

HIGHER-ORDER DFEM 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

ABSTRACT

blah...

DEDICATION

For the Greater Glory of God (AMDG).

“Good ideas are not adopted automatically. They must be driven into practice with courageous impatience. Once implemented, they can be easily overturned or subverted through apath or lack of follow-up - so a continuous effort is required.”

- Admiral Hyman G. Rickover

ACKNOWLEDGEMENTS

I would like to thank my graduate advisor and committee chair, Dr. Jean Ragusa for all of his guidance towards the completion of this research endeavour. I would also like to thank my other committee members: Dr. Marvin Adams, Dr. Jim Morel, Dr. Nancy Amato, and Dr. Troy Becker for all of their support.

This research was performed under appointment to the Rickover Graduate Fellowship Program in Nuclear Engineering sponsored by the Naval Reactors Division of the United States Department of Energy.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
1. INTRODUCTION	1
1.1 Purpose of the Dissertation	1
1.2 Past Work	1
1.3 Current Work	1
1.4 Organization of the Dissertation	1
2. THE DGFEM FORMULATION OF THE MULTIGROUP S_N EQUATIONS	2
2.1 The Neutron Transport Equation	3
2.2 Energy Discretization	5
2.3 Angular Discretization	9
2.3.1 Level-Symmetric Quadrature Set	10
2.3.2 Product Gauss-Legendre-Chebyshev Quadrature Set	10
2.4 Spatial Discretization	13
2.4.1 Elementary Mass Matrices	19
2.4.2 Elementary Streaming Matrices	21
2.4.3 Elementary Surface Matrices	22
2.5 Solution Procedures	22
2.5.1 Angle and Energy Iteration Procedures	23
2.5.2 Spatial Solution Procedures - Transport Sweeping	23
2.6 Conclusions	23
3. FEM BASIS FUNCTIONS FOR UNSTRUCTURED POLYTOPES	25

3.1	Linear Basis Functions on 2D Polygons	25
3.1.1	Linear and BiLinear Basis Functions	26
3.1.2	Wachspress Rational Basis Functions	27
3.1.3	Mean Value Basis Functions	27
3.1.4	Maximum Entropy Basis Functions	27
3.1.5	Piecewise Linear (PWL) Basis Functions	27
3.1.6	Summary of 2D Linear Basis Functions on Polygons	28
3.2	Quadratic Serendipity Basis Functions on 2D Polygons	28
3.3	Linear Basis Functions on 3D Polyhedra	29
3.3.1	3D Linear and TriLinear Basis Functions	29
3.3.2	3D Piecewise Linear (PWL) Basis Functions	32
3.4	Numerical Results	33
3.4.1	Two-Dimensional Exactly-Linear Transport Solutions	33
3.4.2	Convergence Rate Analysis by the Method of Manufactured Solutions	36
3.4.3	Searchlight Problem	42
3.5	Conclusions	43
4.	DIFFUSION SYNTHETIC ACCELERATION FOR DISCONTINUOUS FINITE ELEMENTS ON UNSTRUCTURED GRIDS	45
4.1	Introduction	45
4.1.1	History	45
4.1.2	Theory	45
4.2	Symmetric Interior Penalty Form of the Diffusion Equation	45
4.3	Modified Interior Penalty Form of the Diffusion Equation used for Diffusion Synthetic Acceleration Applications	49
4.4	Multigroup Applications of Diffusion Synthetic Acceleration	49
4.5	Fourier Analysis	51
4.6	Numerical Results	51
4.6.1	Transport Solutions in the Thick Diffusive Limit	52
4.6.2	SIP used as a Diffusion Solver	52
4.6.2.1	Geometry Specification for the SIP Problems	53
4.6.2.2	Purely-Linear Solution	53
4.6.2.3	Method of Manufactured Solutions	53
4.6.3	Fourier Analysis	58
4.6.3.1	Homogeneous Medium Case	58
4.6.3.2	Periodic Horizontal Interface (PHI) Problem	58
4.6.4	MIP Results	58
4.6.4.1	Simple Homogeneous Problem Results	59
4.6.4.2	Heterogenous Problem Results	59
4.7	Conclusions	59

5. CONCLUSIONS AND OPEN ITEMS	60
5.1 Conclusions	60
5.2 Open Items	60
REFERENCES	61

LIST OF FIGURES

FIGURE	Page
2.1 Interval structure of the multigroup methodology.	6
2.2 Level-Symmetric angular quadrature set	11
2.3 Product Gauss-Legendre-Chebyshev angular quadrature set	14
2.4 Two cells of the spatial discretization	16
2.5 Scattering matrices with and without upscattering	24
3.1 Arbitrary polygon with geometric properties used for 2D basis function generation.	25
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).	28
3.3 Vertex structure for a (a) regular pentagonal cell and a (b) degenerate pentagonal cell.	29
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).	30
3.5 Plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d).	31
3.6 Vertex structure for the (a) unit square and (b) unit cube.	31
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.	37
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.	38

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.	39
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.	40
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.	41
3.12	Initial mesh configuration for the searchlight problem before any refinement cycles.	43
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.	44
4.1	blah	50
4.2	Maximum SIP orthogonal projections (solid) and penalty coefficients (dashed) on ordered cartesian (black) and triangular (blue) grids.	50
4.3	Fourier domain for 2D quadrilateral cells or an axial slice of 3D hexahedral cells in a regular grid.	51
4.4	Axial slices of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.	54
4.5	Extrusion of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.	55
4.6	Axial slice showing the contours for the linear solution of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.	56
4.7	blah	57

4.8	blah	57
4.9	blah	58

LIST OF TABLES

TABLE	Page
2.1 2D angle mapping from the first quadrant into the other 3 quadrants.	10
2.2 3D angle mapping from the first octant into the other 7 octants. . . .	10
4.1 Orthogonal projection, h , for different polygonal types: A_K is the area of cell K , L_e is the length of face e , and P_K is the perimeter of cell K .	48
4.2 Orthogonal projection, h , for different polyhedral types: V_K is the volume of cell K , A_e is the area of face e , and SA_K is the surface area of cell K	48

1. INTRODUCTION

1.1 Purpose of the Dissertation

1.2 Past Work

1.3 Current Work

1.4 Organization of the Dissertation

The remainder

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 layed out 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 will conclude our discretization procedures in the spatial domain. Section 2.5 will present the iterative procedures used to converge our solution space. We then present concluding remarks for the chapter in Section 2.6.

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:

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

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), 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 with this high-to-low energy group structure.

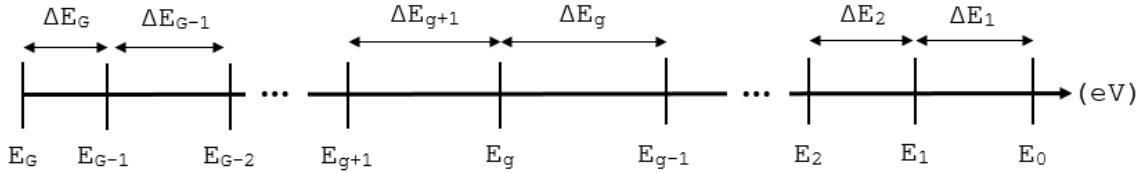


Figure 2.1: Interval structure of the multigroup methodology.

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 analogous 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

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

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):

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

where

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

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. A good example of these kinds of problems are thermal-spectrum nuclear reactors which have historically achieved reasonable solutions with as few as 4-10 energy groups.

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 angular flux in the scattering source and the distributed external source in spherical harmonics and 2) collocate the angular flux at the interpolation points of the trial space. We will perform these discretization procedures by taking the steady-state equation presented in Eq. (2.6), dropping the fission term and spatial parameterization, and using only 1 energy group:

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

We first expand the angular flux, scattering cross section and external source in spherical harmonics:

$$\begin{aligned} \Phi_{p,n} &\equiv \int_{4\pi} d\Omega \Psi(\vec{\Omega}) Y_{p,n}(\vec{\Omega}) \\ \sigma_{s,p} &\equiv \int_{-1}^1 d\mu \sigma_s(\mu) P_p(\mu) \end{aligned} \quad (2.16)$$

$$Q_{p,n} \equiv$$

where we have used the following relationships:

$$\begin{aligned} \mu &\equiv \vec{\Omega}' \cdot \vec{\Omega} \\ \sigma_s(\vec{\Omega}' \cdot \vec{\Omega}) &\equiv \frac{1}{2\pi} \sigma_s(\mu) \\ P_p(\vec{\Omega}' \cdot \vec{\Omega}) &\equiv \frac{1}{2\pi} P_p(\mu) \end{aligned} \quad (2.17)$$

Table 2.1: 2D angle mapping from the first quadrant into the other 3 quadrants.

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

Table 2.2: 3D angle mapping from the first octant into the other 7 octants.

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

We conclude our discussion of angular discretizations by presenting two common angular quadrature sets that will be employed in this dissertation work. Section 2.3.1 presents the Level Symmetric (LS) quadrature set and Section 2.3.2 presents the Product Gauss-Legendre-Chebyshev (PGLC) quadrature set.

2.3.1 Level-Symmetric Quadrature Set

The first quadrature set we present is the common Level Symmetric set.

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. It is formed by the product-wise multiplication of a Gauss-Chebyshev quadrature in the azimuthal direction and a Gauss-Legendre quadrature

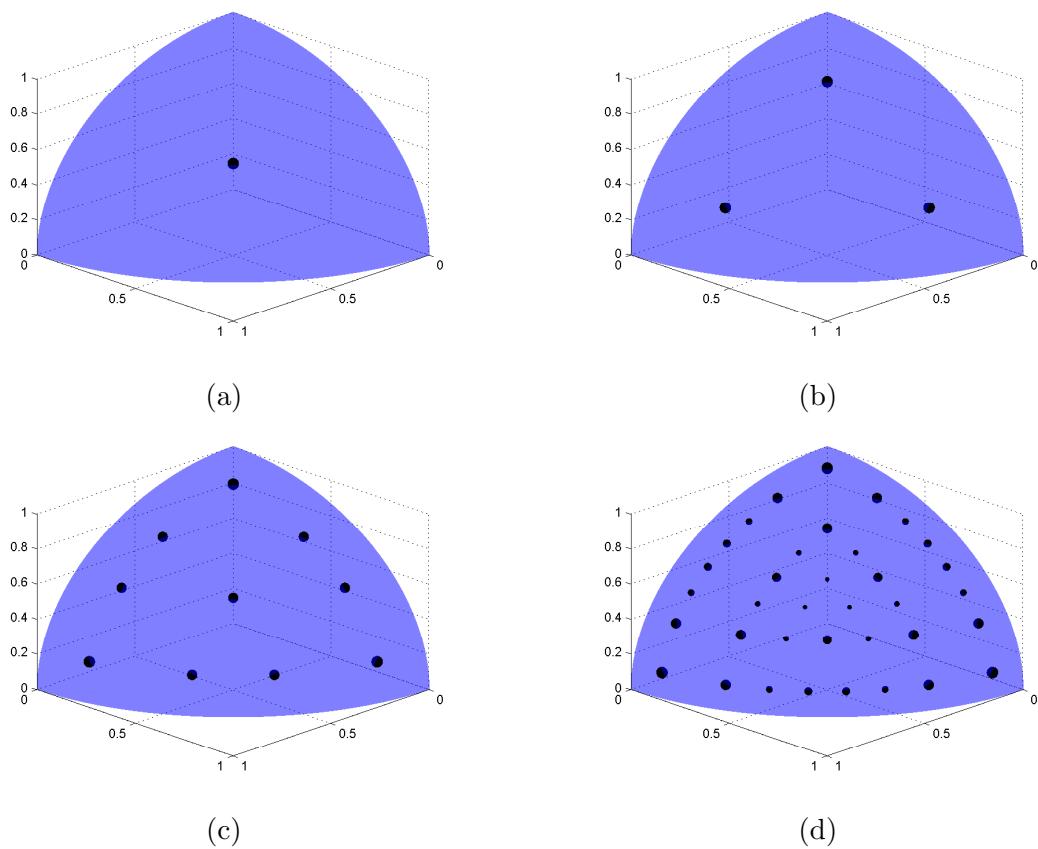


Figure 2.2: Level-Symmetric angular quadrature set of order (a) 2, (b) 4, (c) 8, and (d) 16.

in the polar direction. It has the following key differences from the Level Symmetric set:

- Does not have 90° rotational invariance within the primary octant; still maintains octant-to-octant symmetry 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.

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 2.3 with several combinations of azimuthal and polar directions. Again, the size of the direction marker corresponds to the relative weight

of 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 Spatial Discretization

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 :

$$\begin{aligned} \left(\vec{\Omega}_m \cdot \vec{\nabla} + \sigma_{t,g} \right) \Psi_{m,g} = & \sum_{g'=1}^G \sum_{p=0}^{N_p} \frac{2p+1}{4\pi} \sigma_{s,p}^{g' \rightarrow g} \sum_{n=-p}^p \Phi_{p,n,g'} Y_{p,n}(\vec{\Omega}_m) \\ & + \frac{\chi_g}{4\pi} \sum_{g'=1}^G \nu \sigma_{f,g'} \Phi_{g'} + Q_{m,g} \end{aligned} , \quad (2.18)$$

where we have dropped the spatial parameter for clarity and is beholden to the following general, discretized boundary condition:

$$\Psi_{m,g}(\vec{r}) = \Psi_{m,g}^{inc}(\vec{r}) + \sum_{g'=1}^G \sum_{\vec{\Omega}_{m'} \cdot \vec{n} > 0} \gamma_{g' \rightarrow g}^{m' \rightarrow m}(\vec{r}) \Psi_{m',g'}(\vec{r}). \quad (2.19)$$

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

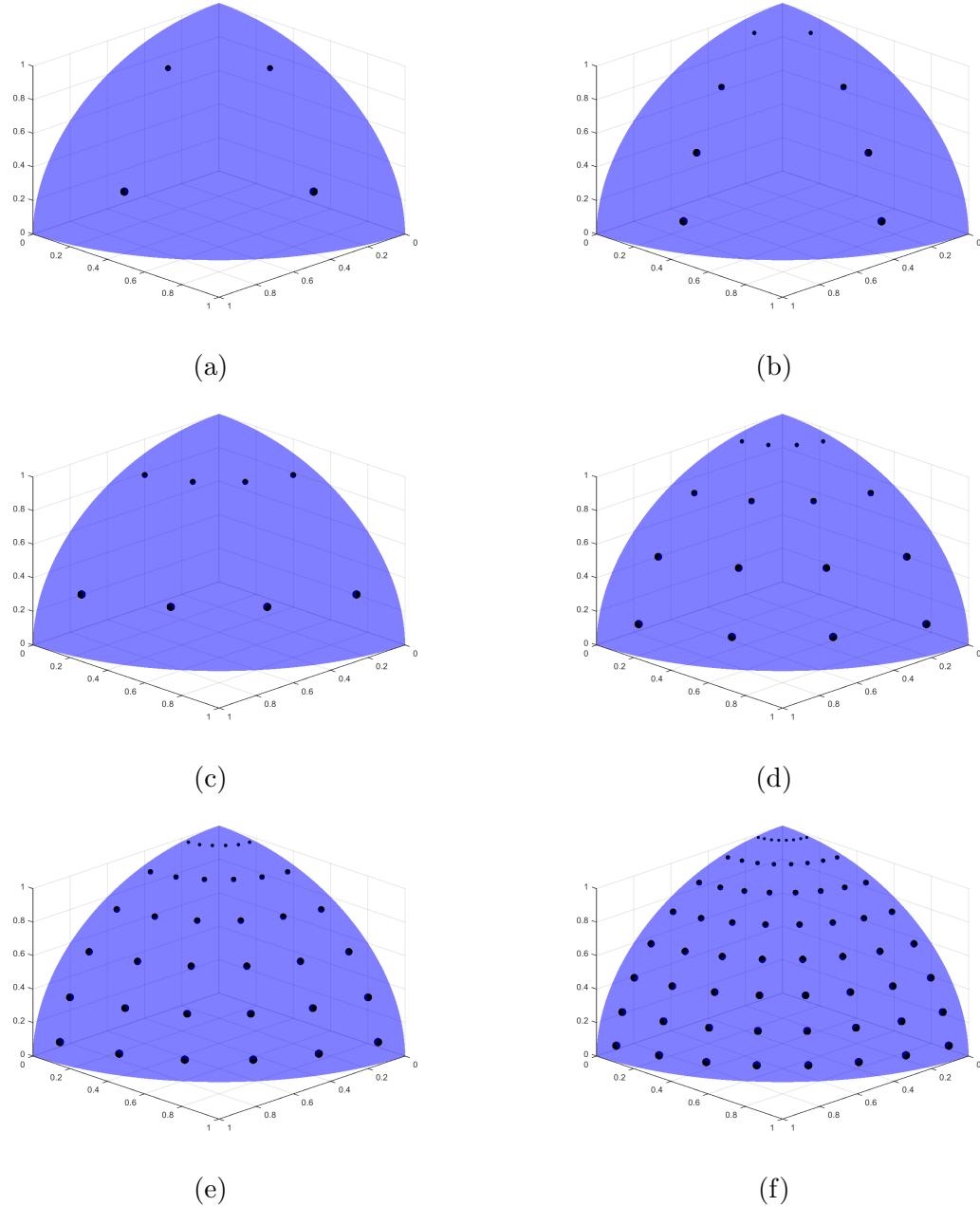


Figure 2.3: Product Gauss-Legendre-Chebyshev angular quadrature set with orders:
 (a) S_2^2 , (b) S_4^2 , (c) S_2^4 , (d) S_4^4 , (e) S_6^6 , and (f) S_8^8 .

$$\Psi_{m,g}(\vec{r}) = \begin{cases} \Psi_{m,g}^{inc}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^d \\ \Psi_{m',g}(\vec{r}), & \vec{r} \in \partial\mathcal{D}^r \end{cases} \quad (2.20)$$

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.

For the spatial discretization of the problem domain, we simplify Eq. (2.18) 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)

$$\vec{\Omega}_m \cdot \vec{\nabla} \Psi_m + \sigma_t \Psi_m = \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^p Y_{p,n}(\vec{\Omega}_m) [\sigma_{s,p} \Phi_{p,n} + Q_{p,n}] \quad (2.21)$$

to form M ($m = 1, \dots, M$) angularly discrete equations. We then lay down an unstructured mesh $\mathcal{T}_h \in \mathbb{R}^d$, 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: $\mathcal{D} = \bigcup_{K \in \mathcal{T}_h} K$. To form

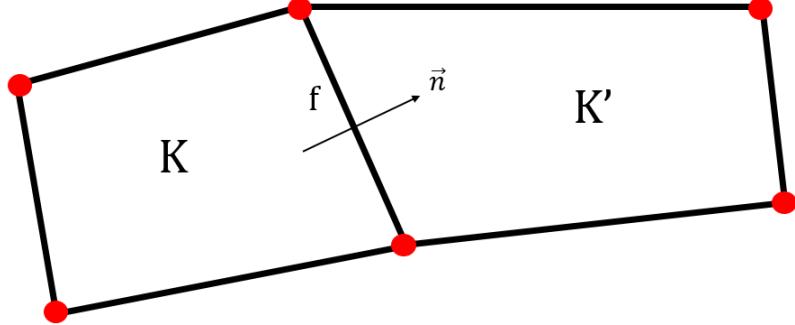


Figure 2.4: Two cells of the spatial discretization with the connecting face, f , with normal direction, \vec{n} , oriented from cell K to cell K' .

the DGFEM set of equations [7, 8], 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 2.4 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.* $\vec{n}' = -\vec{n}$).

Next, we left-multiply Eq. (2.21) by an appropriate test function b_m , integrate over cell K , and apply Gauss' Divergence Theorem to the streaming term to obtain the Galerkin weighted-residual for cell K for an angular direction $\vec{\Omega}_m$:

$$\begin{aligned}
& - \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m \right)_K + \sum_{f=1}^{N_f^K} \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \tilde{\Psi}_m \right\rangle_f + \left(\sigma_t b_m, \Psi_m \right)_K \\
& = \sum_{p=0}^{N_P} \sum_{n=-p}^p \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p} b_m, \Phi_{p,n} \right)_K + \left(b_m, Q_{p,n} \right)_K \right]
\end{aligned} \tag{2.22}$$

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:

$$\left(f, g \right)_K \equiv \int_K f g dr \tag{2.23}$$

and

$$\left\langle f, g \right\rangle_f \equiv \int_f f g ds \tag{2.24}$$

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 but we will not do so here. Therefore, with

the cell boundary terminology as proposed, Eq. (2.22) can be written into the following form:

$$\begin{aligned}
& - \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m \right)_K + \left(\sigma_t b_m, \Psi_m \right)_K \\
& + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \tilde{\Psi}_m \right\rangle_{\partial K^+} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \setminus \partial \mathcal{D}} \\
& + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^d} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \tilde{\Psi}_m \right\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\
= & \sum_{p=0}^{N_p} \sum_{n=-p}^p \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p} b_m, \Phi_{p,n} \right)_K + \left(b_m, Q_{p,n} \right)_K \right]
\end{aligned} \tag{2.25}$$

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,

$$\tilde{\Psi}_m(\vec{r}) = \begin{cases} \Psi_m^-, & \partial K^+ \\ \Psi_m^+, & \partial K^- \setminus \partial \mathcal{D} \\ \Psi_m^{inc}, & \partial K^- \cap \partial \mathcal{D}^d \\ \Psi_{m'}^-, & \partial K^- \cap \partial \mathcal{D}^r \end{cases}, \tag{2.26}$$

when the following trace is applied to the angular fluxes :

$$\Psi_m^\pm(\vec{r}) \equiv \lim_{s \rightarrow 0^\pm} \Psi_m \left(\vec{r} + s(\vec{\Omega}_m \cdot \vec{n}) \vec{\Omega}_m \right). \tag{2.27}$$

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

of DGFEM equations for cell K as

$$\begin{aligned}
& - \left(\vec{\Omega}_m \cdot \vec{\nabla} b_m, \Psi_m \right)_K + \left(\sigma_t b_m, \Psi_m \right)_K + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \Psi_m^- \right\rangle_{\partial K^+} \\
& + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \Psi_m^+ \right\rangle_{\partial K^- \setminus \partial \mathcal{D}} + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \Psi_{m'}^- \right\rangle_{\partial K^- \cap \partial \mathcal{D}^r} \\
& = \sum_{p=0}^{N_p} \sum_{n=-p}^p \frac{2p+1}{4\pi} Y_{p,n}(\vec{\Omega}_m) \left[\left(\sigma_{s,p} b_m, \Phi_{p,n} \right)_K + \left(b_m, Q_{p,n} \right)_K \right] \\
& + \left\langle (\vec{\Omega}_m \cdot \vec{n}) b_m, \Psi_m^{inc} \right\rangle_{\partial K^- \cap \partial \mathcal{D}^d}
\end{aligned} \tag{2.28}$$

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, $\mathcal{D} = \bigcup_{K \in \mathcal{T}_h} K$, while further looping over all faces within each cell, $\partial \mathcal{D}_K = \bigcup_{f \in N_f^K} f$. Section 2.5 will further detail how all the DGFEM equations are formed along with efficient solution methods. We conclude this section by defining the elementary matrix terms for a given cell as follows.

2.4.1 Elementary Mass Matrices

In the spatially discretized equations presented in Section 2.4, 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 [9, 10]. For cell K , we define the elementary mass matrix, \mathbf{M} as

$$\mathbf{M}_K = \int_K \mathbf{b}_K \mathbf{b}_K^T d\vec{r}, \tag{2.29}$$

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. (2.29) 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

$$\mathbf{M}_K = \sum_{q=1}^{N_q} w_q^K \mathbf{b}_K(\vec{x}_q) \mathbf{b}_K^T(\vec{x}_q). \quad (2.30)$$

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:

$$\mathbf{M}_K = \begin{bmatrix} \int_K b_1 b_1 & \dots & \int_K b_1 b_j & \dots & \int_K b_1 b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_i b_1 & \dots & \int_K b_i b_j & \dots & \int_K b_i b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K b_{N_K} b_1 & \dots & \int_K b_{N_K} b_j & \dots & \int_K b_{N_K} b_{N_K} \end{bmatrix}, \quad (2.31)$$

where an individual matrix entry is of the form:

$$M_{i,j}^K = \int_K b_i b_j. \quad (2.32)$$

2.4.2 Elementary Streaming Matrices

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

$$\vec{\mathbf{G}}_K = \int_K \vec{\nabla} \mathbf{b}_K \mathbf{b}_K^T d\vec{r}, \quad (2.33)$$

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 $(\vec{\Omega}_m \cdot \int_K \vec{\nabla} \mathbf{b}_K \mathbf{b}_K^T d\vec{r})$. Instead we simply evaluate the dot product with the appropriate angular direction whenever necessary. This has great benefit when trying to run large transport problems when memory becomes a premium and processor operations are not our limiting bottleneck.

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

$$\vec{\mathbf{G}}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla} \mathbf{b}_K(\vec{x}_q) \mathbf{b}_K^T(\vec{x}_q). \quad (2.34)$$

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

$$\vec{\mathbf{G}}_K = \begin{bmatrix} \int_K \vec{\nabla} b_1 b_1 & \dots & \int_K \vec{\nabla} b_1 b_j & \dots & \int_K \vec{\nabla} b_1 b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_i b_1 & \dots & \int_K \vec{\nabla} b_i b_j & \dots & \int_K \vec{\nabla} b_i b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_{N_K} b_1 & \dots & \int_K \vec{\nabla} b_{N_K} b_j & \dots & \int_K \vec{\nabla} b_{N_K} b_{N_K} \end{bmatrix}, \quad (2.35)$$

where an individual matrix entry is of the form:

$$\vec{G}_{i,j}^K = \int_K \vec{\nabla} b_i b_j. \quad (2.36)$$

2.4.3 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 analogous to the cell mass matrix but are computed on the cell boundary with dimensionality $(d - 1)$.

2.5 Solution Procedures

To this point, we have properly described the procedures to discretize the transport problem in energy, angle, and space. 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 2.5.1 will discuss the different iterative procedures that can be employed to efficiently converge the angular and energy portions of the solution space and Section 2.5.2 will then discuss how we can efficiently solve the spatial portion of the solution for any given iteration.

2.5.1 Angle and Energy Iteration Procedures

2.5.2 Spatial Solution Procedures - Transport Sweeping

2.6 Conclusions

In this chapter

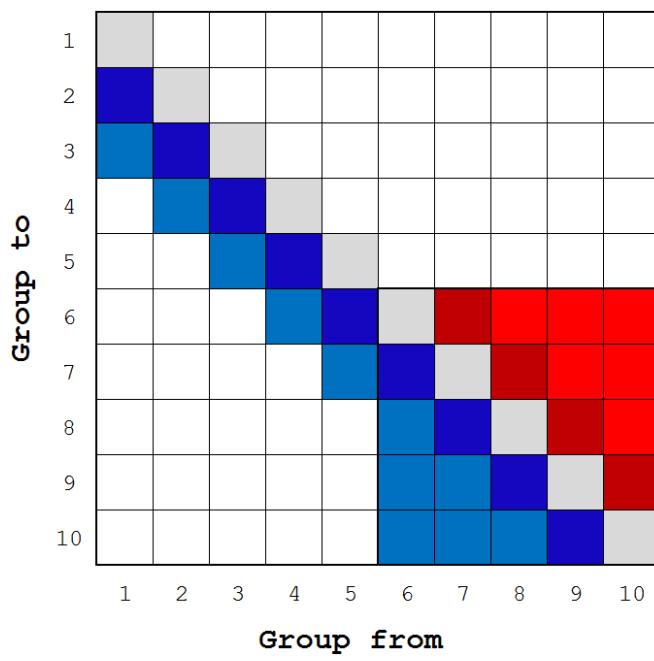
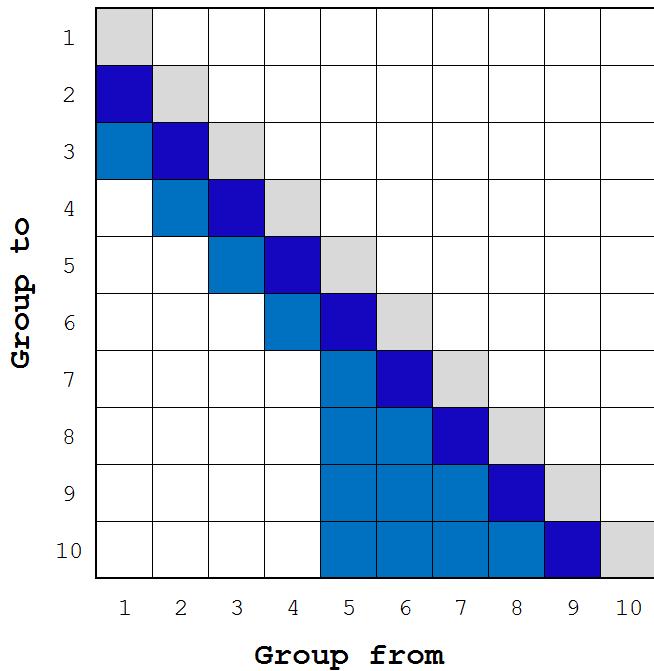


Figure 2.5: Scattering matrices (top) without and (bottom) with upscattering. The gray corresponds to within-group scattering; the blue corresponds to down-scattering in energy; and the red corresponds to up-scattering in energy.

3. FEM BASIS FUNCTIONS FOR UNSTRUCTURED POLYTOPES

discuss benefits/desires here and some history...

3.1 Linear Basis Functions on 2D Polygons

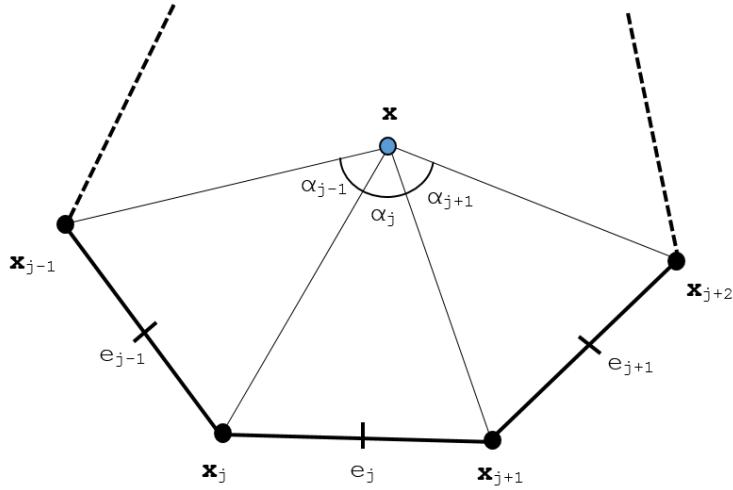


Figure 3.1: Arbitrary polygon with geometric properties used for 2D basis function generation.

In this dissertation, all 1st-order, two-dimensional basis functions for a cell 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.

3.1.1 Linear and BiLinear Basis Functions

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

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

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

3.1.3 Mean Value Basis Functions

3.1.4 Maximum Entropy Basis Functions

3.1.5 Piecewise Linear (PWL) Basis Functions

$$b_j(x, y) = t_j(x, y) + \alpha_j^K t_c(x, y) \tag{3.8}$$

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

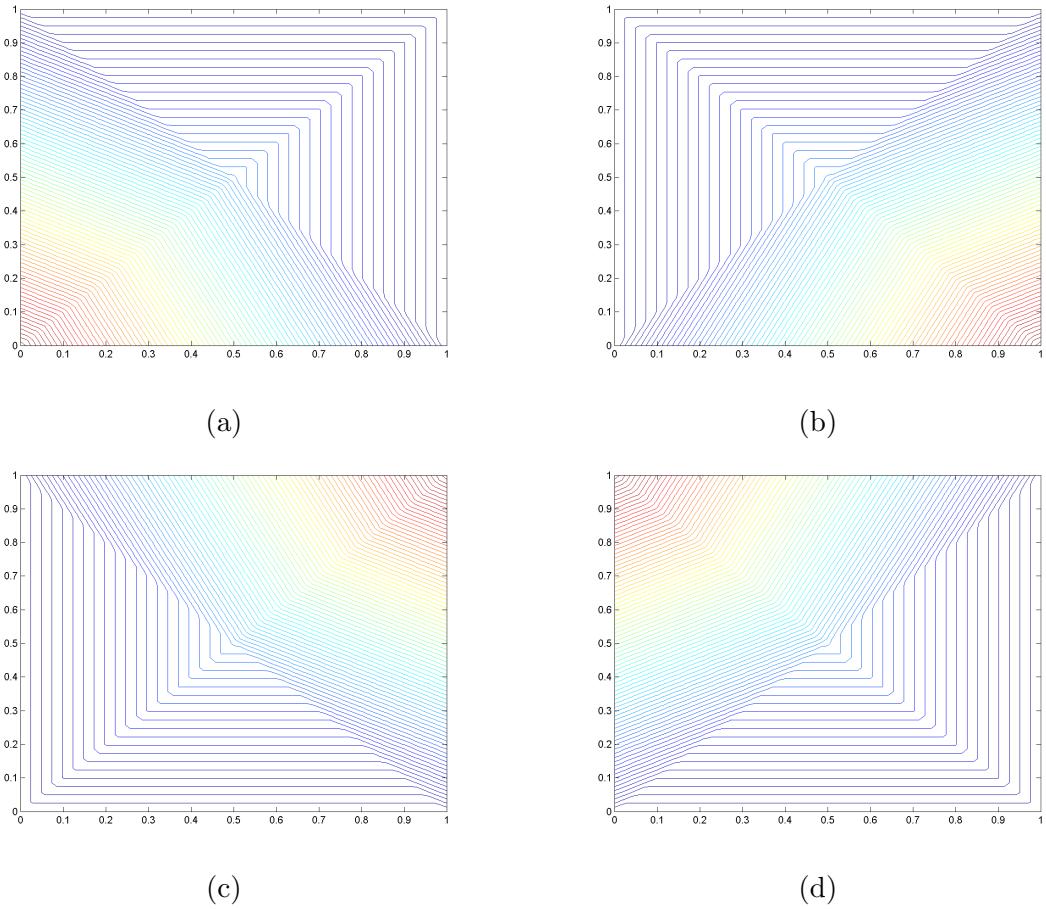


Figure 3.2: Contour plots of the PWL basis functions on the unit square for the vertices located at: (a) $(0,0)$, (b) $(1,0)$, (c) $(1,1)$, and (d) $(0,1)$.

3.1.6 Summary of 2D Linear Basis Functions on Polygons

3.2 Quadratic Serendipity Basis Functions on 2D Polygons

$$\sum_{i=1}^{N_K} b_i(\vec{x}) = 1 \quad (3.9)$$

$$\sum_{i=1}^{N_K} b_i(\vec{x}) \vec{x}_i = \vec{x} \quad (3.10)$$

$$\sum_{i=1}^{N_K} \sum_{j=1}^{N_K} \mu_{i,j} \left(\frac{\vec{x}_i \otimes \vec{x}_j + \vec{x}_j \otimes \vec{x}_i}{2} \right) = \vec{x} \otimes \vec{x} \quad (3.11)$$

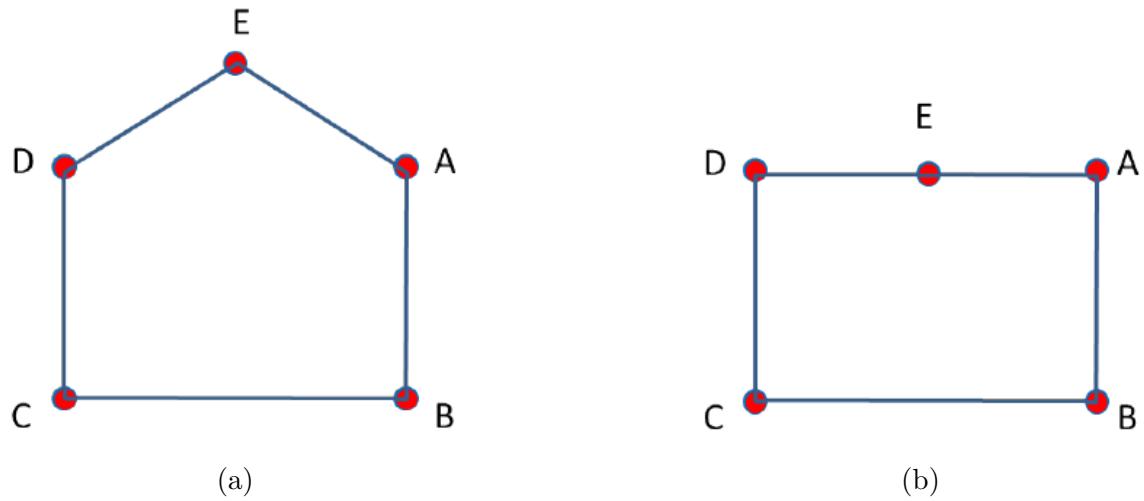


Figure 3.3: Vertex structure for a (a) regular pentagonal cell and a (b) degenerate pentagonal cell.

where $\mu_{i,j}$ is a weight function corresponding to particular basis function pairings.

3.3 Linear Basis Functions on 3D Polyhedra

3.3.1 3D Linear and TriLinear Basis Functions

$$b_1(r, s) = 1 - r - s - t$$

$$\begin{aligned} b_2(r, s) &= r \\ b_3(r, s) &= s \end{aligned} \tag{3.12}$$

$$b_4(r, s) = t$$

and

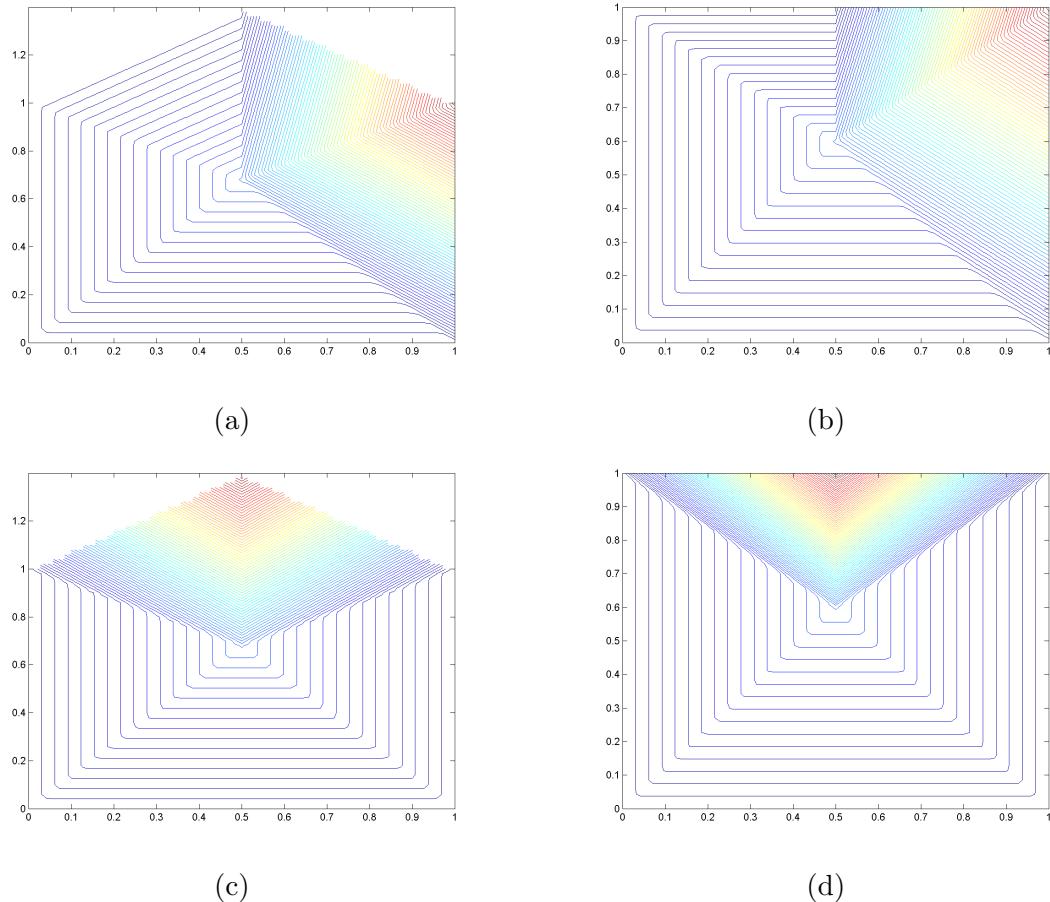
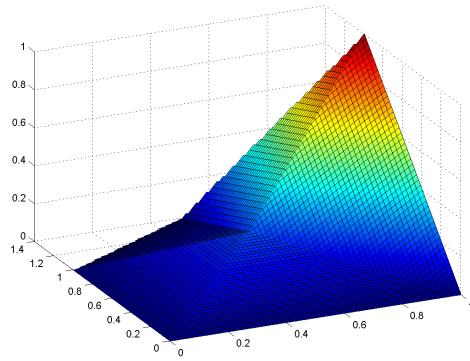
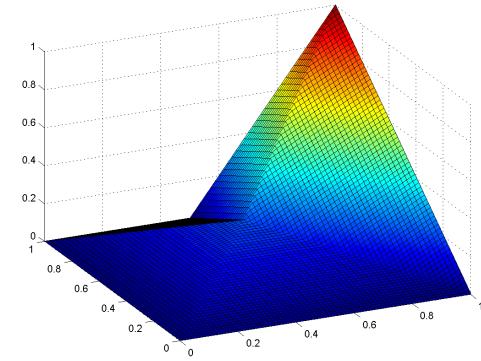


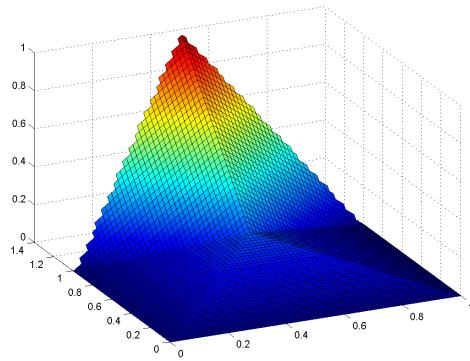
Figure 3.4: Contour plots of the PWL basis functions for a regular pentagon: (a) and (c) as well as a degenerate pentagon: (b) and (d).



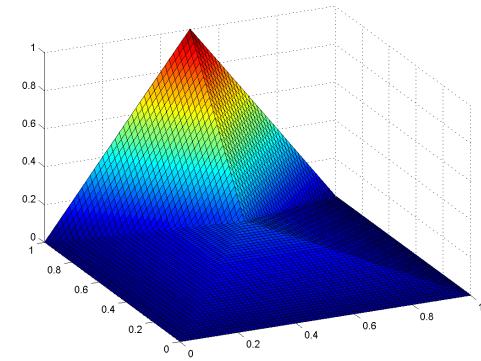
(a)



(b)

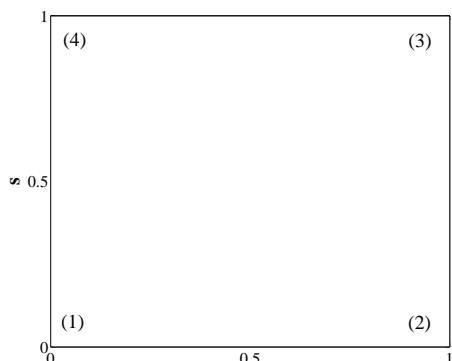


(c)

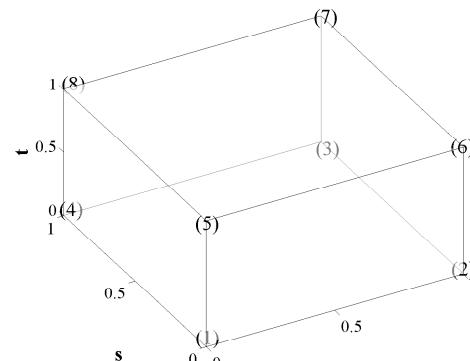


(d)

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).



(a)



(b)

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

$$\begin{aligned}
b_1(r, s, t) &= (1 - r)(1 - s)(1 - t) \\
b_2(r, s, t) &= r(1 - s)(1 - t) \\
b_3(r, s, t) &= rs(1 - t) \\
b_4(r, s, t) &= (1 - r)s(1 - t) \\
b_5(r, s, t) &= (1 - r)(1 - s)t \\
b_6(r, s, t) &= r(1 - s)t \\
b_7(r, s, t) &= rst \\
b_8(r, s, t) &= (1 - r)st
\end{aligned} \tag{3.13}$$

3.3.2 3D Piecewise Linear (PWL) Basis Functions

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

$$b_j(x, y, z) = t_j(x, y, z) + \sum_{f=1}^{F_j} \beta_f^j t_f(x, y, z) + \alpha_j^K t_c(x, y, z) \tag{3.14}$$

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

$$\alpha_{K,j} = \frac{1}{N_K} \quad \text{and} \quad \beta_{f,j} = \frac{1}{N_f}, \tag{3.15}$$

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

3.4 Numerical Results

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

3.4.1 Two-Dimensional Exactly-Linear Transport Solutions

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

$$\mu \frac{\partial \Psi}{\partial x} + \eta \frac{\partial \Psi}{\partial y} + \sigma_t \Psi = Q(x, y, \mu, \eta), \quad (3.16)$$

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 solution:

$$\begin{aligned}\Psi(x, y, \mu, \eta) &= ax + by + c\mu + d\eta + e \\ \Phi(x, y) &= 2\pi(ax + by + e)\end{aligned}. \quad (3.17)$$

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:

$$\sum_q w_q = 2\pi \quad \text{and} \quad \sum_q w_q \begin{bmatrix} \mu_q \\ \eta_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3.18)$$

Our boundary conditions for all inflow boundaries are then uniquely determined by the angular flux solution of Eq. (3.17). Inserting the angular flux solution of Eq. (3.17) into Eq. (3.16), we obtain the distributed source that will produce our exactly-linear solution space:

$$Q(x, y, \mu, \eta) = a\mu + b\eta + \sigma_t(c\mu + d\eta) + \sigma_t(ax + by + e). \quad (3.19)$$

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

$$\begin{aligned}a &= -c\sigma_t, \\ b &= -d\sigma_t.\end{aligned} \quad (3.20)$$

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 [12, 13]. 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 [14]. 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. (3.17) 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.

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 [15] lemma, the difference between the exact solution, Φ_e , and the discretized solution, Φ_h , residing in the Sobolev space, $W_D^h \in \mathcal{T}_h$, for a given polynomial interpolation order, k , is

$$\|\Phi_e - \Phi_h\|_2 \leq C \frac{h^{q+1/2}}{(k+1)^q}, \quad (3.21)$$

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 [16]. 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

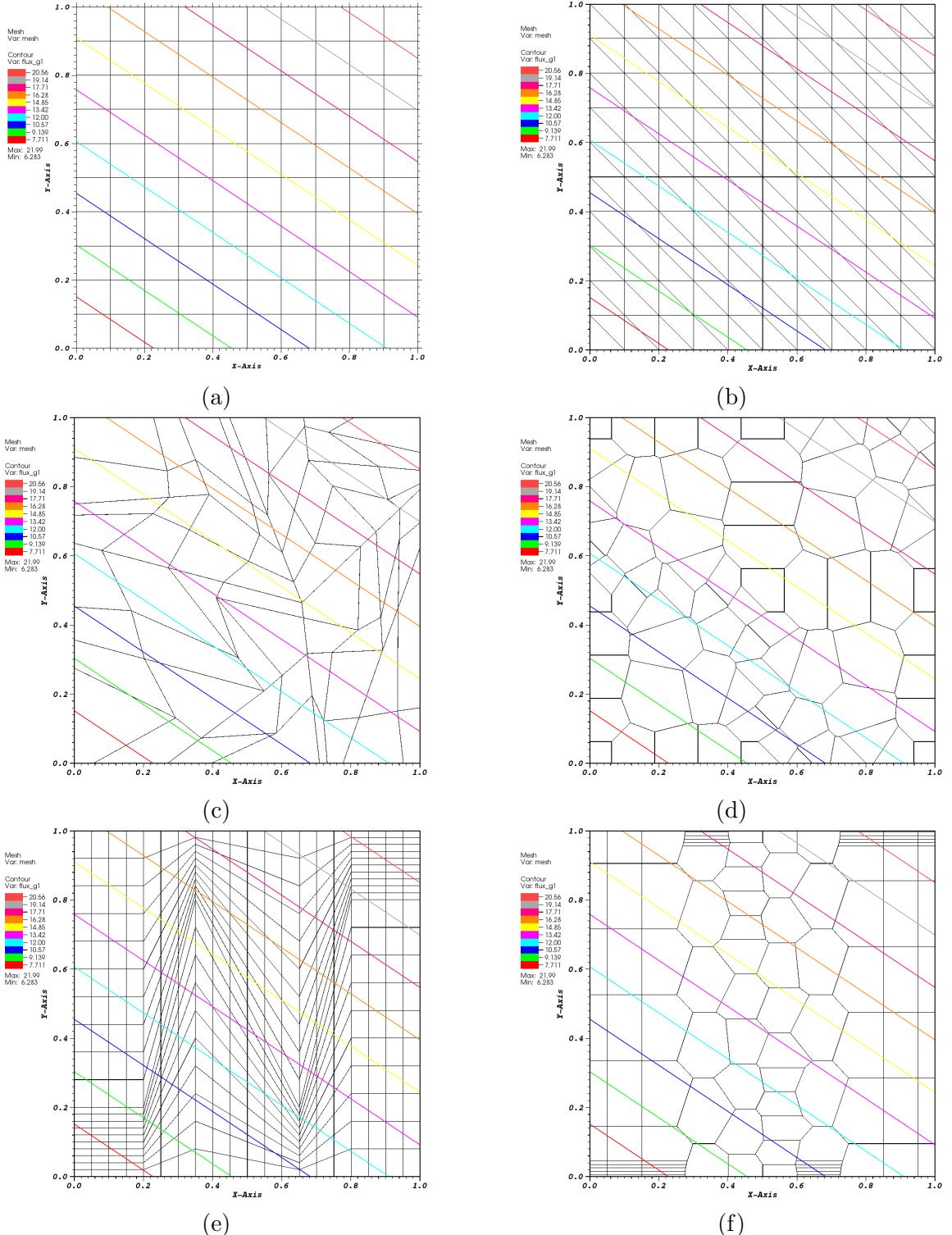


Figure 3.7: Contour plots of the exactly-linear solution with the Wachspress basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

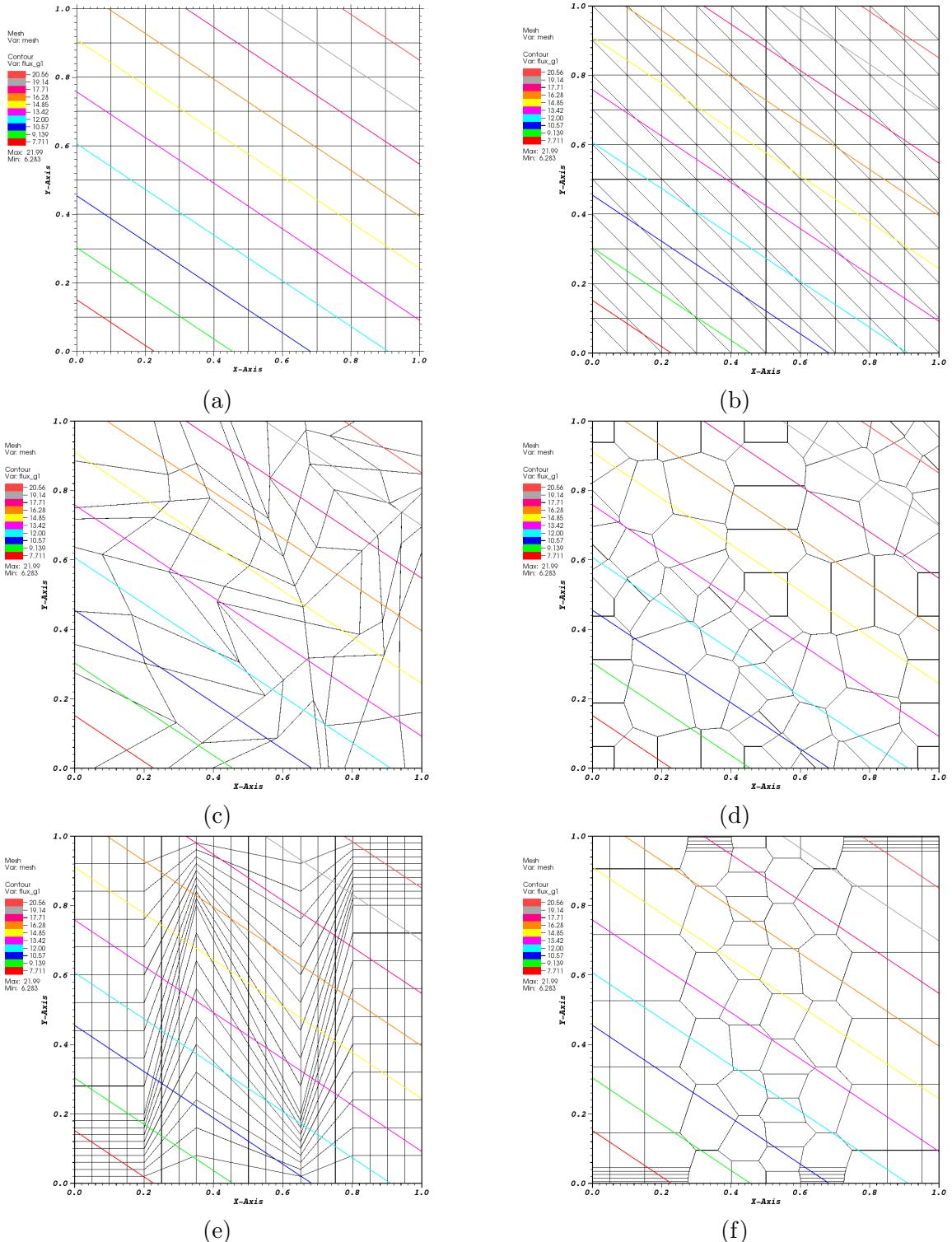


Figure 3.8: Contour plots of the exactly-linear solution with the PWL basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

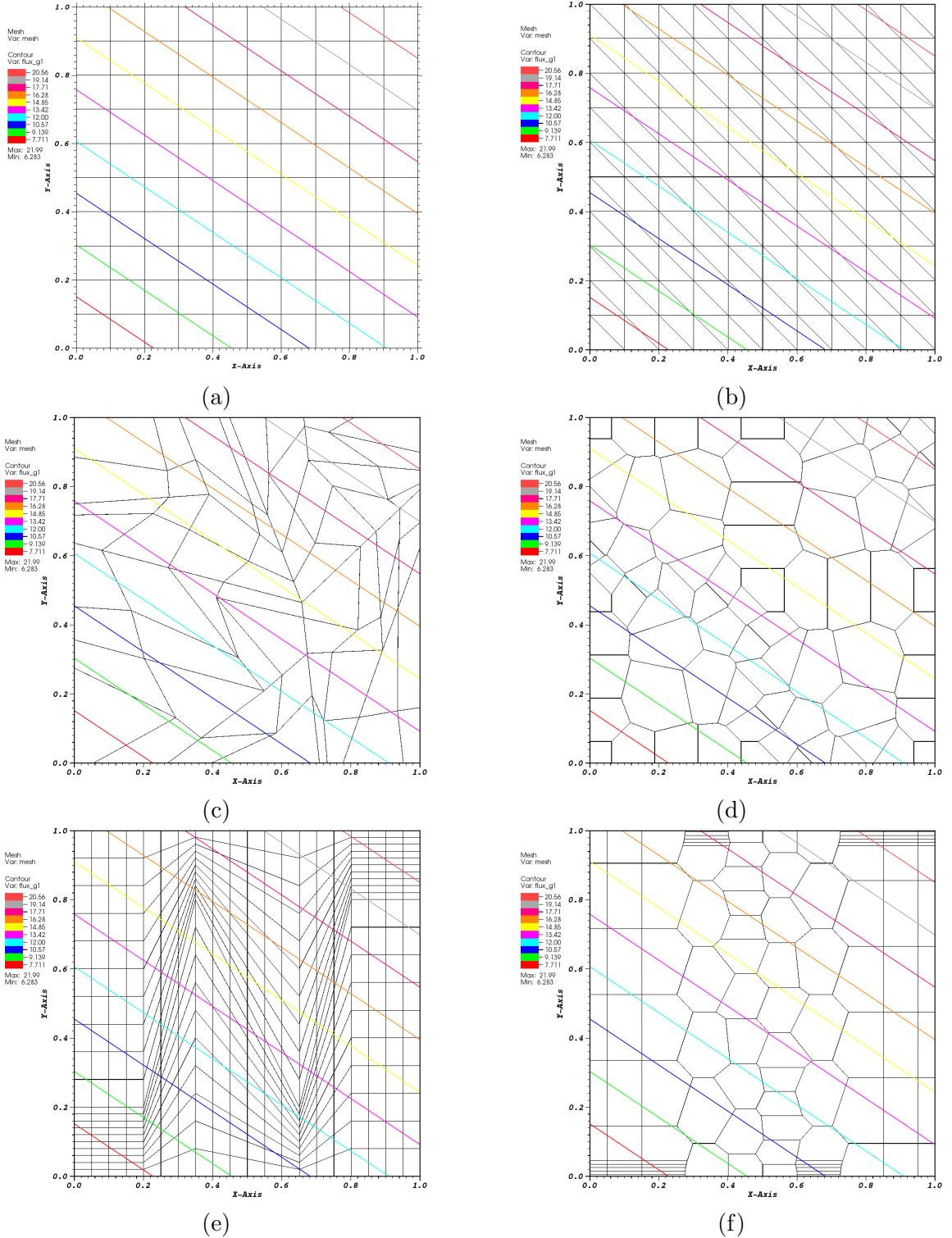


Figure 3.9: Contour plots of the exactly-linear solution with the mean value basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

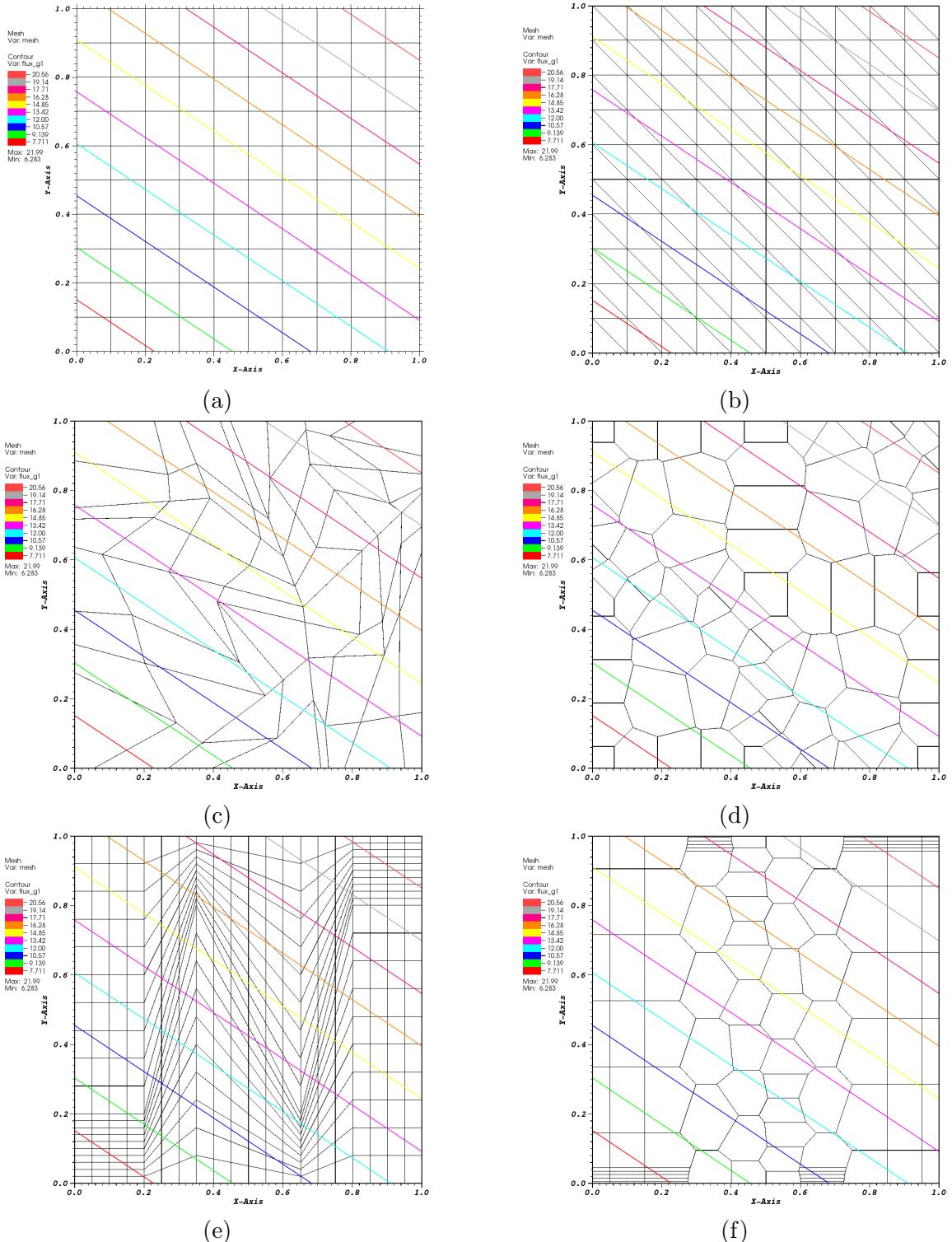


Figure 3.10: Contour plots of the exactly-linear solution with the linear maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

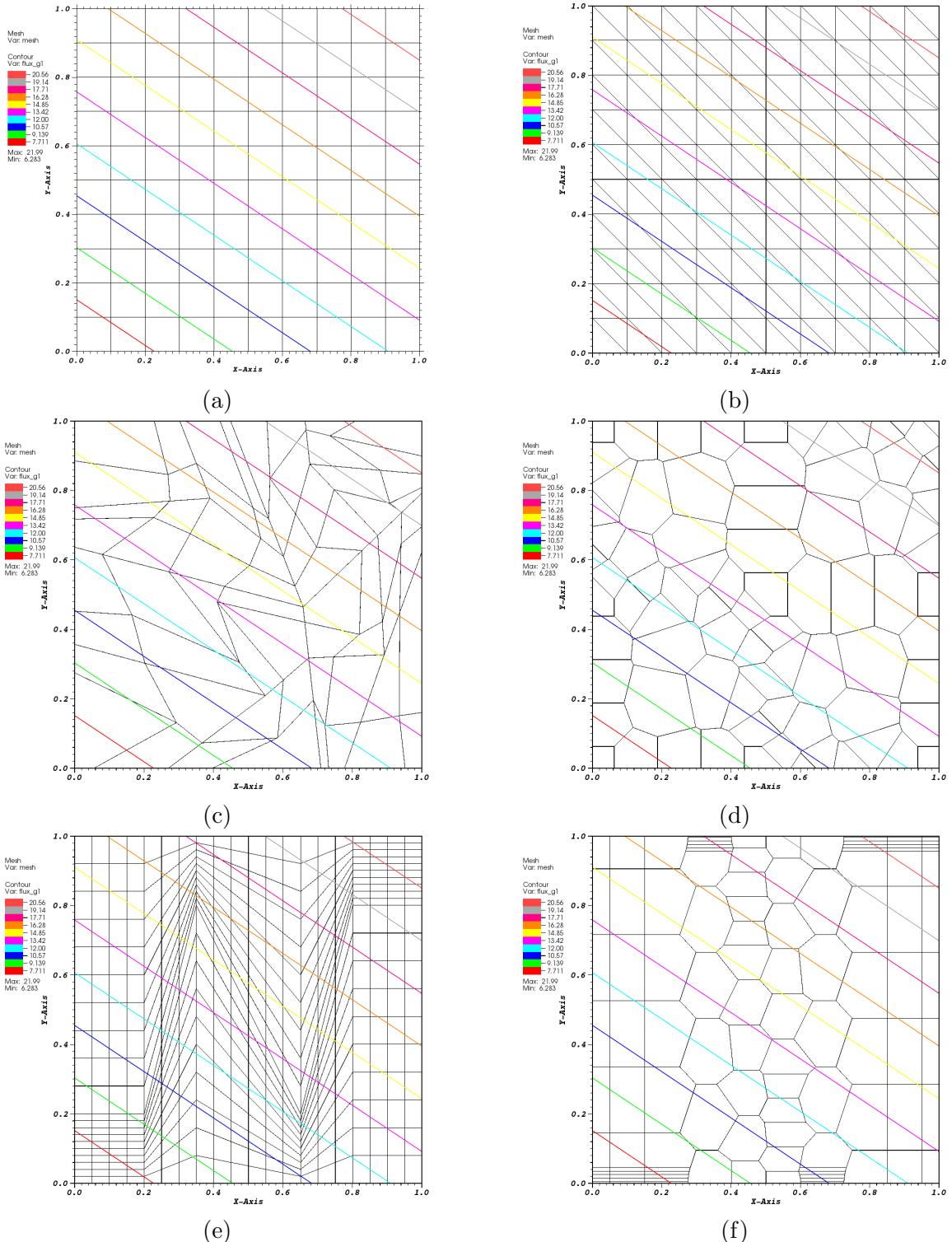


Figure 3.11: Contour plots of the exactly-linear solution with the quadratic serendipity maximum entropy basis functions on (a) cartesian mesh, (b) ordered-triangular mesh, (c) quadrilateral shestakov mesh, (d) sinusoidal polygonal mesh, (e) quadrilateral z-mesh, and (f) polygonal z-mesh.

grid resolution can be related to the number of spatial degrees of freedom (N_{dof}),

$$N_{dof} \propto h^{-d}, \quad (3.22)$$

where d is the dimensionality of the problem. Using this relationship for the grid resolution, along with the error norm of Eq. (3.21), we can define a general relationship between the error norm and the number of spatial degrees of freedom:

$$\|\Phi_e - \Phi_h\|_2 \propto N_{dof}^{-\frac{k+1}{d}}. \quad (3.23)$$

The result of Eq. (3.23) 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

$$\begin{aligned} \Psi(x, y) &= C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right) \\ \Phi(x, y) &= 2\pi C_M x(L_x - x)y(L_y - y) \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{\gamma}\right) \end{aligned} \quad (3.24)$$

where the constants in the equations are:

$$C_M = \frac{200}{L_x^2 L_y^2} \quad \gamma = \frac{L_x L_y}{100}. \quad (3.25)$$

3.4.3 Searchlight Problem

The next example models a beam or searchlight. Similar problems were investigated in Dedner and Vollmöller [17] and Wang and Ragusa [18]. In this problem, an incident beam of neutrons is shined onto a small portion of a boundary, propagates

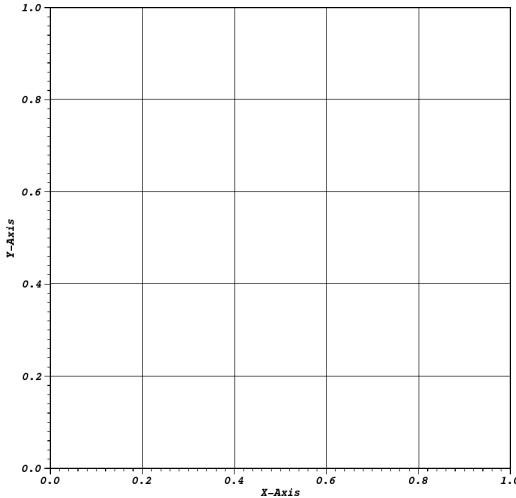


Figure 3.12: Initial mesh configuration for the searchlight problem before any refinement cycles.

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

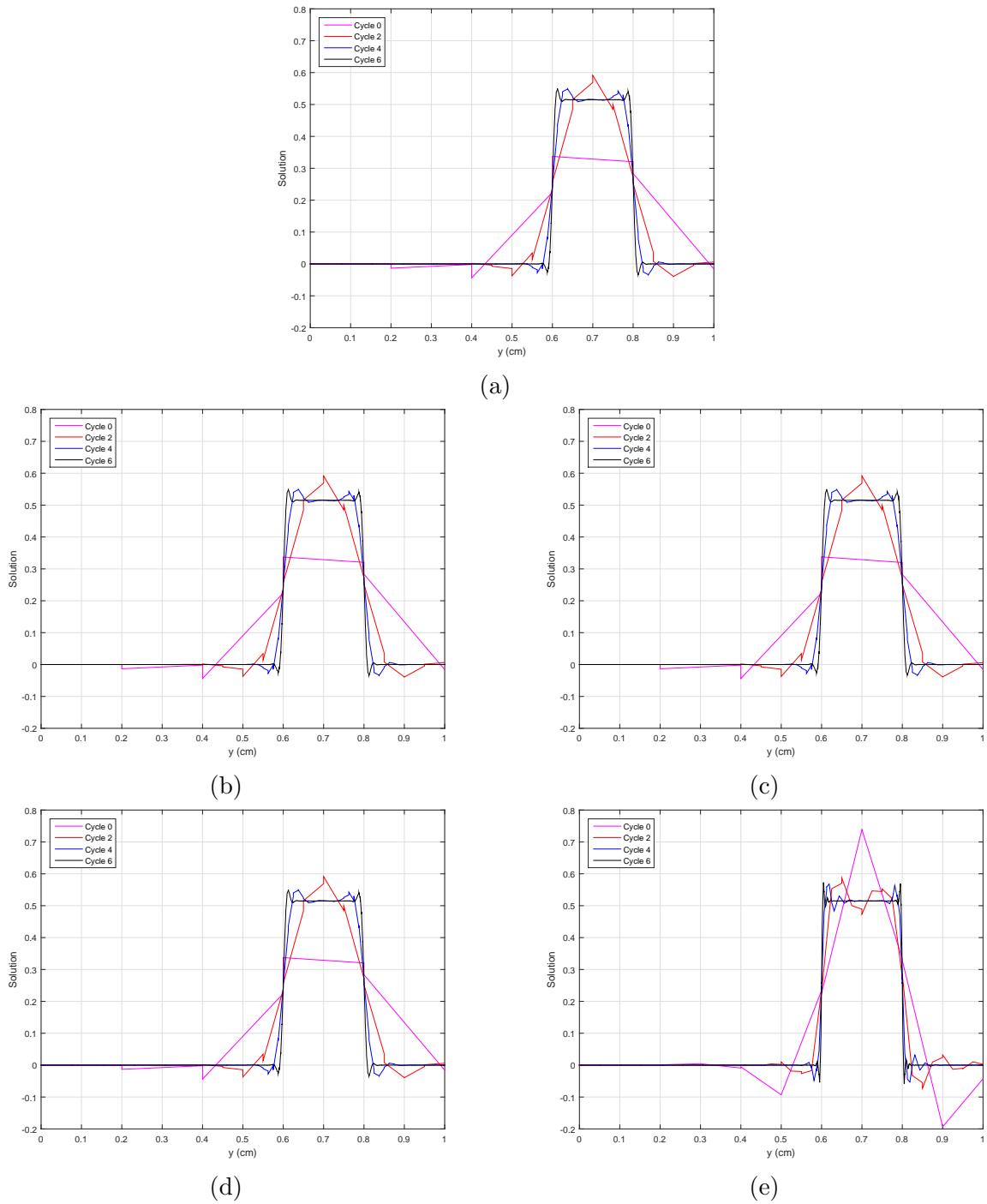


Figure 3.13: Exiting angular flux on the right boundary with uniform refinement using the (a) Wachspress basis functions, (b) PWL basis functions, (c) mean value basis functions, (d) linear maximum entropy coordinates and (e) quadratic serendipity maximum entropy coordinates.

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

4.1 Introduction

4.1.1 *History*

4.1.2 *Theory*

$$\mathbf{L}\Psi = \mathbf{M}\Sigma\Phi + \mathbf{Q} \quad (4.1)$$

$$\mathbf{L}\Psi^{(\ell+1)} = \mathbf{M}\Sigma\Phi^{(\ell)} + \mathbf{Q} \quad (4.2)$$

$$\mathbf{L}(\Psi^{(\ell+1)} - \Psi) = \mathbf{M}\Sigma(\Phi^{(\ell)} - \Phi) \quad (4.3)$$

$$\delta\Psi^{(\ell+1)} \equiv (\Psi^{(\ell+1)} - \Psi) \quad \text{and} \quad \delta\Phi^{(\ell+1)} = \mathbf{D}\delta\Psi^{(\ell+1)} \quad (4.4)$$

$$-\vec{\nabla} \cdot D\vec{\nabla}\Phi + \sigma\Phi = Q, \quad \vec{r} \in \mathcal{D} \quad (4.5)$$

$$\left\langle D\vec{\nabla}\Phi, \vec{\nabla}\Phi^* \right\rangle_{\mathcal{D}} + \left\langle \sigma\Phi, \Phi^* \right\rangle_{\mathcal{D}} = \left\langle Q, \Phi^* \right\rangle_{\mathcal{D}} + \left\{ D\partial_n\Phi, \Phi^* \right\}_{\partial\mathcal{D}} \quad (4.6)$$

4.2 Symmetric Interior Penalty Form of the Diffusion Equation

$$\Phi(\vec{r}) + \frac{1}{\kappa} D\partial_n\Phi(\vec{r}) = \Phi_0(\vec{r}), \quad \vec{r} \in \partial\mathcal{D}^d, \quad \kappa \gg 1 \quad (4.7)$$

$$\begin{aligned} & \left\langle D\vec{\nabla}\Phi, \vec{\nabla}\Phi^* \right\rangle_{\mathcal{D}} - \left\{ D\partial_n\Phi, \Phi^* \right\}_{\partial\mathcal{D}^d} - \left\{ \Phi, D\partial_n\Phi^* \right\}_{\partial\mathcal{D}^d} \\ & + \left\{ \kappa(\Phi - \Phi_0), \Phi^* \right\}_{\partial\mathcal{D}^d} = - \left\{ \Phi_0, D\partial_n\Phi^* \right\}_{\partial\mathcal{D}^d} \end{aligned} \quad (4.8)$$

$$\left\{ \kappa[\Phi], [\Phi^*] \right\}_{E_h^i} + \left\{ [\Phi], \{D\partial_n\Phi^*\} \right\}_{E_h^i} + \left\{ \{D\partial_n\Phi\}, [\Phi^*] \right\}_{E_h^i} = 0 \quad (4.9)$$

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

$$\{\{\Phi\}\} \equiv \frac{\Phi^+ + \Phi^-}{2} \quad \text{and} \quad [\Phi] \equiv \Phi^+ - \Phi^-, \quad (4.10)$$

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

$$\Phi^\pm \equiv \lim_{s \rightarrow 0^\pm} \Phi(\vec{r} + s\vec{n}), \quad (4.11)$$

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

Using Eqs. (4.8) and (4.9), the SIP form of the diffusion equation can be succinctly written as

$$a^{SIP}(\Phi, \Phi^*) = b^{SIP}(\Phi^*), \quad (4.12)$$

with the following bilinear matrix:

$$\begin{aligned} a^{SIP}(\Phi, \Phi^*) &= \left\langle D\vec{\nabla}\Phi, \vec{\nabla}\Phi^* \right\rangle_{\mathcal{D}} + \left\langle \sigma\Phi, \Phi^* \right\rangle_{\mathcal{D}} + \frac{1}{2} \left\{ \Phi, \Phi^* \right\}_{\partial\mathcal{D}^r} \\ & + \left\{ \kappa_e^{SIP}[\Phi], [\Phi^*] \right\}_{E_h^i} + \left\{ [\Phi], \{D\partial_n\Phi^*\} \right\}_{E_h^i} + \left\{ \{D\partial_n\Phi\}, [\Phi^*] \right\}_{E_h^i}, \\ & + \left\{ \kappa_e^{SIP}\Phi, \Phi^* \right\}_{\partial\mathcal{D}^d} - \left\{ \Phi, D\partial_n\Phi^* \right\}_{\partial\mathcal{D}^d} - \left\{ D\partial_n\Phi, \Phi^* \right\}_{\partial\mathcal{D}^d} \end{aligned} \quad (4.13)$$

and with the following linear right-hand-side:

$$b^{SIP}(\Phi^*) = \left\langle Q, \Phi^* \right\rangle_{\mathcal{D}} - \left\{ J_0, \Phi^* \right\}_{\partial\mathcal{D}^n} + 2 \left\{ J^{inc}, \Phi^* \right\}_{\partial\mathcal{D}^r} + \left\{ \kappa_e^{SIP} \Phi_0, \Phi^* \right\}_{\partial\mathcal{D}^d} - \left\{ \Phi_0, D\partial_n \Phi^* \right\}_{\partial\mathcal{D}^d}. \quad (4.14)$$

As previously stated, the general penalty term, κ of Eqs. (4.7 - 4.9) needs to have sufficient positive measure to ensure stability. From previous investigations [?, 19, 20], we choose the penalty coefficient to be face dependent:

$$\kappa_e^{SIP} = \begin{cases} \frac{c}{2} \left(\frac{D^+}{h^+} + \frac{D^-}{h^-} \right) & e \in E_h^i \\ c \frac{D^-}{h^-} & e \in \partial\mathcal{D} \end{cases}, \quad (4.15)$$

for interior, E_h^i , and boundary, $\partial\mathcal{D}$, faces respectively. In Eq. (4.15), h^\pm is the orthogonal projection of the face, e , into the cells defined by the trace in Eq. (4.11). Turcksin and Ragusa, [21], defined h^\pm for 2D polygons, whose definitions can be seen in Table 4.1. The orthogonal projection for both triangles and quadrilaterals can be explicitly defined from simple geometric relationships. However, for polygons with > 4 faces, there is no explicit geometric relationship to define the orthogonal projection. Instead, the polygon is approximated as regular, and the orthogonal projection is no longer face-dependent. For polygons with an even number of faces greater than 4, the orthogonal projection is twice the apothem, which is the line segment between the polygon's center and the midpoint of each polygon's side. For odd number of faces greater than 4, the polygon's orthogonal projection becomes the sum of the apothem and the circumradius.

In a similar manner to the 2D orthogonal projections defined in Table 4.1, we define our choice for the extension of the orthogonal projections to 3D in Table 4.2. Like triangles and quadrilaterals in 2D, the orthogonal projections for tetrahedra

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

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

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

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

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

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

$$a^{MIP}(\delta\Phi, \Phi^*) = b^{MIP}(\Phi^*), \quad (4.16)$$

with the following bilinear matrix:

$$\begin{aligned} a^{MIP}(\delta\Phi, \Phi^*) &= \left\langle D\vec{\nabla}\delta\Phi, \vec{\nabla}\Phi^* \right\rangle_{\mathcal{D}} + \left\langle \sigma\delta\Phi, \Phi^* \right\rangle_{\mathcal{D}} \\ &+ \left\{ \kappa_e^{MIP} [\delta\Phi], [\Phi^*] \right\}_{E_h^i} + \left\{ [\delta\Phi], \{D\partial_n\Phi^*\} \right\}_{E_h^i} + \left\{ \{D\partial_n\delta\Phi\}, [\Phi^*] \right\}_{E_h^i}, \quad (4.17) \\ &+ \left\{ \kappa_e^{MIP} \delta\Phi, \Phi^* \right\}_{\partial\mathcal{D}^d} - \frac{1}{2} \left\{ \delta\Phi, D\partial_n\Phi^* \right\}_{\partial\mathcal{D}^d} - \frac{1}{2} \left\{ D\partial_n\delta\Phi, \Phi^* \right\}_{\partial\mathcal{D}^d} \end{aligned}$$

and with the following linear right-hand-side:

$$b^{MIP}(\Phi^*) = \left\langle Q, \Phi^* \right\rangle_{\mathcal{D}} + \left\{ \delta J^{inc}, \Phi^* \right\}_{\partial\mathcal{D}^r}. \quad (4.18)$$

$$\kappa_e^{MIP} = \max \left(\kappa_e^{SIP}, \frac{1}{4} \right) \quad (4.19)$$

4.4 Multigroup Applications of Diffusion Synthetic Acceleration

$$\mathbf{L}_g \Psi_g = \mathbf{M} \sum_{g'=1}^G \mathbf{S}_{gg'} \Phi_{g'} + Q_g, \quad \Phi_g \equiv \mathbf{D}\Psi_g \quad (4.20)$$

$$\delta\Psi_g^{(\ell+1)} \equiv \Psi_g - \Psi_g^{(\ell+1)} \quad \text{and} \quad \delta\Phi_g^{(\ell+1)} \equiv \mathbf{D}\delta\Psi_g^{(\ell+1)}, \quad (4.21)$$

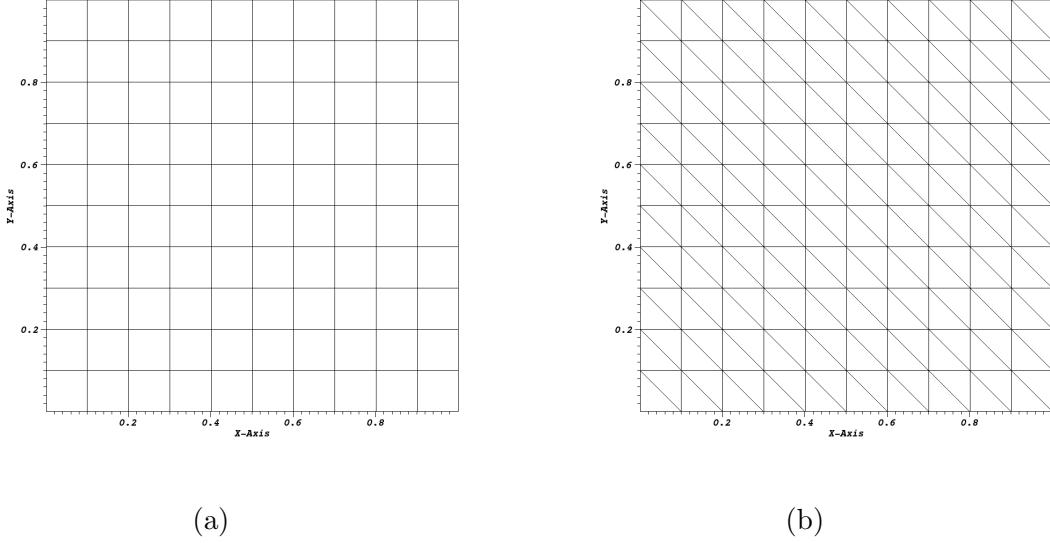


Figure 4.1: blah

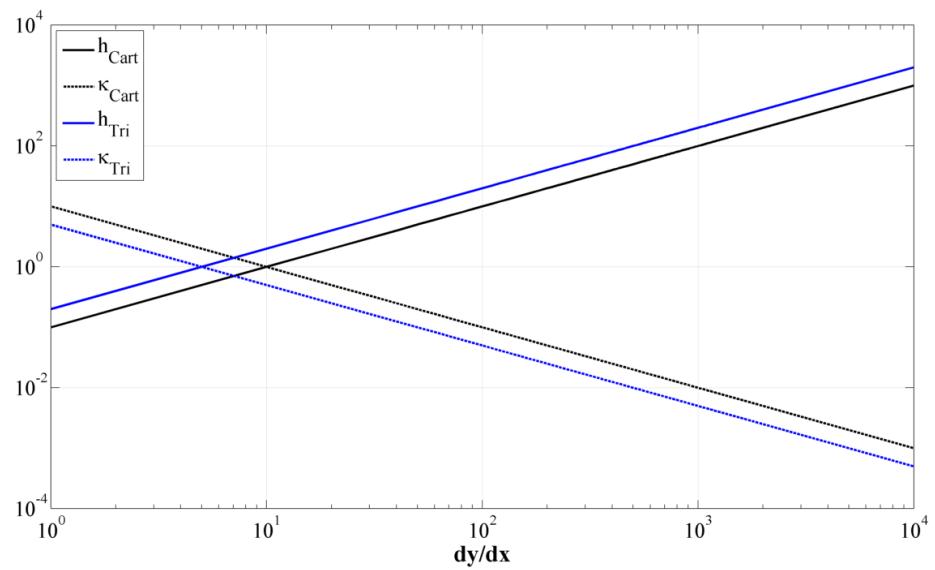


Figure 4.2: Maximum SIP orthogonal projections (solid) and penalty coefficients (dashed) on ordered cartesian (black) and triangular (blue) grids.

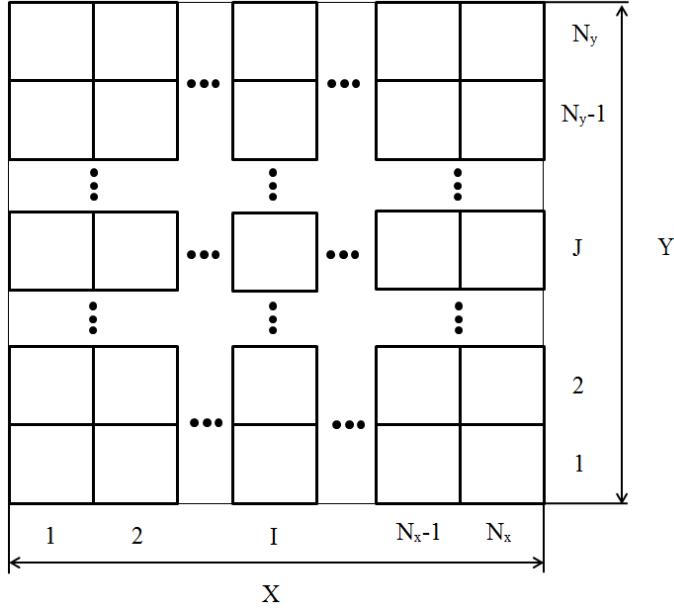


Figure 4.3: Fourier domain for 2D quadrilateral cells or an axial slice of 3D hexahedral cells in a regular grid.

4.5 Fourier Analysis

$$\begin{aligned}\Psi_m(\vec{r}) &= \hat{\Psi}_m e^{i\vec{\lambda} \cdot \vec{r}} \\ \Phi(\vec{r}) &= \hat{\Phi} e^{i\vec{\lambda} \cdot \vec{r}}\end{aligned}\tag{4.22}$$

where $i = \sqrt{-1}$

4.6 Numerical Results

We now present the results showing the efficacy of a discontinuous diffusion form which can be used as a DSA preconditioner for the transport equation discretized over arbitrary grids. We first present the SIP form and how it performs as a pure diffusion solver on unstructured polyhedral grids in Section 4.6.2. We next present the theoretical performance limits of the MIP form as a DSA preconditioner using Fourier Analysis in Section 4.6.3. We finish by presenting the results of MIP's use

as a DSA preconditioner for several different problem types in Section 4.6.4.

4.6.1 Transport Solutions in the Thick Diffusive Limit

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

$$\begin{aligned}\sigma_t &\rightarrow \frac{\sigma_t}{\epsilon} \\ \sigma_a &\rightarrow \epsilon\sigma_t \\ \frac{Q_0}{4\pi} &\rightarrow \epsilon\frac{Q_0}{4\pi}\end{aligned}\tag{4.23}$$

$$\vec{\Omega} \cdot \vec{\nabla} \Psi + \frac{\sigma_t}{\epsilon} \Psi = \sigma_t \left(\frac{1}{\epsilon} - \epsilon \right) \frac{\Phi}{4\pi} + \epsilon \frac{Q_0}{4\pi}\tag{4.24}$$

$$\epsilon \vec{\nabla} \cdot \frac{1}{3\sigma_t} \vec{\nabla} \Phi + \epsilon \sigma_t \Phi = \epsilon Q_0\tag{4.25}$$

$$\begin{aligned}\vec{\Omega} \cdot \vec{\nabla} \Psi + \frac{1}{\epsilon} \Psi &= \left(\frac{1}{\epsilon} - \epsilon \right) \frac{\Phi}{4\pi} + \frac{\epsilon}{4\pi} \\ \frac{\epsilon}{3} \nabla^2 \Phi + \epsilon \Phi &= \epsilon\end{aligned}\tag{4.26}$$

4.6.2 SIP used as a Diffusion Solver

We first wish to know how an interior penalty form of the diffusion equation will perform on unstructured polyhedral grids. We will perform this analysis with a stand-alone MATLAB code. The polyhedral mesh types for this analysis are presented in Section 4.6.2.1.

1. A purely-linear solution to determine if linear basis functions will span the solution space;
2. The Method of Manufactured Solutions to test basis function convergence rates;

in Sections 4.6.2.2 and 4.6.2.3, respectively.

4.6.2.1 Geometry Specification for the SIP Problems

For this work in

4.6.2.2 Purely-Linear Solution

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

$$\Phi(x, y, z) = \frac{4J^{inc}}{L + 4D} (L + 2D - y), \quad (4.27)$$

with the following boundary conditions in the y-direction:

$$\begin{aligned} \Phi - 2D\partial_y\Phi &= 4J^{inc}, & \forall(x, z), y = 0 \\ \Phi + 2D\partial_y\Phi &= 0, & \forall(x, z), y = L \end{aligned} \quad . \quad (4.28)$$

4.6.2.3 Method of Manufactured Solutions

$$\Phi^{quad}(x, y, z) = x(1 - x)y(1 - y)z(1 - z) \quad (4.29)$$

$$\Phi^{gauss}(x, y, z) = \Phi^{quad}(x, y, z) \exp(-(\vec{r} - \vec{r}_0) \cdot (\vec{r} - \vec{r}_0)^T) \quad (4.30)$$

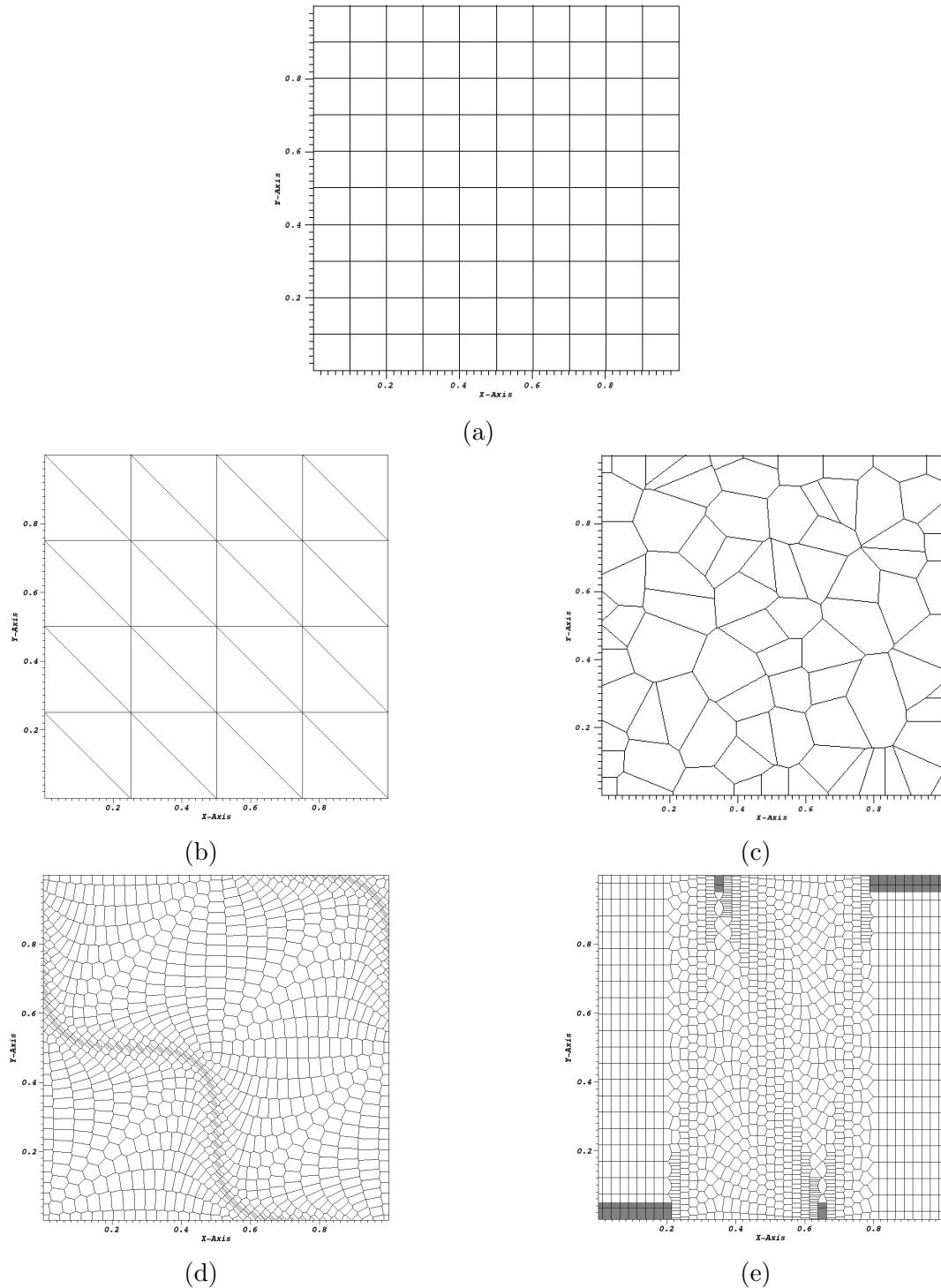
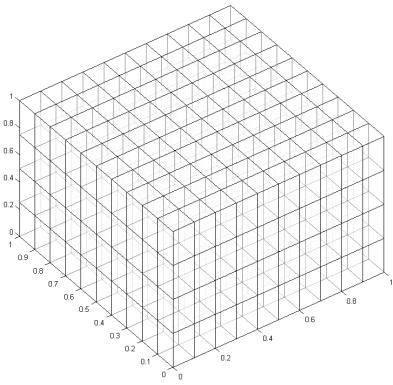
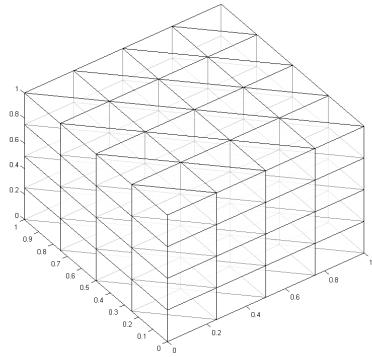


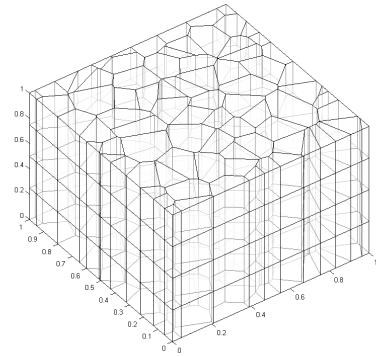
Figure 4.4: Axial slices of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.



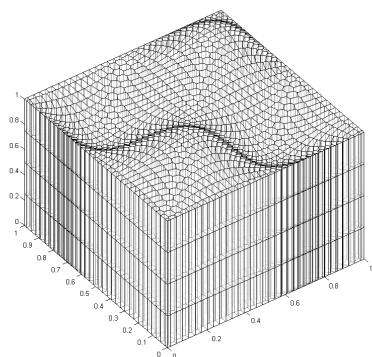
(a)



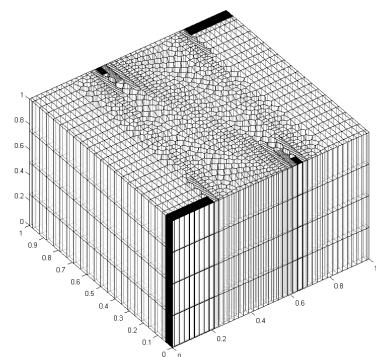
(b)



(c)



(d)



(e)

Figure 4.5: Extrusion of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

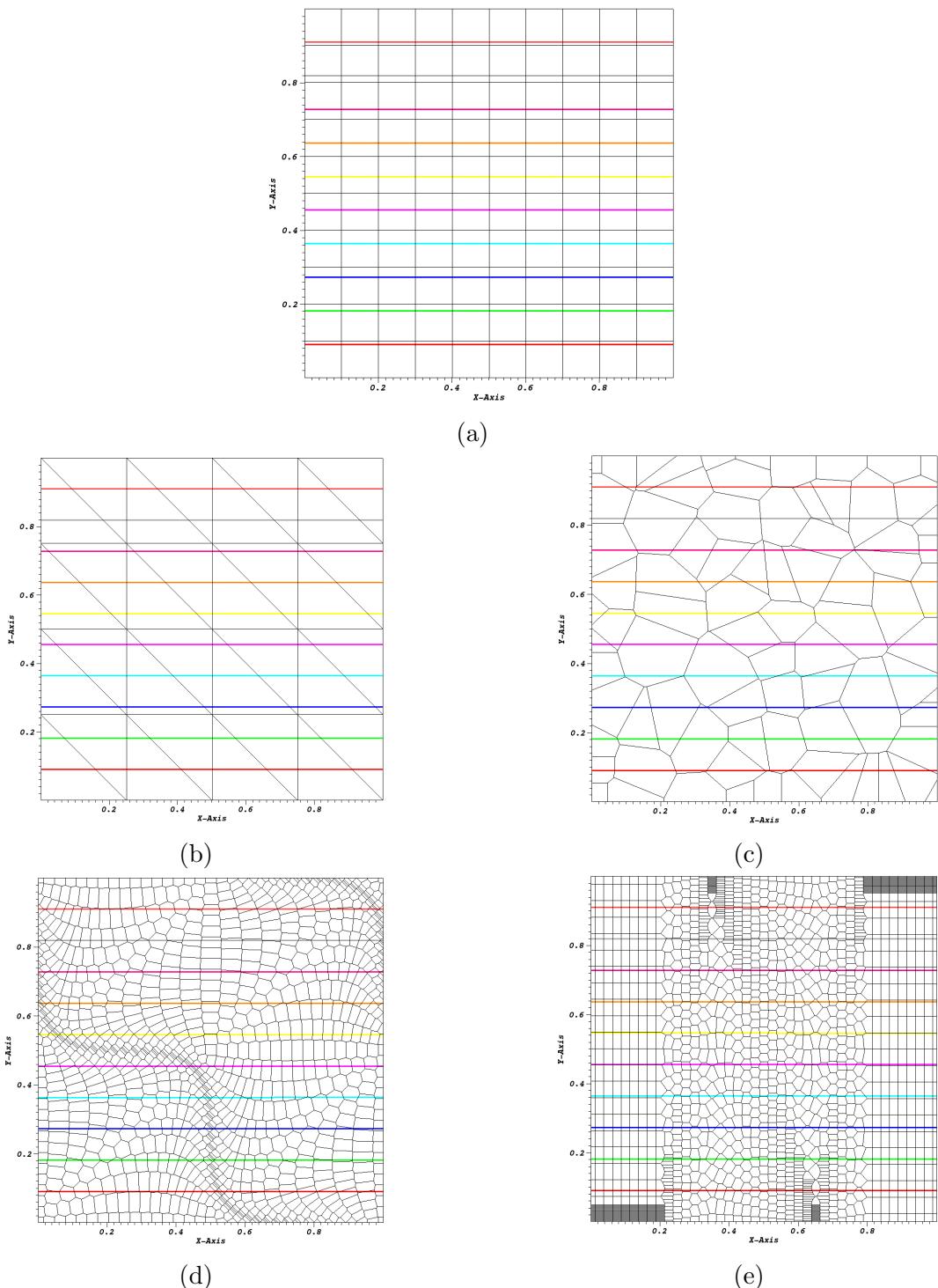


Figure 4.6: Axial slice showing the contours for the linear solution of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

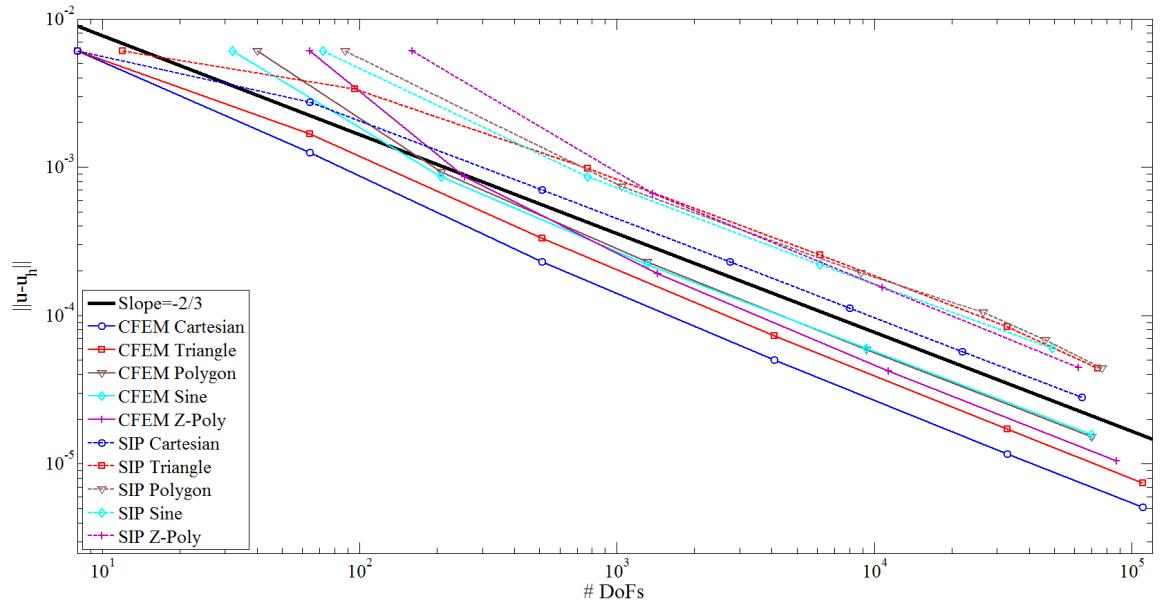


Figure 4.7: blah

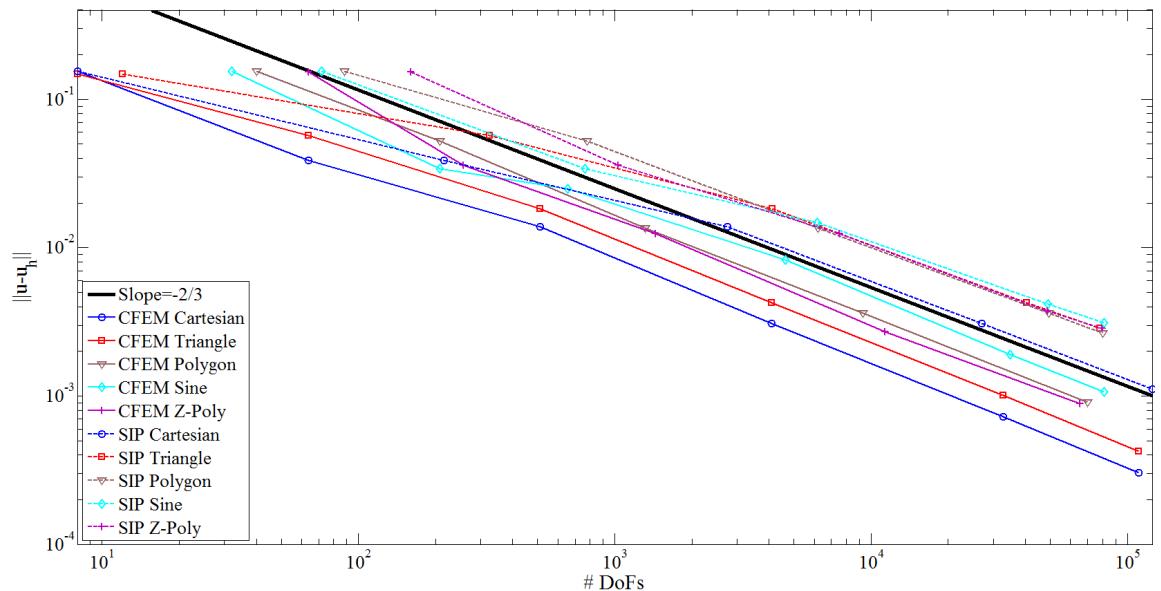


Figure 4.8: blah

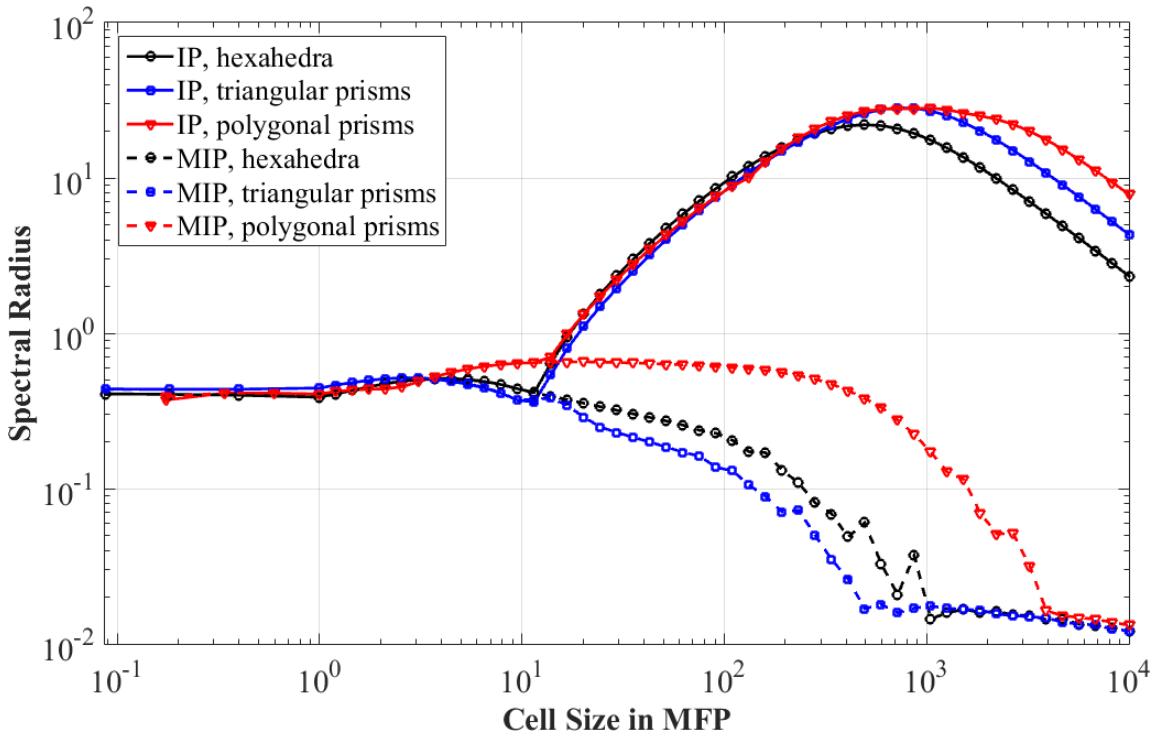


Figure 4.9: blah

4.6.3 Fourier Analysis

4.6.3.1 Homogeneous Medium Case

4.6.3.2 Periodic Horizontal Interface (PHI) Problem

4.6.4 MIP Results

We have given a theoretical basis for MIP's stability using Fourier Analysis in Section 4.6.3. In this Section, we provide numerical results of DSA preconditioning of the Discontinuous S_N Transport Equation. We first provide

4.6.4.1 Simple Homogeneous Problem Results

4.6.4.2 Heterogenous Problem Results

4.7 Conclusions

In this chapter, we have presented a discontinuous form of the diffusion equation, known as the Modified Interior Penalty Method (MIP), which can be applied to the discontinuous finite element form of the transport equation as a DSA preconditioner.

5. CONCLUSIONS AND OPEN ITEMS

5.1 Conclusions

In this dissertation

5.2 Open Items

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. DUDEKSTADT 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. DUDEKSTADT 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] A. ERN and J.-L. GUERMOND, *Theory and practice of finite elements*, vol. 159, Springer Science & Business Media (2013).
- [8] 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).
- [9] O. ZEINKIEWICZ, R. TAYLOR, and J. ZHU, *The finite element method: its basis and fundamentals*, SEElsevier Butterworth-Heinemann (2005).
- [10] J. AKIN, *Application and implementation of finite element methods*, Academic Press, Inc. (1982).

- [11] 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).
- [12] 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).
- [13] A. SHESTAKOV, D. KERSHAW, and G. ZIMMERMAN, “Test problems in radiative transfer calculations,” *Nuclear science and engineering*, **105**, 1, 88–104 (1990).
- [14] D. S. KERSHAW, “Differencing of the diffusion equation in Lagrangian hydrodynamic codes,” *Journal of Computational Physics*, **39**, 2, 375–395 (1981).
- [15] 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).
- [16] 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).
- [17] 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).
- [18] 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).

- [19] Y. WANG, *Adaptive mesh refinement solution techniques for the multigroup SN transport equation using a higher-order discontinuous finite element method*, Ph.D. thesis, Texas A&M University (2009).
- [20] Y. WANG and J. RAGUSA, “Diffusion Synthetic Acceleration for High-Order Discontinuous Finite Element S_n Transport Schemes and Application to Locally Refined Unstructured Meshes,” *Nuclear Science and Engineering*, **166**, 145–166 (2010).
- [21] B. TURCKSIN and J. C. RAGUSA, “Discontinuous diffusion synthetic acceleration for Sn transport on 2D arbitrary polygonal meshes,” *Journal of Computational Physics*, **274**, 356–369 (2014).