16].

- The iterative solver does 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 computationally 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).



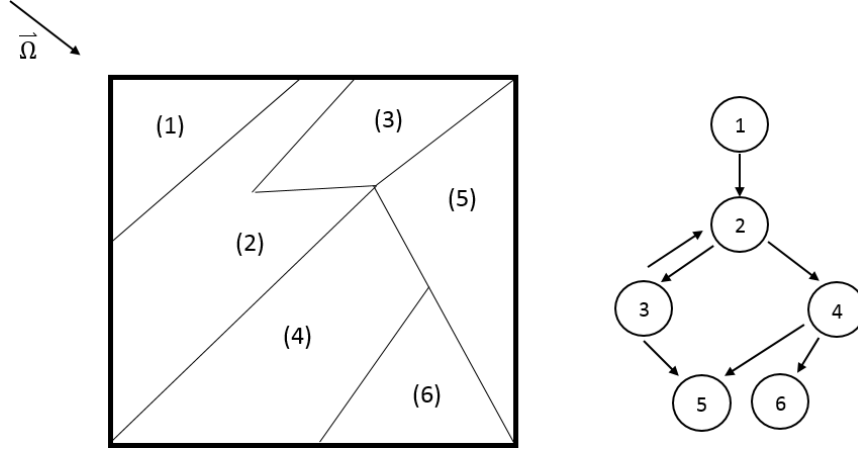


Figure 2.11: blah.

### 2.6.2.2 Polytope Grids Formed from Voronoi Mesh Generation

### 2.6.2.3 Spatial Adaptive Mesh Refinement

$$\eta_K^r = \int_{\partial K} \llbracket \Phi^r \rrbracket^2 ds = \int_{\partial K} \left( \sum_m w_m \llbracket \Psi_m^r \rrbracket \right)^2 \quad (2.69)$$

where  $\llbracket \cdot \rrbracket$  is the jump operator along a face defined as,

$$\llbracket \Phi(\vec{r}) \rrbracket = \Phi^+(\vec{r}) - \Phi^-(\vec{r}), \quad (2.70)$$

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

$$\Phi^\pm(\vec{r}) = \lim_{s \rightarrow 0^\pm} \Phi(\vec{r} + s\vec{n}). \quad (2.71)$$

In this case, the outward normal,  $\vec{n}$ , is determined with respect to the element  $K$  along its boundary,  $\partial K$ . 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:

$$\Phi^+(\vec{r}) = \begin{cases} \lim_{s \rightarrow 0^+} \Phi(\vec{r} + s\vec{n}) & \vec{r} \notin \partial\mathcal{D} \\ \sum_{\vec{\Omega}_m \cdot \vec{n} > 0} \Psi_m^-(\vec{r}) + \sum_{\vec{\Omega}_m \cdot \vec{n} < 0} \Psi_m^{inc}(\vec{r}) & \vec{r} \in \partial\mathcal{D}^d . \\ \Phi^-(\vec{r}) & \vec{r} \in \partial\mathcal{D}^r \end{cases} \quad (2.72)$$

From Eq. (2.72), 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 ( $\Psi_m^-$ ) and incident boundary fluxes ( $\Psi_m^{inc}$ ). 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.

$$\eta_K^r \geq \alpha \max_{K' \in \mathbb{T}_h^r} (\eta_{K'}^r) \quad (2.73)$$

where  $\alpha$  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.

## 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 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 presents the 3D, linearly-complete, polyhedral basis functions that will be exclusively used in Chapter ?? 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  $(1, \dots, N_K)$  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  $j + 1 = \text{mod}(j, N_K) + 1$ ,

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.  $\alpha_j$  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,  $\alpha_j$  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  $\vec{x}_{j+1}$ .

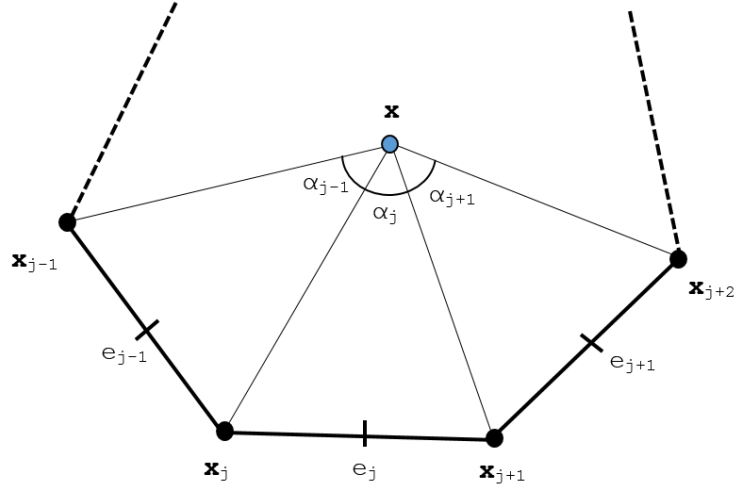


Figure 3.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

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*,

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

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

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

and they will satisfy the *Lagrange property*,

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

$N_K$  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

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

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

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

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

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

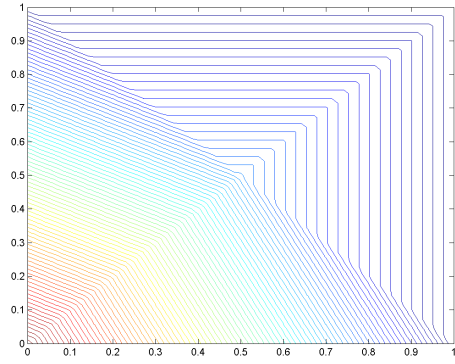
where the Wachspress weight function for vertex  $j$ ,  $w_j$ , has the following definition:

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

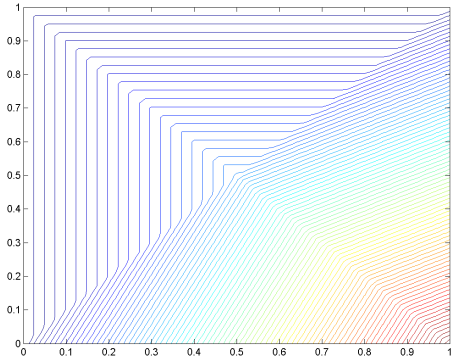
### 3.1.3 Piecewise Linear (PWL) Basis Functions

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

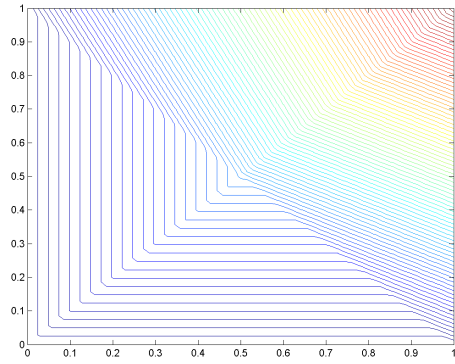
$t_j$  is the standard 2D linear function with unity at vertex  $j$  that linearly decreases to zero to the cell center and each adjoining vertex.  $t_c$  is the 2D cell “tent” function which is unity at the cell center and linearly decreases to zero to each cell vertex.



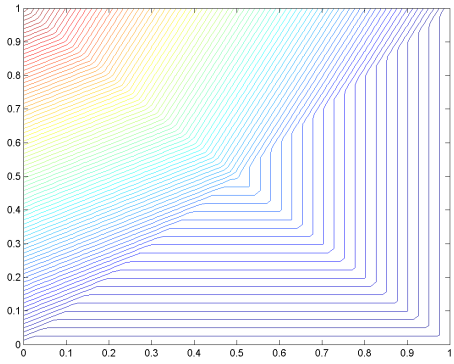
(a)



(b)



(c)



(d)

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

$\alpha_j^K$  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 [17, 18]. 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.



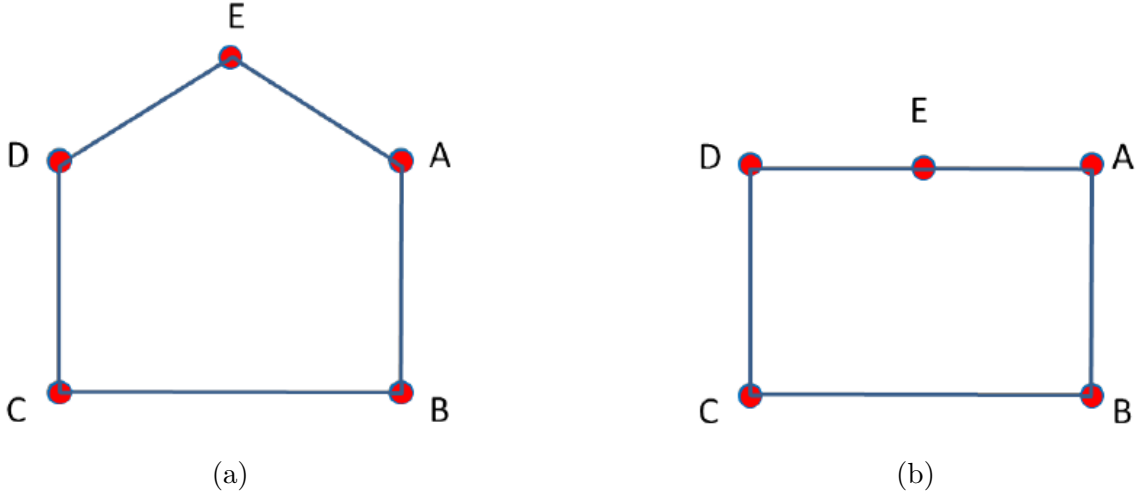


Figure 3.3: Vertex structure for a (a) regular pentagonal cell and a (b) degenerate pentagonal cell.

$$\nabla^2 u = 0, \quad (3.11)$$

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

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

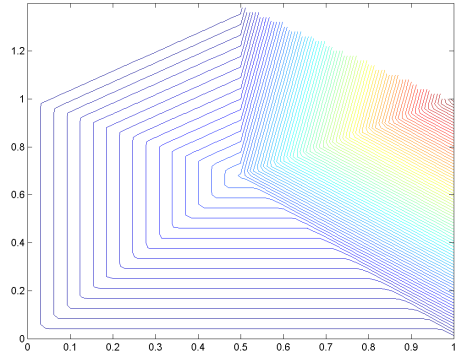
where the mean value weight function for vertex  $j$ ,  $w_j$ , has the following definition:

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

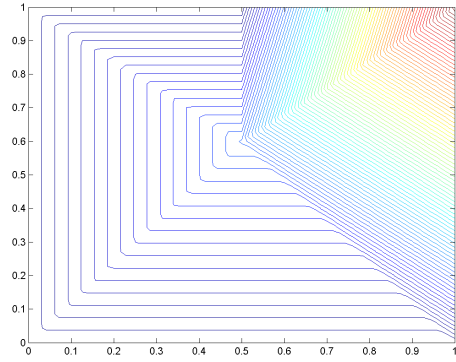
### 3.1.5 Maximum Entropy Basis Functions

The final linearly-complete 2D basis functions that we will analyze in this work are generated by use of the *maximum entropy coordinates* (ME) [19, 20, 21].

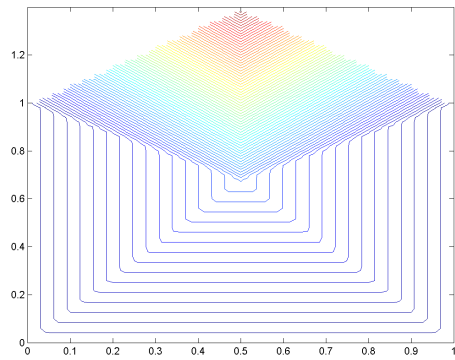
$$b_j^{ME}(\vec{x}) = \frac{w_j(\vec{x})}{\sum_i w_i(\vec{x})}. \quad (3.14)$$



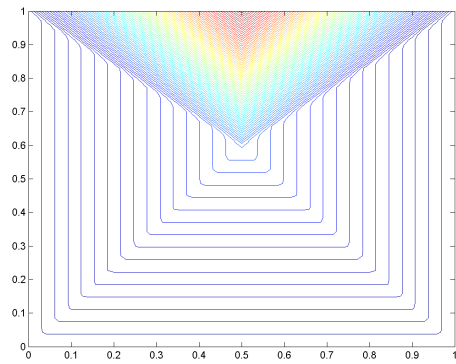
(a)



(b)

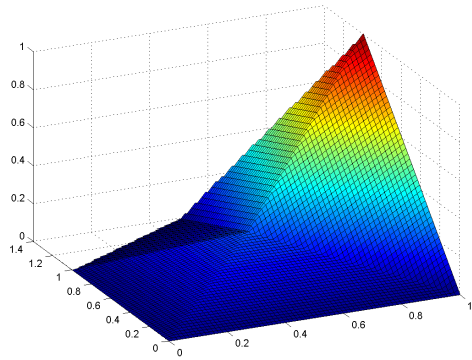


(c)

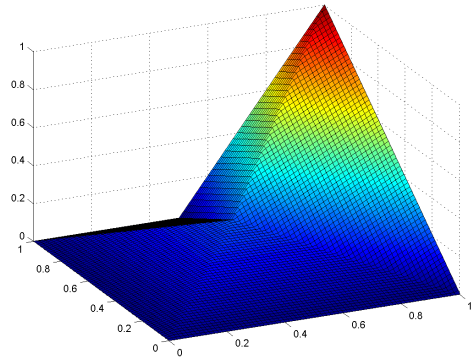


(d)

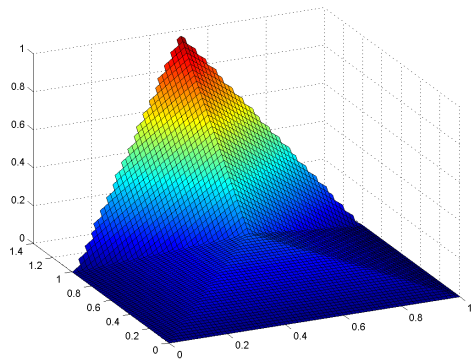
Figure 3.4: Contour plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d).



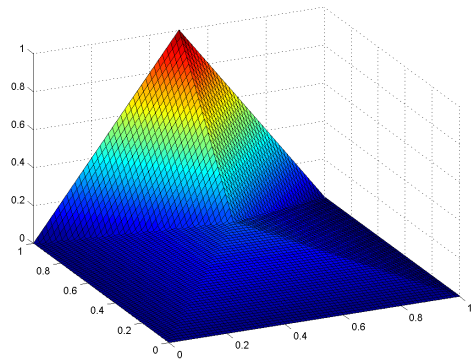
(a)



(b)



(c)



(d)

Figure 3.5: Plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d).

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

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

where  $\kappa$  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 [22, 23]. 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.

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

where

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

where

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

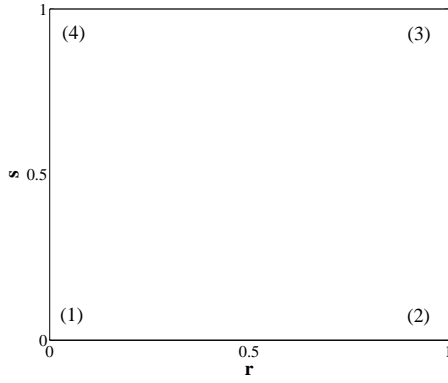
### 3.1.6 Summary of 2D Linear Basis Functions on Polygons

## 3.2 Quadratic Serendipity Basis Functions on 2D Polygons

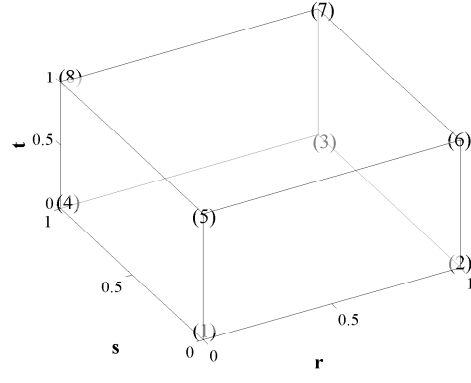
$$A^+ = A^T (AA^T)^{-1} \quad (3.19)$$

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



(a)



(b)

Figure 3.6: Vertex structure for the (a) unit square and (b) unit cube.

$$x = A^+b + [I - A^+A] w \quad (3.20)$$

### 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

## 3.3 Linear Basis Functions on 3D Polyhedra

### 3.3.1 3D Linear and TriLinear Basis Functions

$$\begin{aligned}
b_1(r, s) &= 1 - r - s - t \\
b_2(r, s) &= r \\
b_3(r, s) &= s \\
b_4(r, s) &= t
\end{aligned} \tag{3.21}$$

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 [24], we also choose to assume the cell and face weighting parameters are

$$\alpha_{K,j} = \frac{1}{N_K} \quad \text{and} \quad \beta_{f,j} = \frac{1}{N_f}, \quad (3.24)$$

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  $\alpha$  and  $\beta$  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,

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

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:

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

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:



$$\sum_q w_q = 2\pi \quad \text{and} \quad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3.27)$$

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

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:

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

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

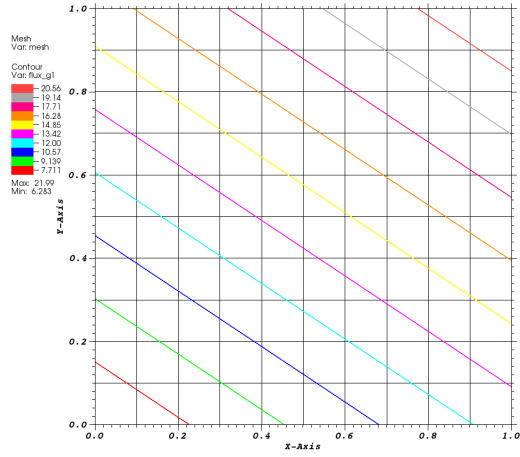
$$\begin{aligned} a &= -c\sigma_t, \\ b &= -d\sigma_t. \end{aligned} \quad (3.29)$$

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:

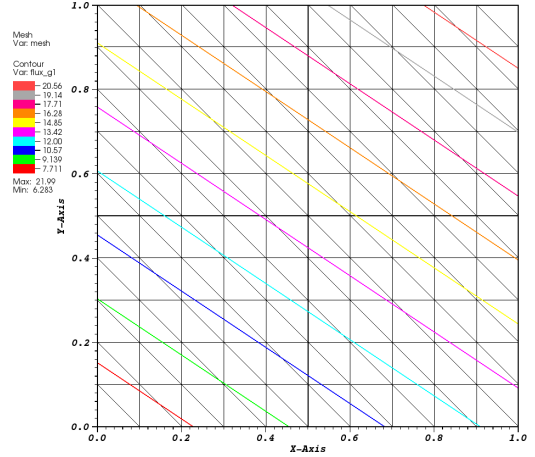
1. Orthogonal cartesian mesh formed by the intersection of 11 equally-spaced vertices in both the  $x$  and  $y$  dimensions. This forms a 10x10 array of quadrilateral mesh cells.
2. Ordered-triangular mesh formed by the bisection of the previous orthogonal cartesian mesh (forming 200 triangles all of the same size/shape).

3. Quadrilateral shestakov grid formed by the randomization of vertices based on a skewness parameter [25, 26]. 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 [27]. 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.

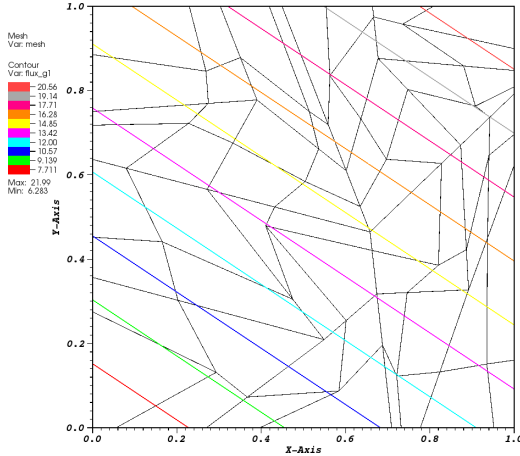
We also wish that both the angular flux solution as well as the 0th moment solution are strictly positive everywhere. Therefore, we set the function parameters in Eq. (3.26) to  $\sigma_t = a = c = d = e = 1.0$  and  $b = 1.5$ . We gave the solution the 40% tilt in space ( $a \neq b$ ) so that it would 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 3.7, 3.8, 3.9, 3.10, and 3.11, 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.



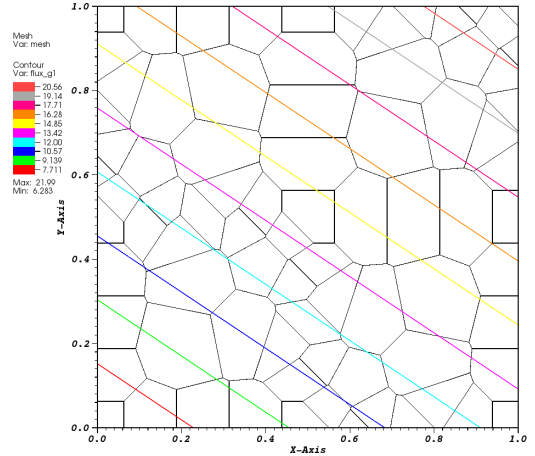
(a)



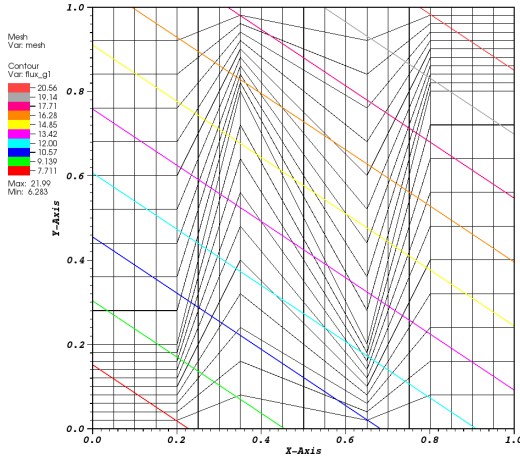
(b)



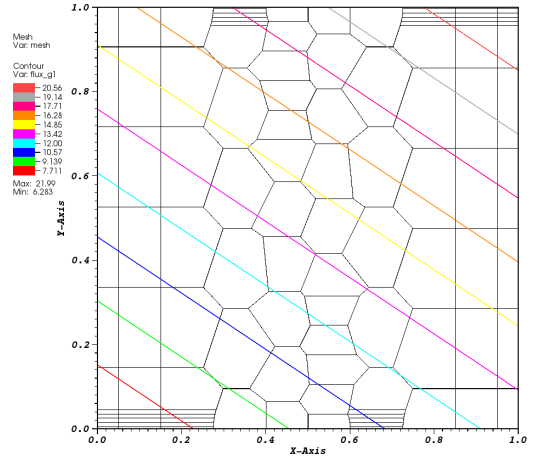
(c)



(d)

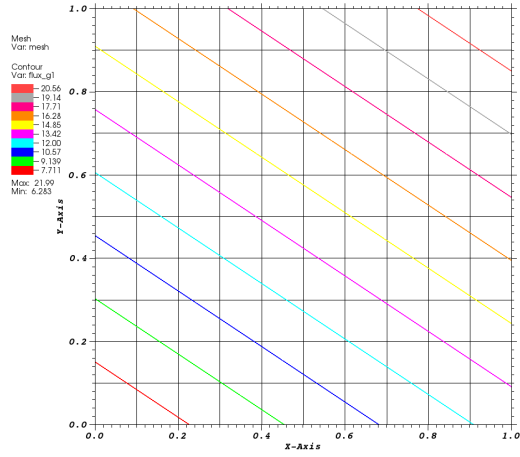


(e)

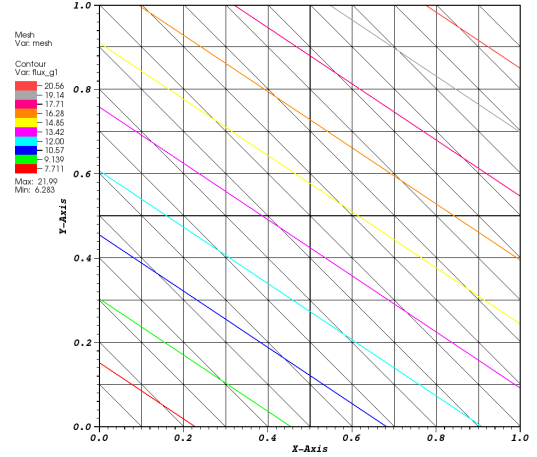


(f)

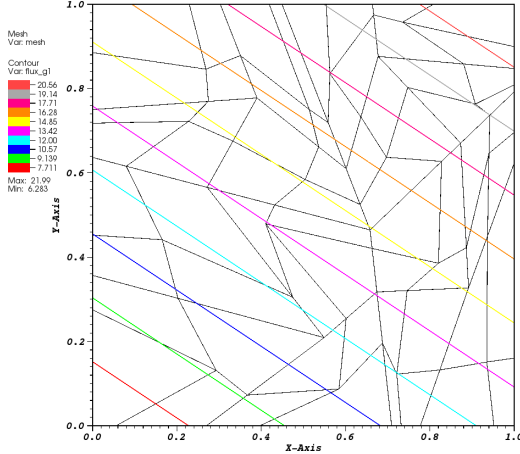
Figure 3.7: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.



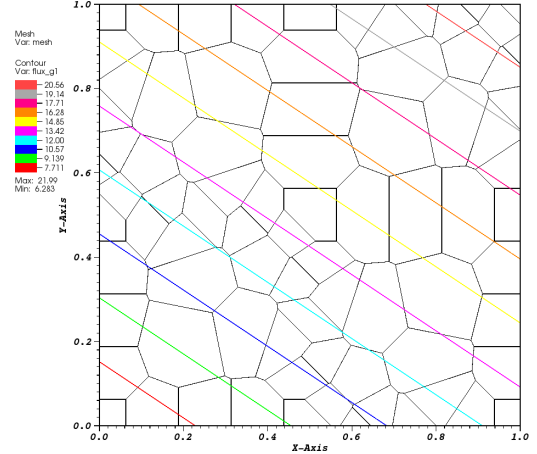
(a)



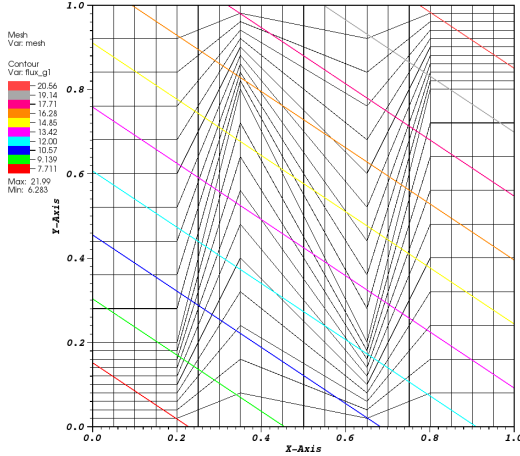
(b)



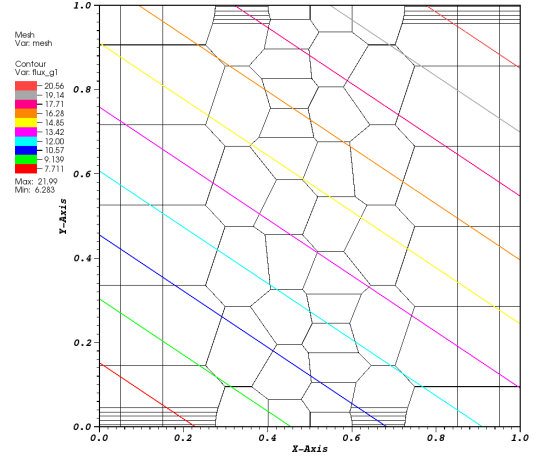
(c)



(d)

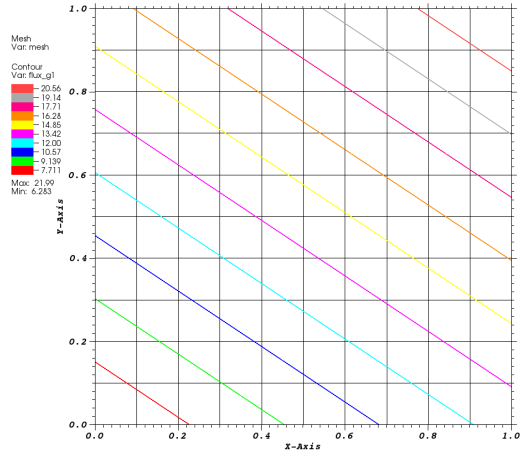


(e)

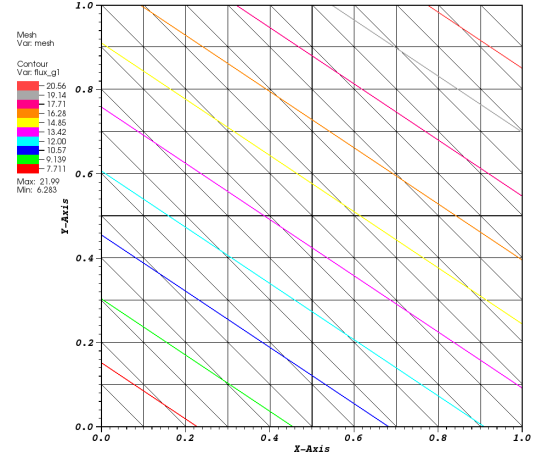


(f)

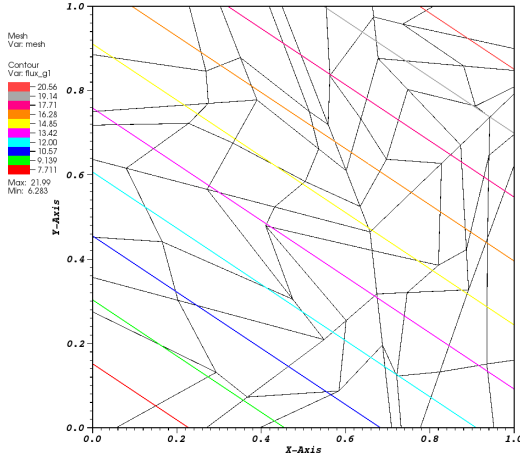
Figure 3.8: Contour plots of the exactly-linear solution with the PWL basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.



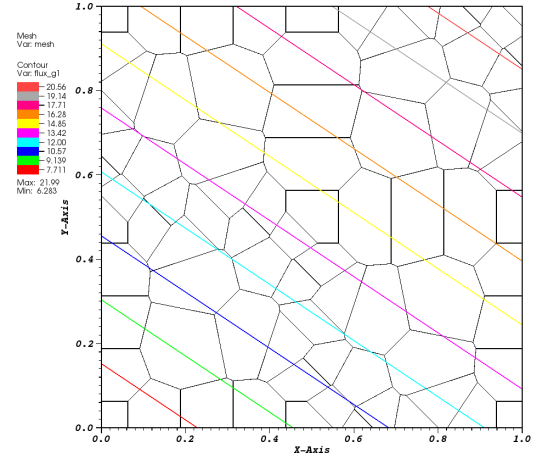
(a)



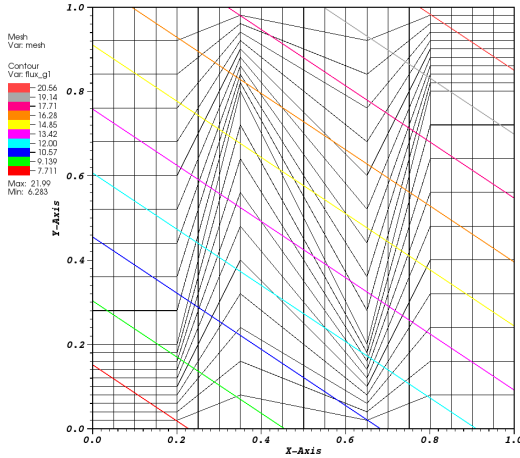
(b)



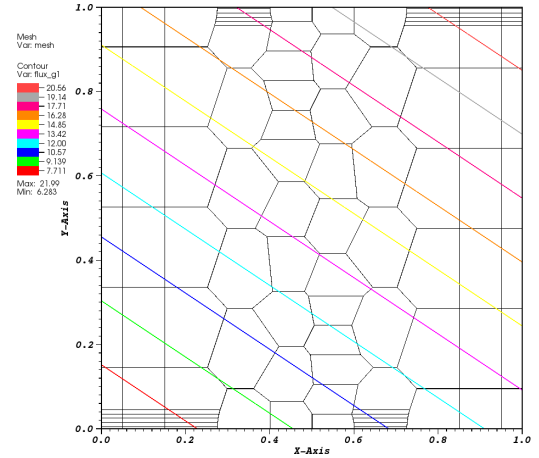
(c)



(d)

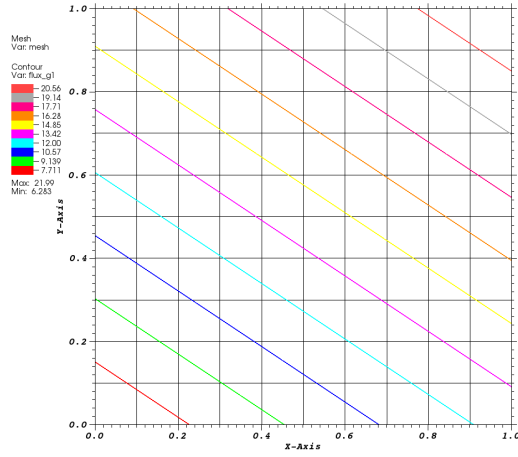


(e)

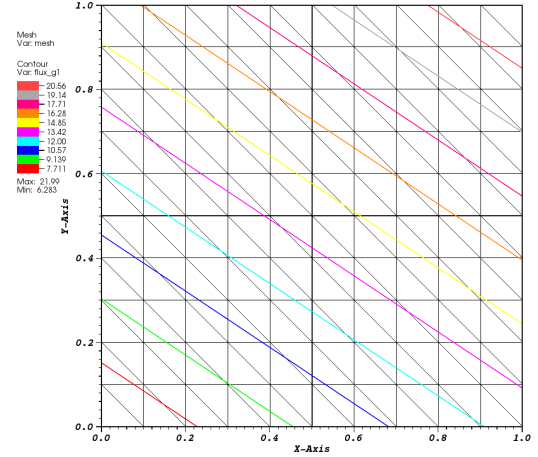


(f)

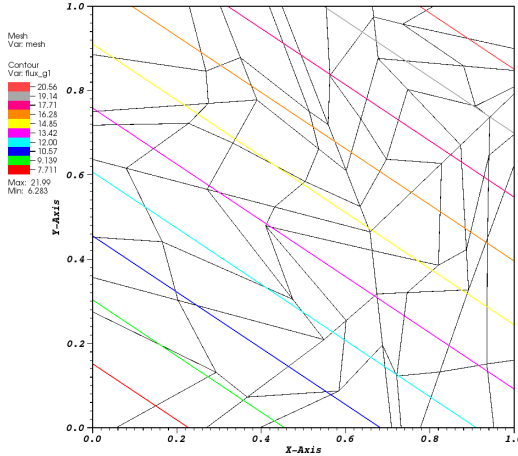
Figure 3.9: Contour plots of the exactly-linear solution with the mean value basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.



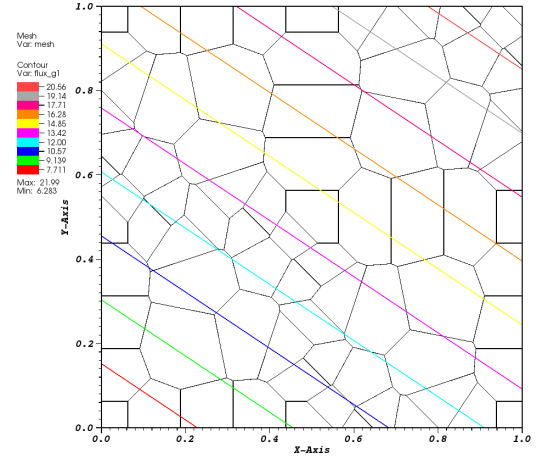
(a)



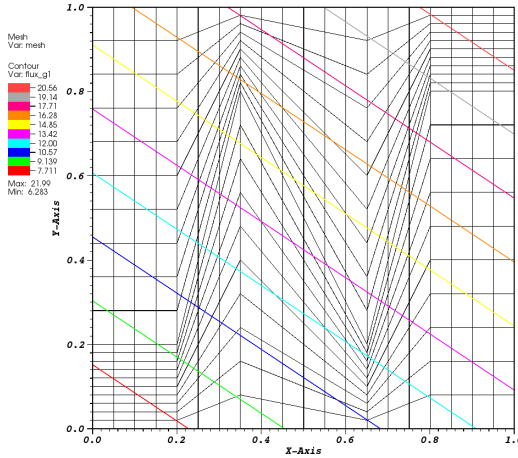
(b)



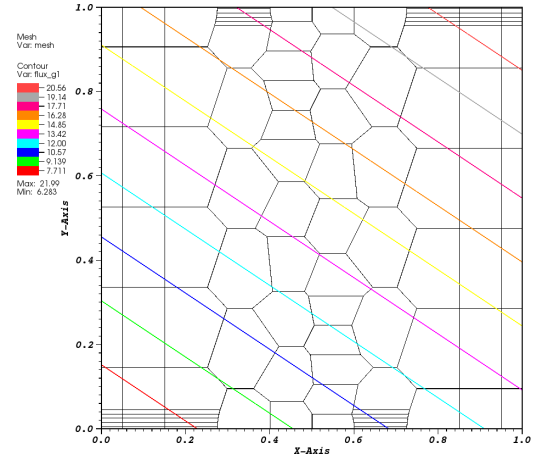
(c)



(d)

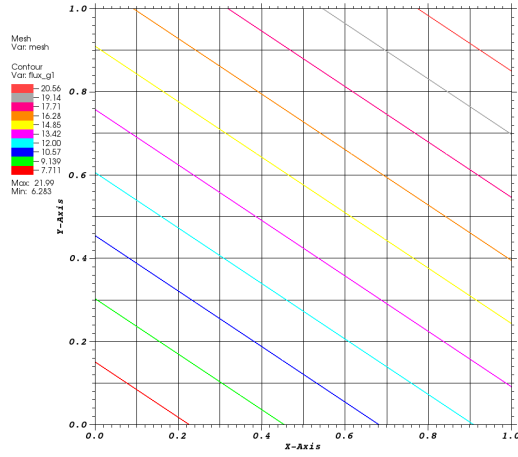


(e)

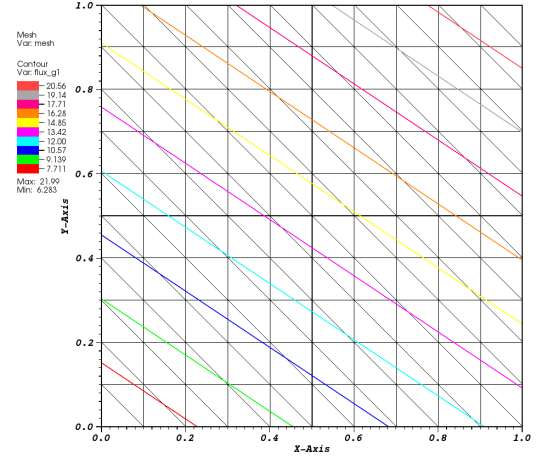


(f)

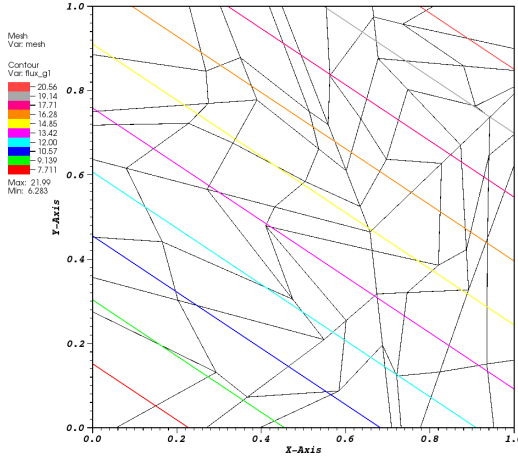
Figure 3.10: Contour plots of the exactly-linear solution with the linear maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.



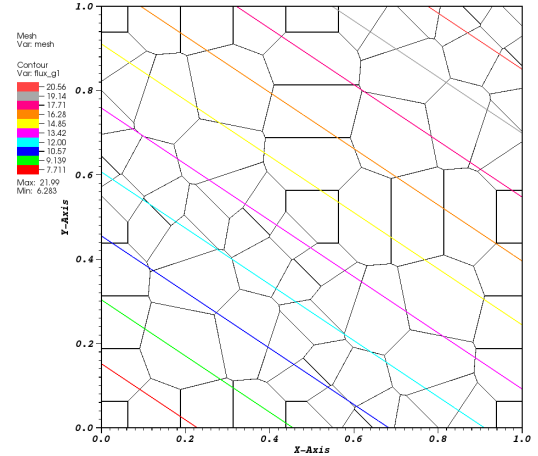
(a)



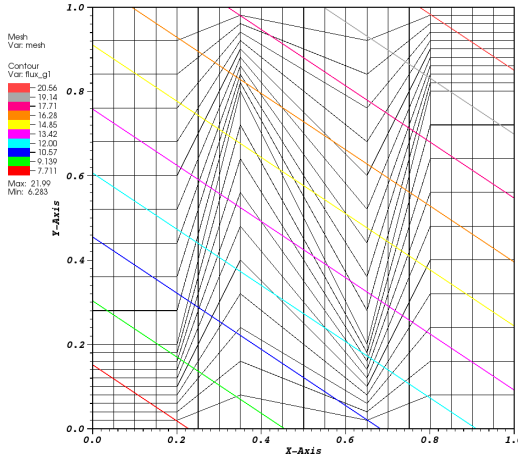
(b)



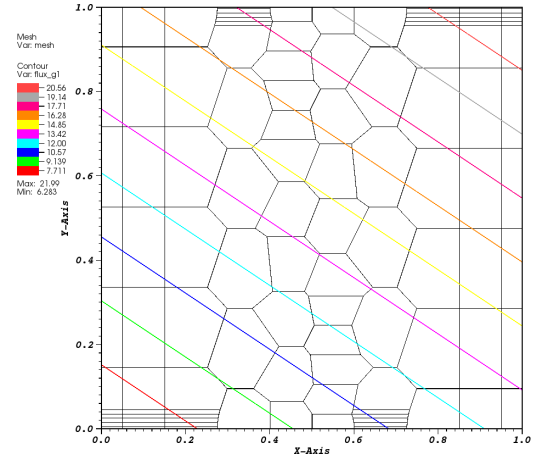
(c)



(d)



(e)



(f)

Figure 3.11: 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.

### 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 [28] lemma, the difference between the exact solution,  $\Phi_e$ , and the discretized solution,  $\Phi_h$ , residing in the Sobolev space,  $W_{\mathcal{D}}^h \in \mathcal{T}_h$ , for a given polynomial interpolation order,  $k$ , is

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

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 [29]. For transport solutions with 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 ( $N_{dof}$ ),

$$N_{dof} \propto h^{-d}, \quad (3.31)$$

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:



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

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  $C^\infty$  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,

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

where  $\nu$  is a frequency parameter. We restrict this parameter to positive integers ( $\nu = 1, 2, 3, \dots$ ) 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,

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

where the constants in the equations are:

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

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,  $\nu$ , 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  $-2\pi$  and  $2\pi$ , 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 [30] and Wang and Ragusa [31]. In this problem, an incident beam of neutrons is shined onto a small portion of a boundary, propagates

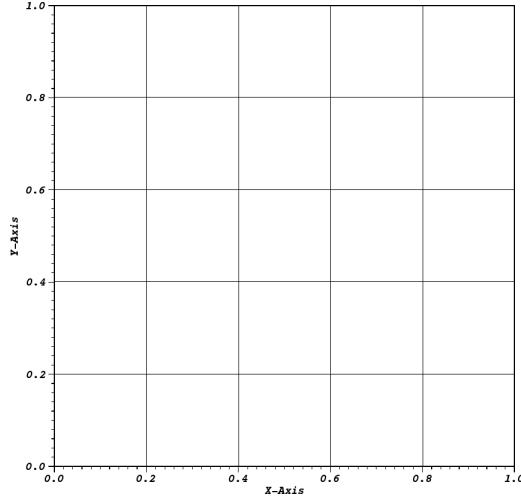


Figure 3.12: Initial mesh configuration for the searchlight problem before any refinement cycles.

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 the 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

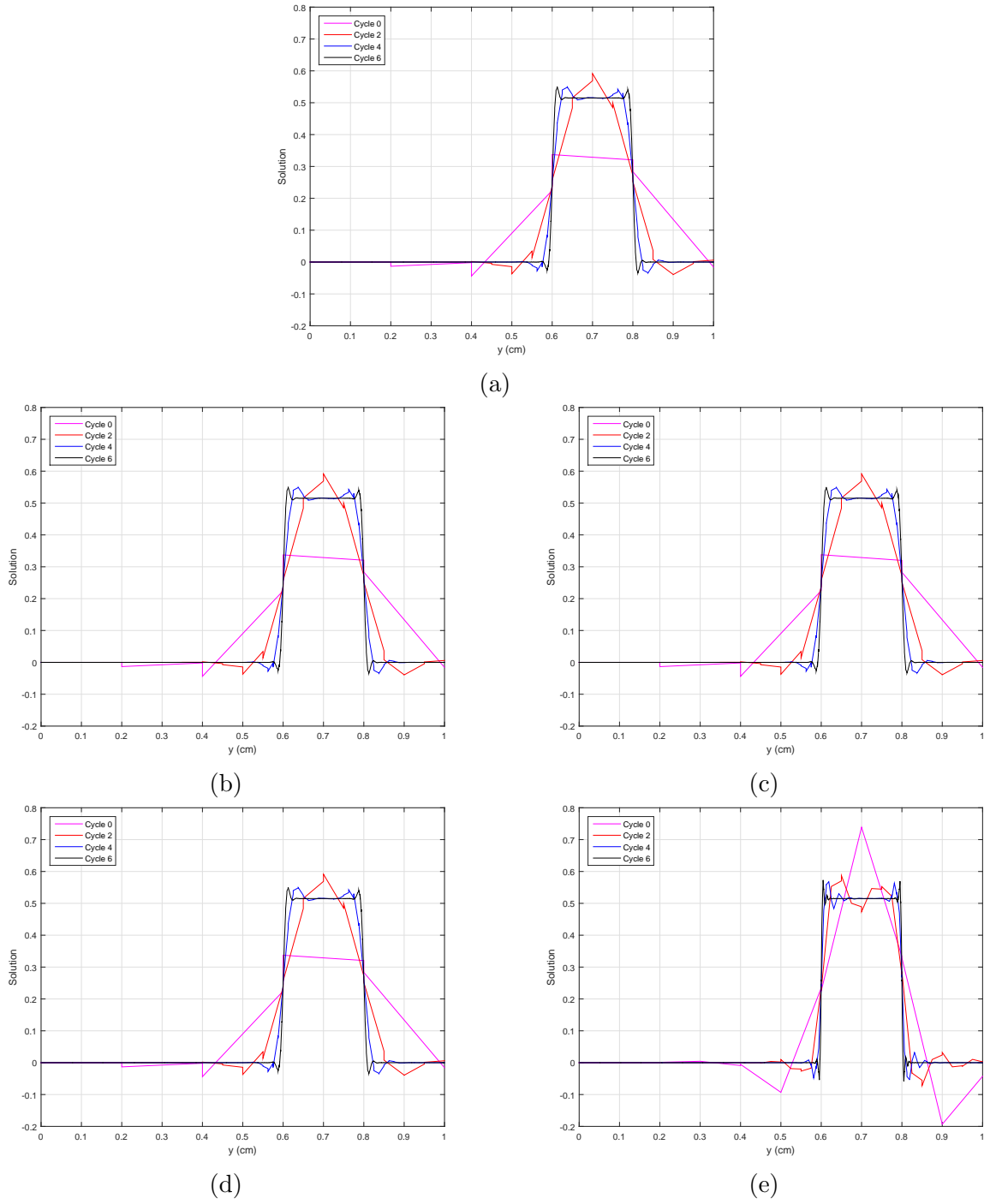


Figure 3.13: Exiting angular flux on the right boundary with uniform refinement using the (a) Wachspress basis functions, (b) PWL basis functions, (c) mean value basis functions, (d) linear maximum entropy coordinates and (e) quadratic serendipity maximum entropy coordinates.

## 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 Commission (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*, Elsevier Butterworth-Heinemann (2005).
- [15] J. AKIN, *Application and implementation of finite element methods*, Academic Press, Inc. (1982).
- [16] 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).
- [17] M. S. FLOATER, “Mean value coordinates,” *Computer aided geometric design*, **20**, 1, 19–27 (2003).
- [18] K. HORMANN and M. S. FLOATER, “Mean value coordinates for arbitrary planar polygons,” *ACM Transactions on Graphics (TOG)*, **25**, 4, 1424–1441 (2006).
- [19] N. SUKUMAR, “Construction of polygonal interpolants: a maximum entropy approach,” *International Journal for Numerical Methods in Engineering*, **61**, 12, 2159–2181 (2004).
- [20] 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).

- [21] K. HORMANN and N. SUKUMAR, “Maximum entropy coordinates for arbitrary polytopes,” in “Computer Graphics Forum,” Wiley Online Library (2008), vol. 27, pp. 1513–1520.
- [22] S. KULLBACK and R. A. LEIBLER, “On information and sufficiency,” *The annals of mathematical statistics*, pp. 79–86 (1951).
- [23] E. T. JAYNES, “Information theory and statistical mechanics,” in “Statistical Physics,” (1963), vol. 3, pp. 181–218.
- [24] 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).
- [25] A. SHESTAKOV, J. HARTE, and D. KERSHAW, “Solution of the diffusion equation by finite elements in lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **76**, 2, 385–413 (1988).
- [26] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, “Test problems in radiative transfer calculations,” *Nuclear science and engineering*, **105**, 1, 88–104 (1990).
- [27] D. S. KERSHAW, “Differencing of the diffusion equation in Lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **39**, 2, 375–395 (1981).
- [28] 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).
- [29] 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).

- [30] 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).
- [31] 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).