AN ADAPTIVE ANGULAR DISCRETIZATION METHOD FOR NEUTRAL-PARTICLE TRANSPORT IN THREE-DIMENSIONAL GEOMETRIES

A Dissertation

by

JOSHUA JOHN JARRELL

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

DOCTOR OF PHILOSOPHY

December 2010

Major Subject: Nuclear Engineering

AN ADAPTIVE ANGULAR DISCRETIZATION METHOD FOR NEUTRAL-PARTICLE TRANSPORT IN THREE-DIMENSIONAL GEOMETRIES

A Dissertation

by

JOSHUA JOHN JARRELL

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

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Marvin L. Adams
Committee Members,	Robert E. Grove
	Raytcho Lazarov
	Jim E. Morel
	Jean C. Ragusa
	Aaron M. Watson
Head of Department,	Raymond J. Juzaitis

December 2010

Major Subject: Nuclear Engineering

ABSTRACT

An Adaptive Angular Discretization Method for Neutral-Particle Transport in
Three-Dimensional Geometries. (December 2010)
Joshua John Jarrell, B.S., Texas A&M University
Chair of Advisory Committee: Dr. Marvin L. Adams

In this dissertation, we discuss an adaptive angular discretization scheme for the neutral-particle transport equation in three dimensions. We mesh the direction domain by dividing the faces of a regular octahedron into equilateral triangles and projecting these onto "spherical triangles" on the surface of the sphere. We choose four quadrature points per triangle, and we define interpolating basis functions that are linear in the direction cosines. The quadrature point's weight is the integral of the point's linear discontinuous finite element (LDFE) basis function over its local triangle. Variations in the locations of the four points produce variations in the quadrature set.

The new quadrature sets are amenable to local refinement and coarsening, and hence can be used with an adaptive algorithm. If local refinement is requested, we use the LDFE basis functions to build an approximate angular flux, $\Psi_{\text{interpolated}}$, by interpolation through the existing four points on a given triangle. We use a transport sweep to find the actual values, Ψ_{calc}, at certain test directions in the triangle and compare against $\Psi_{\text{interpolated}}$ at those directions. If the results are not within a user-defined tolerance, the test directions are added to the quadrature set.

The performance of our uniform sets (no local refinement) is dramatically better than that of commonly used sets (level-symmetric (LS), Gauss-Chebyshev (GC) and variants) and comparable to that of the Abu-Shumays Quadruple Range (QR) sets. On simple problems, the QR sets and the new sets exhibit 4th-order convergence in the scalar flux as the directional mesh is refined, whereas the LS and GC sets exhibit 1.5-order and 2nd-order convergence, respectively. On difficult problems (near discontinuities in the direction domain along directions that are not perpendicular to coordinate axes), these convergence orders diminish and the new sets outperform the others. We remark that the new LDFE sets have strictly positive weights and that arbitrarily refined sets can be generated without the numerical difficulties that plague the generation of high-order QR sets.

Adapted LDFE sets are more efficient than uniform LDFE sets only in difficult problems. This is due partly to the high accuracy of the uniform sets, partly to basing refinement decisions on purely local information, and partly to the difficulty of mapping among differently refined sets. These results are promising and suggest interesting future work that could lead to more accurate solutions, lower memory requirements, and faster solutions for many transport problems.

To my loving wife, Heather

ACKNOWLEDGMENTS

I would like to thank my graduate advisor and committee chair, Dr. Marvin Adams. Without his guidance, patience, and ideas, this research would have never happened. I would also like to thank my committee members: Dr. Bob Grove, Dr. Raytcho Lazarov, Dr. Jim Morel, Dr. Jean Ragusa, and Dr. Aaron Watson. Their ideas have greatly improved this research.

I would also like to specifically thank Dr. Aaron Watson and Mike Shearer at Knolls Atomic Power Laboratory. They spent hours teaching best coding practices for c++, answering incredibly specific questions, and patiently listening to my ideas (and then telling me how I should truly do things). Without Aaron's patience and guidance, this research would have taken many years longer. My thanks to Joel Risner at Bettis Atomic Power Laboratory for explaining many details of shielding calculations and data management.

This research was performed under appointment to the Rickover Graduate Fellowship Program sponsored by Naval Reactors Division of the U.S. Department of Energy.

At Texas A&M University, I have been helped and mentored by many other graduate students. I want to thank Alex Maslowski, Teresa Bailey, Jae Chang, and Timmie Smith for answering numerous questions for this research as well as many other topics. Without Jae's and Timmie's help, I would have never been able to complete this research in a timely manner. I would also like to thank Anthony Barbu, Kevin Hogan, Tom Lewis, Vijay Mahadevan, Tara Pandya, and Yaqi Wang for making my graduate school worthwhile.

Most importantly, I thank my family for loving and supporting me. My parents, siblings, parents-in-law, and sisters-in-law have helped in numerous ways. Thanks to Kenlee and Heather for encouraging me to complete this degree as quickly as possible.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

In this dissertation we introduce new quadrature sets for solving the discrete ordinate (S_n) neutral-particle transport equation. Our new quadrature sets are based on discontinuous finite-element basis functions in the direction (or "angle") variable. They are amenable to local refinement and coarsening and thus can form the foundation of adaptive discrete-ordinates algorithms. We develop and test such algorithms in this work, including refinement and coarsening strategies as well as algorithms for mapping between different quadrature sets. In this section we describe the transport equation that we are attempting to solve and briefly describe previous work that has laid the foundation for our work.

1.1 Neutral Particle Transport Equation

The neutral-particle transport equation is the linear form of the Boltzmann equation, which is a conservation equation in a seven-dimensional "phase-space." The solution, called the angular flux, is dependent on seven variables: time, energy, three position variables, and two direction (or "angular") variables. For neutrons, the transport equation is

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)}{\partial t}+\overrightarrow{\Omega}\cdot\overrightarrow{\nabla}\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)+\sigma_t(\overrightarrow{r},E,t)\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t) = \\ \frac{\chi(E)}{4\pi}\int_0^\infty dE'\int_{4\pi}d\Omega\nu(E')\sigma_f(\overrightarrow{r},E',t)\Psi(\overrightarrow{r},\overrightarrow{\Omega},E',t)+ \\ \int_{4\pi}d\Omega'\int_0^\infty dE'\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega},E'\to E,t)\Psi(\overrightarrow{r},\overrightarrow{\Omega}',E',t)+ \\ q(\overrightarrow{r},\overrightarrow{\Omega},E,t)\ ,\end{aligned} \tag{1.1}$$

This dissertation follows the style of International Journal of Heat and Mass Transfer.

where

$$
\begin{aligned}
\overrightarrow{r} =& \text{space coordinate vector } (cm),\\
\overrightarrow{\Omega} =& \text{unit vector in the direction}\\
& \text{of the particle motion,}\\
E =& \text{particle's energy } (MeV),\\
t =& \text{time } (s),\\
v(E) =& \text{particle's speed corresponding}\\
& \text{to energy } E\ (\frac{cm}{s}),\\
\chi(E) =& \text{fraction of fission particles}\\
& \text{emitted per unit energy}\\
& \text{centered about } E\ (\frac{1}{MeV}),\\
\nu(E) =& \text{average number of particles}\\
& \text{emitted from fission caused by}\\
& \text{neutrons of energy } E\ (\frac{neutrons}{fission}),\\
\sigma_t(\overrightarrow{r},\overrightarrow{\Omega},E,t) =& \text{total cross-section } (cm^{-1}),\\
\sigma_f(\overrightarrow{r},E,t) =& \text{fission cross-section } (cm^{-1}),\\
\sigma_s(\overrightarrow{r},\overrightarrow{\Omega'}\cdot\overrightarrow{\Omega},E'\rightarrow E,t) =& \text{scattering cross-section}\\
& \ \text{from } (E',\Omega') \text{ into } (E,\Omega)\ (cm^{-1}MeV^{-1}ster^{-1}),\\
\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t) =& \text{particle angular flux}\\
& (\frac{particles}{MeV\,ster\,cm^2\,s}),\\
q(\overrightarrow{r},\overrightarrow{\Omega},E,t) =& \text{particle source rate density}\\
& (\frac{particles}{MeV\,ster\,cm^3\,s})\ .
\end{aligned}
\tag{1.2}
$$

When we write the time-dependent equation in this form, we implicitly assume that the delayed neutron emission is included in the $q(\overrightarrow{r},\overrightarrow{\Omega},E,t)$ term and the fission term is just the prompt neutrons. Also, we assume that the scattering cross section has been modified to include $(n, 2n)$ and similar reactions. Assuming the boundaries are non-reentrant, the boundary conditions are given for all incoming angles:

$$\Psi(\overrightarrow{r}_s, \overrightarrow{\Omega}, E, t) = F(\overrightarrow{r}_s, \overrightarrow{\Omega}, E, t) \quad \overrightarrow{n} \cdot \overrightarrow{\Omega} < 0 \quad on \quad \delta D, \tag{1.3}$$

where F is a known function, $\overrightarrow{n}$ is the outward facing normal on the boundary δD, and $\overrightarrow{r}_s$ is a point on δD. Boundary conditions that specify the incident angular flux in terms of the exiting angular flux are also possible. Examples include specular reflection (mirror) or periodic boundary conditions. We assume an initial condition is given as

$$\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E, t_0) = F_0(\overrightarrow{r}, \overrightarrow{\Omega}, E), \tag{1.4}$$

where F_0 is a known function and t_0 is the time at the start of the problem.

1.1.1 Transport Equation Assumptions

Equation 1.1 has a number of assumptions or approximations built into it [1]. The first is that particles may be considered as points. That is, we assume the quantum mechanical wavelength is small compared to the atomic diameter. This assumption is used because particles, in general, travel many atomic diameters between collisions. The next assumption is that particles are not affected by gravitational, electric, or magnetic fields. This ensures that particles travel in straight lines. Because we are utilizing the linear form of the transport equation, we assume particles do not interact with each other. This assumption is generally valid because the number of particles is dramatically (orders of magnitude) less than the number of particles in the surrounding matter. Another assumption is that interaction between the matter and a particle occurs instantaneously. We also assume that the background matter is moving isotropically and the angular particle distribution from fission is isotropic. This ensures that particles see the same distribution of background matter from any direction, which causes the total cross section to be independent of particle direction and causes the scattering cross section to depend only on the change in direction and not separately on the incident and exiting directions. There are some instances where this could break down including crystalline lattice structures and high velocity background matter. We also assume that the properties of the nuclei and the compositions of materials under consideration are known and time-independent unless explicitly stated. This means that feedback mechanisms are not explicitly treated in this equation. In general, these mechanisms are treated using other iterative processes [1]. The final assumption is that $\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E, t)$ is the mean value for the particle density. The density in a given physical object will fluctuate about this mean even in a "steady state" problem or will vary from object to object even in systems that are macroscopically similar. These fluctuations and variations become small relative to the mean when the density is integrated over a sufficiently large phase-space volume that the expected number of particles in the volume is large.

1.1.2 Eigenvalue Problems

We can utilize Equation 1.1 to solve steady-state or time-dependent problems that are driven by fixed sources. However, in reactor analysis, the k-eigenvalue problem is studied extensively to determine the criticality of a system [1]. A system is said to be critical if it can support a self-sustaining time-independent chain reaction in the absence of external sources of neutrons. This means the rate of neutron production is just equal to the losses due to absorption and leakage from the system. If this equilibrium is not established, the asymptotic distribution of neutrons will either increase or decrease. Therefore, a system is critical if a time-independent nonnegative solution to the source-free transport equation, as shown in Equation 1.5, can be found with appropriate boundary conditions.

$$\begin{aligned}\overrightarrow{\Omega} \cdot \overrightarrow{\nabla}\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) + \sigma_t(\overrightarrow{r}, E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) = \\ \frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega\nu(E')\sigma_f(\overrightarrow{r}, E')\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E') + \\ \int_{4\pi} d\overrightarrow{\Omega}' \int_0^\infty dE'\sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega}, E' \to E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}', E')\end{aligned} \tag{1.5}$$

In order to determine the multiplication factor k, we replace ν with $\frac{\nu}{k}$ as shown in Equation 1.6:

$$\begin{aligned}\overrightarrow{\Omega} \cdot \overrightarrow{\nabla}\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) + \sigma_t(\overrightarrow{r}, E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) = \\ \frac{1}{k}\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega\nu(E')\sigma_f(\overrightarrow{r}, E')\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E') + \\ \int_{4\pi} d\overrightarrow{\Omega}' \int_0^\infty dE'\sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega}, E' \to E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}', E') \,.\end{aligned} \tag{1.6}$$

A solution of this problem is a combination of k and Ψ – an eigenvalue-eigenfunction pair – that satisfies the equation and boundary conditions. There is exactly one such solution whose eigenfunction is positive throughout the domain of the problem. Its associated eigenvalue is the largest of the eigenvalues; we shall refer to it simply as k. If $k > 1$, $k < 1$, or $k = 1$, the system is supercritical, subcritical, or critical, respectively.

1.2 Solving the Neutral-Particle Transport Equation

There are currently two categories of methods for solving the transport equation: stochastic (Monte Carlo) and deterministic. These numerical methods have been developed because the transport equation can be analytically solved only for highly idealized problems. With both categories of methods, solutions of the transport equation can be computationally expensive in both time and memory.

1.2.1 Stochastic Methods

Stochastic methods are based on using the random nature of the particle interacting with the medium to simulate the histories of representative particle. These methods model the physics of the problem using probability functions. One benefit of these methods is the ability to accurately represent arbitrary geometries. These methods can use a continuous treatment of energy, space, and angle to avoid the errors associated with discretizing or averaging. A disadvantage of these methods is that they tend to consume a lot of computation time. Another is that it can be difficult for them to produce solutions that are statistically relevant in all portions of the problem domain.

1.2.2 Deterministic Methods

In this work, we will focus on deterministic methods. In these methods, we discretize the transport equation in order to approximate the problem with systems of algebraic equations that can be solved. The current methods associated with solving these equations include methods that are applied to the first-order integro-differential form of the equation, integral form, and the second-order forms (such as even-parity). We will focus on the first-order integro-differential form. Methods applied to this form can be further sub-divided based on the angular discretization treatments: discrete ordinate (S_n), spherical harmonic (P_n), and finite-element. We classify any method using discrete ordinates as a discrete ordinate method regardless of whether finite-element methods are used to discretize the spatial variables.

1.3 Dissertation Layout

We will discuss the discretization and iterative solution method of the transport equation in Section 2. We briefly describe methods for discretizing the time, energy, and spatial variables. We then give a detailed look at the discretization of the angular variables. It should be noted that our angular discretization can be applied to most spatial, time, and energy discretizations.

We then discuss the current state of the art methods employed to solve the transport equation using discretizations in the angular variable in Section 3, including a brief background in static quadrature sets. We give overviews of the Oak Ridge National Laboratory's DORT/TORT code, the University of Florida's PENTRAN code , and Los Alamos National Laboratory's Adaptive Strategies. We discuss J.C. Stone's work in adaptive angular discretizations in two dimensions and how we have extended this research.

In Section 4, we describe our derivation of four new static quadrature sets and describe our method for adaptation using these sets. We detail when we test for adaptation, where we perform these tests, how we refine our quadrature mesh in both weights and ordinates, and how we map from one quadrature region to the next. We define the implementation of our method in the Knolls Atomic Power Laboratory slice-balance, neutral-particle transport code, Jaguar. This includes the structure of the code, the structure of Jaguar, and the memory requirements necessary to implement our method. We then discuss strengths and weaknesses of our method.

In Section 5, we describe a series of test problems and apply our methods to each problem. We discuss the results from each problem and how they shaped our understanding of this adaptive angular discretization method.

In Section 6, we summarize our main results, draw conclusions, and offer suggestions for future work.

2. DISCRETIZATION OF THE NEUTRAL-PARTICLE TRANSPORT EQUATION AND SOLUTION

This research is focused on the discretization of the angular variable in the transport equation, but for completeness we briefly describe the discretization of the time, energy, and spatial variables. We begin with the neutral-particle transport equation:

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)}{\partial t}+\overrightarrow{\Omega}\cdot\overrightarrow{\nabla}\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)+\sigma_t(\overrightarrow{r},E,t)\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)=\\ \frac{\chi(E)}{4\pi}\int_0^\infty dE'\int_{4\pi}d\Omega\nu(E')\sigma_f(\overrightarrow{r},E',t)\Psi(\overrightarrow{r},\overrightarrow{\Omega},E',t)+\\ \int_{4\pi}d\overrightarrow{\Omega}'\int_0^\infty dE'\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega},E'\rightarrow E,t)\Psi(\overrightarrow{r},\overrightarrow{\Omega}',E',t)+\\ q(\overrightarrow{r},\overrightarrow{\Omega},E,t)\ .\end{aligned} \tag{2.1}$$

2.1 Time Discretization

Many time-discretization methods for the transport equation begin by integrating over a given time step, Δt. We utilize an over-bar to indicate average values of function over the time step:

$$\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)=\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}}dt\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)\ , \tag{2.2a}$$

and

$$\bar{q}(\overrightarrow{r},\overrightarrow{\Omega},E)=\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}}dtq(\overrightarrow{r},\overrightarrow{\Omega},E,t)\ . \tag{2.2b}$$

We then define the flux-weighted average for a cross section as

$$\bar{\sigma}(...)\equiv\frac{\int_{t_n}^{t_{n+1}}\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)\sigma(...)}{\int_{t_n}^{t_{n+1}}\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t)}\ . \tag{2.3}$$

Equation 2.3 is usually approximated using the value at the beginning of the time step or some extrapolated value. This is usually sufficient because the cross sections usually change very little over a given time step. We integrate over the time step, divide by Δt, and utilize Equations 2.2 and 2.3 to yield the following equation:

$$
\begin{aligned}
&\frac{1}{v(E)}\frac{\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_{n+1})-\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_n)}{\Delta t}+\\
&\overrightarrow{\Omega}\cdot\overrightarrow{\nabla}\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)+\bar{\sigma}_t(\overrightarrow{r},E)\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)=\\
&\frac{\chi(E)}{4\pi}\int_0^\infty dE'\int_{4\pi}d\Omega\nu(E')\bar{\sigma}_f(\overrightarrow{r},E')\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E')+\\
&\int_{4\pi}d\Omega'\int_0^\infty dE'\bar{\sigma}_s(\overrightarrow{r},\overrightarrow{\Omega'}\cdot\overrightarrow{\Omega},E'\to E)\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega'},E')+\\
&\bar{q}(\overrightarrow{r},\overrightarrow{\Omega},E)\ .
\end{aligned}
\tag{2.4}
$$

We assume the values of the functions at the previous time step, $\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_n)$, are known; therefore, there are two unknown functions: $\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_{n+1})$ and $\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)$. Many time-discretization methods of interest can be viewed as providing a closure equation that relates the two functions:

$$
\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)=\beta\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_{n+1})+(1-\beta)\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_n)\ ,
\tag{2.5}
$$

where β is a user-defined constant (most commonly $\frac{1}{2}$ for Crank-Nicholson or 1 for fully implicit). Some methods involve two or more steps, each of which can be cast in the form of Equations 2.4 and 2.5. Substituting Equation 2.5 into Equation 2.4 to remove the Ψ_{n+1} terms yields

$$
\begin{aligned}
&\frac{1}{v(E)\Delta t\beta}(\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)-\Psi(\overrightarrow{r},\overrightarrow{\Omega},E,t_n))+\\
&\overrightarrow{\Omega}\cdot\overrightarrow{\nabla}\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)+\bar{\sigma}_t(\overrightarrow{r},E)\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E)=\\
&\frac{\chi(E)}{4\pi}\int_0^\infty dE'\int_{4\pi}d\Omega\nu(E')\bar{\sigma}_f(\overrightarrow{r},E')\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E')+\\
&\int_{4\pi}d\Omega'\int_0^\infty dE'\bar{\sigma}_s(\overrightarrow{r},\overrightarrow{\Omega'}\cdot\overrightarrow{\Omega},E'\to E)\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega'},E')+\\
&\bar{q}(\overrightarrow{r},\overrightarrow{\Omega},E)\ .
\end{aligned}
\tag{2.6}
$$

We define the effective source and the effective cross-section as

$$
\bar{q}_{\text{eff}}(\overrightarrow{r},\overrightarrow{\Omega},E)\equiv\bar{q}(\overrightarrow{r},\overrightarrow{\Omega},E)+\frac{1}{v(E)\Delta t\beta}\bar{\Psi}(\overrightarrow{r},\overrightarrow{\Omega},E,t_n)
\tag{2.7a}
$$

and

$$\bar{\sigma}_{\text{eff},t}(\overrightarrow{r}, E) \equiv \bar{\sigma}_t(\overrightarrow{r}, E) + \frac{1}{v(E)\Delta t\beta} \ . \tag{2.7b}$$

We suppress the over-bars and the eff subscript and use Equations 2.7a and 2.7b to simplify Equation 2.6 to a "steady-state" problem that must be solved at a given step of a time-dependent problem:

$$\begin{aligned}
&\overrightarrow{\Omega} \cdot \overrightarrow{\nabla}\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) + \sigma_t(\overrightarrow{r}, E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) = \\
&\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega \nu(E')\sigma_f(\overrightarrow{r}, E')\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E') + \\
&\int_{4\pi} d\Omega' \int_0^\infty dE' \sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega}, E' \to E)\Psi(\overrightarrow{r}, \overrightarrow{\Omega}', E') + \\
&q(\overrightarrow{r}, \overrightarrow{\Omega}, E) \ .
\end{aligned} \tag{2.8}$$

Thus, time discretization can reduce a time-dependent problem to a sequence of steady-state problems.

2.2 Energy Discretization

In order to discretize over energy, we assume the angular flux can be approximated by a product of a function of energy, $F(E)$, and an energy-independent angular flux, $\Psi(\overrightarrow{r}, \overrightarrow{\Omega})$:

$$\Psi(\overrightarrow{r}, \overrightarrow{\Omega}, E) \approx \Psi(\overrightarrow{r}, \overrightarrow{\Omega})F(E) \ , \tag{2.9}$$

where $F(E)$ is the energy shape function. Dividing the energy range into G intervals and defining all particles with energy less than E_{g-1} and greater than E_g to be in group g yields the following definition for the "group-g" angular flux:

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

When we integrate the transport equation over the energy interval associated with the $g-$th group, we can manipulate the result so that it contains cross sections that have been averaged with a weight function of F [1]. These "multigroup" cross sections are as follows

$$\sigma_{t,g}(\overrightarrow{r}) = \frac{\int_g^{g-1} dE\sigma_t(\overrightarrow{r}, E)F(E)}{\int_g^{g-1} dEF(E)} \ , \tag{2.11a}$$

$$\sigma_{s,g'\to g}(\overrightarrow{r}, \overrightarrow{\Omega} \cdot \overrightarrow{\Omega}) = \frac{\int_{g'}^{g'-1} dE' \int_g^{g-1} dE\sigma_s(\overrightarrow{r}, \overrightarrow{\Omega} \cdot \overrightarrow{\Omega}, E' \to E)F(E')}{\int_{g'}^{g'-1} dE'F(E')} \ , \tag{2.11b}$$

and

$$\sigma_{f,g}(\overrightarrow{r}) = \frac{\int_g^{g-1} dE\sigma_f(\overrightarrow{r}, E)F(E)}{\int_g^{g-1} dEF(E)} \ . \tag{2.11c}$$

We also define

$$\chi_g = \int_g^{g-1} dE\chi(E) \ , \tag{2.12}$$

$$q_g(\overrightarrow{r}, \overrightarrow{\Omega}) = \int_g^{g-1} dEq(\overrightarrow{r}, \overrightarrow{\Omega}, E) \ , \tag{2.13}$$

and

$$\Phi_g(\overrightarrow{r}) = \int_{4\pi} d\Omega\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) \ . \tag{2.14}$$

We integrate Equation 2.8 over the $g-$th energy interval. Using Equations 2.10 - 2.14, we can re-write the integrated equation as a single group equation:

$$\begin{aligned} &\overrightarrow{\Omega} \cdot \overrightarrow{\nabla}\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_{t,g}(\overrightarrow{r})\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) = \frac{\chi_g}{4\pi}\sum_{g'=1}^{G} \nu_{g'}\sigma_{f,g'}(\overrightarrow{r})\Phi_{g'}(\overrightarrow{r}) + \\ &\int_{4\pi} d\Omega' \sum_{g'=1}^{G} \sigma_{s,g'\to g}(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega})\Psi_{g'}(\overrightarrow{r}, \overrightarrow{\Omega}') + q_g(\overrightarrow{r}, \overrightarrow{\Omega}) \quad \text{for} \ \ g = 1 : G \ . \end{aligned} \tag{2.15}$$

We often rearrange the scattering term as follows

$$\begin{aligned} &\overrightarrow{\Omega} \cdot \overrightarrow{\nabla}\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_{t,g}(\overrightarrow{r})\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) = \\ &\int_{4\pi} d\Omega'\sigma_{s,g\to g}(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega})\Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}') + \frac{\chi_g}{4\pi}\sum_{g'=1}^{G} \nu_{g'}\sigma_{f,g'}(\overrightarrow{r})\Phi_{g'}(\overrightarrow{r}) + \\ &\int_{4\pi} d\Omega' \sum_{g'=1, g'\neq g}^{G} \sigma_{s,g'\to g}(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega})\Psi_{g'}(\overrightarrow{r}, \overrightarrow{\Omega}') + q_g(\overrightarrow{r}, \overrightarrow{\Omega}) \ \text{for} \ \ g = 1 : G \ . \end{aligned} \tag{2.16}$$

We then define

$$\begin{aligned} S_{ext,g}(\overrightarrow{r}, \overrightarrow{\Omega}) = & \frac{\chi_g}{4\pi} \sum_{g'=1}^{G} \nu_{g'} \sigma_{f,g'}(\overrightarrow{r}) \Phi_{g'}(\overrightarrow{r}) + \\ & \int_{4\pi} d\Omega' \sum_{g'=1, g' \neq g}^{G} \sigma_{s,g' \to g}(\overrightarrow{r}, \overrightarrow{\Omega}' \to \overrightarrow{\Omega}) \Psi_{g'}(\overrightarrow{r}, \overrightarrow{\Omega}') + \\ & q_g(\overrightarrow{r}, \overrightarrow{\Omega}) \ \text{ for } \ g = 1 : G \ . \end{aligned} \tag{2.17}$$

Inserting Equation 2.17 into Equation 2.16 results in

$$\begin{aligned} & \overrightarrow{\Omega} \cdot \overrightarrow{\nabla} \Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_{t,g}(\overrightarrow{r}, \overrightarrow{\Omega}) \Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}) = \\ & \int_{4\pi} d\Omega' \sigma_{s,g \to g}(\overrightarrow{r}, \overrightarrow{\Omega}' \to \overrightarrow{\Omega}) \Psi_g(\overrightarrow{r}, \overrightarrow{\Omega}') + S_{ext,g}(\overrightarrow{r}, \overrightarrow{\Omega}) \ \text{ for } \ g = 1 : G \ . \end{aligned} \tag{2.18}$$

We suppress the group subscript and are left with the one-group transport equation:

$$\begin{aligned} & \overrightarrow{\Omega} \cdot \overrightarrow{\nabla} \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_t(\overrightarrow{r}, \overrightarrow{\Omega}) \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) = \\ & \qquad \int_{4\pi} d\Omega' \sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \to \overrightarrow{\Omega}) \Psi(\overrightarrow{r}, \overrightarrow{\Omega}') + S_{ext}(\overrightarrow{r}, \overrightarrow{\Omega}) \ . \end{aligned} \tag{2.19}$$

Thus, the multigroup approximation reduces an energy-dependent equation to a series of coupled one-group equations. The coupling is through the scattering and fission terms.

2.3 Angle Discretization /Quadrature

The discrete-ordinate (D-O) method replaces direction integrals with quadrature sums. The quadrature sums use a quadrature set, which is a finite set of directions and associated weights. Each sum needs the angular flux at the quadrature directions. Let us determine exactly what integrals need to be approximated by quadrature sums. To do this we will manipulate the scattering term in the way that has become standard in neutron transport. We begin by expanding the angular flux, $\Psi(\overrightarrow{r}, \overrightarrow{\Omega})$, in spherical harmonics functions and the differential scattering cross section, $\sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \to \overrightarrow{\Omega})$, in Legendre polynomials:

$$\Psi(\overrightarrow{r}, \overrightarrow{\Omega}) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \sum_{\tilde{m}=-l}^{l} \phi_{l\tilde{m}}(\overrightarrow{r}) Y_{l\tilde{m}}(\overrightarrow{\Omega}) \ , \tag{2.20}$$

$$\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\rightarrow\overrightarrow{\Omega})=\frac{1}{2\pi}\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega})=\sum_{l'=0}^{L}\frac{2l'+1}{4\pi}\sigma_{sl'}(\overrightarrow{r})P_{l'}(\mu_0)\ , \tag{2.21}$$

where

$$\phi_{l\tilde{m}}(\overrightarrow{r})=\int_{4\pi}d\Omega\Psi(\overrightarrow{r},\overrightarrow{\Omega})Y^*_{l\tilde{m}}(\overrightarrow{\Omega})\ , \tag{2.22}$$

and

$$\sigma_{sl'}(\overrightarrow{r})=\int_{-1}^{1}d\mu_0\sigma_s(\overrightarrow{r},\mu_0)P'_l(\mu_0)\ . \tag{2.23}$$

In order to perform this expansion, we have assumed that the scattering probability depends only on the scattering angle, whose cosine is

$$\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega}=\mu_0\ . \tag{2.24}$$

We then insert Equations 2.20 and 2.21 into the scattering term,

$$\int_{4\pi}d\Omega'\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\rightarrow\overrightarrow{\Omega})\Psi(\overrightarrow{r},\overrightarrow{\Omega}')$$

, as shown below:

$$\begin{aligned}&\int_{4\pi}d\Omega'\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega})\Psi(\overrightarrow{r},\overrightarrow{\Omega}')=\\&\int_{4\pi}d\Omega'\sum_{l'=0}^{L}\frac{2l'+1}{4\pi}\sigma_{sl'}(\overrightarrow{r})P_{l'}(\mu_0)\sum_{l=0}^{\infty}\frac{2l+1}{4\pi}\sum_{\tilde{m}=-l}^{l}\phi_{l\tilde{m}}(\overrightarrow{r})Y_{l\tilde{m}}(\overrightarrow{\Omega}')\ .\end{aligned} \tag{2.25}$$

The addition theorem states

$$P_l(\overrightarrow{\Omega}\cdot\overrightarrow{\Omega}')=\frac{1}{2l+1}\sum_{m=-l}^{l}Y^*_{lm}(\overrightarrow{\Omega}')Y_{lm}(\overrightarrow{\Omega})\ . \tag{2.26}$$

We will utilize the addition theorem to simplify Equation 2.25 to the following:

$$\begin{aligned}&\int_{4\pi}d\Omega'\sigma_s(\overrightarrow{r},\overrightarrow{\Omega}'\cdot\overrightarrow{\Omega})\Psi(\overrightarrow{r},\overrightarrow{\Omega}')=\\&\int_{4\pi}d\Omega'\sum_{l'=0}^{L}\frac{2l'+1}{4\pi}\sigma_{sl'}(\overrightarrow{r})\frac{1}{2l'+1}\sum_{m=-l'}^{l'}Y^*_{l'm}(\overrightarrow{\Omega}')Y_{l'm}(\overrightarrow{\Omega})*\\&\sum_{l=0}^{\infty}\frac{2l+1}{4\pi}\sum_{\tilde{m}=-l}^{l}\phi_{l\tilde{m}}(\overrightarrow{r})Y_{l\tilde{m}}(\overrightarrow{\Omega}')\ .\end{aligned} \tag{2.27}$$

We realize $Y^*_{lm}(\overrightarrow{\Omega})$ and $Y_{l'm'}(\overrightarrow{\Omega})$ are orthogonal and the following equation is true:

$$\int_{4\pi} d\Omega Y^*_{lm}(\overrightarrow{\Omega}) Y_{l'm'}(\overrightarrow{\Omega}) = 4\pi\delta_{ll'}\delta_{mm'} \ . \tag{2.28}$$

Using this result yields the following representation of the scattering term:

$$\begin{gathered}\int_{4\pi} d\Omega' \sigma_s(\overrightarrow{r}, \overrightarrow{\Omega}' \cdot \overrightarrow{\Omega}) \Psi(\overrightarrow{r}, \overrightarrow{\Omega}') = \\ \sum_{l=0}^{L} \frac{2l+1}{4\pi} \sigma_{sl}(\overrightarrow{r}) \sum_{m=-l}^{l} Y_{lm}(\overrightarrow{\Omega}) \phi_{lm}(\overrightarrow{r}) \ .\end{gathered} \tag{2.29}$$

Inserting Equation 2.29 into Equation 2.19 results in

$$\begin{gathered}\overrightarrow{\Omega} \cdot \overrightarrow{\nabla} \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_t(\overrightarrow{r}) \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) = \\ \sum_{l=0}^{L} \frac{2l+1}{4\pi} \sigma_{sl}(\overrightarrow{r}) \sum_{m=-l}^{l} Y_{lm}(\overrightarrow{\Omega}) \phi_{lm}(\overrightarrow{r}) + S_{ext}(\overrightarrow{r}, \overrightarrow{\Omega}) \ .\end{gathered} \tag{2.30}$$

We see that our quadrature set must be able to accurately integrate the integrals that appear in Equation 2.22:

$$\sum_{m=0}^{M} w_m \Psi(\overrightarrow{r}, \Omega_m) Y^*_{l\tilde{m}}(\Omega_m) \approx \int_{4\pi} d\Omega \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) Y^*_{l\tilde{m}}(\overrightarrow{\Omega}) \ . \tag{2.31}$$

In many applications, we assume isotropic scattering, or $L = 0$, which reduces Equation 2.30 to

$$\overrightarrow{\Omega} \cdot \overrightarrow{\nabla} \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) + \sigma_t(\overrightarrow{r}) \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) = \frac{1}{4\pi} \sigma_s(\overrightarrow{r}) \phi(\overrightarrow{r}) + S_{ext}(\overrightarrow{r}, \overrightarrow{\Omega}) \ . \tag{2.32}$$

We can solve Equation 2.32 along a given direction, $\overrightarrow{\Omega}_m$, and use the following equation to close the system:

$$\phi(\overrightarrow{r}) \equiv \int_{4\pi} d\Omega \Psi(\overrightarrow{r}, \overrightarrow{\Omega}) \approx \sum_{m=0}^{M} w_m \Psi(\overrightarrow{r}, \overrightarrow{\Omega}_m) \ . \tag{2.33}$$

Using the subscript m, we can re-write Equation 2.32 along a given direction as:

$$\overrightarrow{\Omega}_m \cdot \overrightarrow{\nabla} \Psi_m(\overrightarrow{r}) + \sigma_t(\overrightarrow{r}) \Psi_m(\overrightarrow{r}) = \frac{1}{4\pi} \sigma_s(\overrightarrow{r}) \phi(\overrightarrow{r}) + S_{ext,m}(\overrightarrow{r}) \ . \tag{2.34}$$

Therefore, we solve for the angular flux along specific directions and use Equation 2.33 to determine the scalar flux (and Equation 2.22 to form higher flux moments if scattering is anisotropic).

2.4 Spatial Discretization

In order to discretize the spatial variable, we divide our problem geometry into three-dimensional regions. Each region consists of an arbitrary number of non-overlapping cells. Each cell can be an arbitrary polyhedron with an arbitrary number of faces. For a standard rectangular mesh, the cells would be regular cuboids with six rectangular faces. Once the mesh is defined, the transport equation is approximated on each cell so that the approximate solution has a small number of unknowns per cell. Exactly how this is done is different for different spatial discretization methods. Below we briefly outline some of the many spatial discretization methods in use today.

2.4.1 General

In three-dimensional Cartesian coordinates, the three-dimensional transport equation, Equation 2.34, becomes:

$$\begin{aligned}&(\mu_m\frac{\partial}{\partial x}+\eta_m\frac{\partial}{\partial y}+\xi_m\frac{\partial}{\partial z})\Psi_m(\overrightarrow{r})+\sigma_t(\overrightarrow{r})\Psi_m(\overrightarrow{r})=\\&\frac{1}{4\pi}\sigma_s(\overrightarrow{r})\phi(\overrightarrow{r})+S_{ext,m}(\overrightarrow{r})\ \text{ for }\ m=1:M\ ,\end{aligned}\tag{2.35}$$

where

$$\mu_m=\overrightarrow{\Omega}_m\cdot\overrightarrow{n}_x\ ,\tag{2.36a}$$

$$\eta_m=\overrightarrow{\Omega}_m\cdot\overrightarrow{n}_y\ ,\tag{2.36b}$$

and

$$\xi_m=\overrightarrow{\Omega}_m\cdot\overrightarrow{n}_z\ .\tag{2.36c}$$

For simplicity we assume that cross sections are constant in a given cell:

$$\begin{aligned}&(\mu_m\frac{\partial}{\partial x}+\eta_m\frac{\partial}{\partial y}+\xi_m\frac{\partial}{\partial z})\Psi_m(\overrightarrow{r})+\sigma_t\Psi_m(\overrightarrow{r})=\\&\frac{1}{4\pi}\sigma_s\phi(\overrightarrow{r})+S_{ext,m}(\overrightarrow{r})\ \text{ for }\ m=1:M\ .\end{aligned}\tag{2.37}$$

On our rectangular cell, we define the following:

$$V = \Delta x \Delta y \Delta z \; , \tag{2.38a}$$

$$\frac{1}{V}\int_V dV \Psi_m(\overrightarrow{r}) \equiv \Psi_{m,V} \; , \tag{2.38b}$$

$$\frac{1}{V}\int_V dV S_{ext,m}(\overrightarrow{r}) \equiv S_{ext,m,V} \; , \tag{2.38c}$$

and

$$\frac{1}{V}\int_V dV \phi(\overrightarrow{r}) \equiv \phi_V \; . \tag{2.38d}$$

We define the boundary angular fluxes as

$$\Psi_{T,m} \equiv \int_{\Delta x}\int_{\Delta y} dxdy \Psi_m(\overrightarrow{r}_T) \; , \tag{2.39a}$$

$$\Psi_{\text{BOT},m} \equiv \int_{\Delta x}\int_{\Delta y} dxdy \Psi_m(\overrightarrow{r}_{\text{BOT}}) \; , \tag{2.39b}$$

$$\Psi_{R,m} \equiv \int_{\Delta y}\int_{\Delta z} dydz \Psi_m(\overrightarrow{r}_R) \; , \tag{2.39c}$$

$$\Psi_{L,m} \equiv \int_{\Delta y}\int_{\Delta z} dydz \Psi_m(\overrightarrow{r}_L) \; , \tag{2.39d}$$

$$\Psi_{F,m} \equiv \int_{\Delta x}\int_{\Delta z} dxdz \Psi_m(\overrightarrow{r}_F) \; , \tag{2.39e}$$

and

$$\Psi_{\text{BACK},m} \equiv \int_{\Delta x}\int_{\Delta z} dxdz \Psi_m(\overrightarrow{r}_{\text{BACK}}) \; , \tag{2.39f}$$

where $\overrightarrow{r}_T$, $\overrightarrow{r}_{\text{BOT}}$, $\overrightarrow{r}_R$, $\overrightarrow{r}_L$, $\overrightarrow{r}_F$, and $\overrightarrow{r}_{\text{BACK}}$ are the points on the top, bottom, right, left, front, and back faces, respectively. We will now integrate Equation 2.37 over the spatial phase-space $(\Delta x, \Delta y, \Delta z)$ and divide by the volume, V:

$$\begin{aligned}
&\frac{\mu_m}{V}(\Psi_{R,m} - \Psi_{L,m}) + \frac{\eta_m}{V}(\Psi_{F,m} - \Psi_{\text{BACK},m}) + \\
&\frac{\xi_m}{V}(\Psi_{T,m} - \Psi_{\text{BOT},m}) + \sigma_t \Psi_{m,V} = \\
&\frac{1}{4\pi}\sigma_s \phi_V + S_{ext,m,V} \;\; \text{for} \;\; m = 1 : M \; .
\end{aligned} \tag{2.40}$$

This equation is exact, to the extent that Equation 2.37 is exact; however, it is one equation for four unknown quantities: the volume-averaged angular flux, $\Psi_{m,V}$, and three of the six surface-averaged angular fluxes (the ones on “outgoing” surfaces for the given direction). Different spatial-discretization methods make different approximations to provide three independent equations that relate the four unknowns to each other. Once these equations are known, the system can be solved for the four unknowns. If this is done for each direction in the quadrature set then it is a simple matter to use Equation 2.33 to determine the scalar flux and thus update the scattering source for a new iteration.

2.4.2 Slice Balance Approach

Because this research was performed using the Knolls Atomic Power Laboratory’s neutral-particle transport code, Jaguar, we utilize a slice-balance approach (SBA) for spatial discretization in our testing [2] [3] [4]. This method uses an angle-dependent spatial decomposition, as does the method of characteristics, to divide the three-dimensional cells into slices for each discrete ordinate direction. The SBA uses a multiple balance approach using exact spatial moments balance equations on cells and slices along with auxiliary relations on slices [3]. The three-dimensional decomposition of cells into slices can be seen in Figure 2.1. This figure shows a 12-sided polyhedron decomposed into 17 slices for direction $\overrightarrow{\Omega}$. The ability to decompose

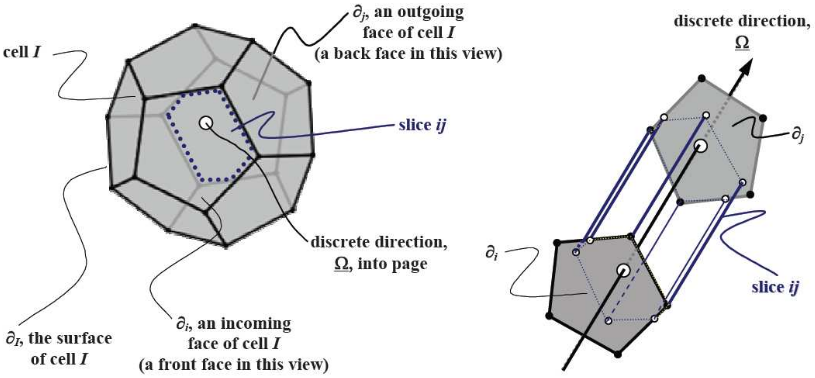
cell I
∂_j, an outgoing face of cell I (a back face in this view)
slice ij
discrete direction, $\underline{\Omega}$, into page
∂_I, the surface of cell I
∂_i, an incoming face of cell I (a front face in this view)
discrete direction, $\underline{\Omega}$
∂_j
∂_i
slice ij

Fig. 2.1. Decomposition of a 3D polyhedron into slices.

cells into slices allows the transport equation to be solved on each slice in a cell and then summed to get cell averaged and face values. These slices are cell local which ensures the slice does not extend into adjacent cells, unlike the method of characteristics. This is obvious in the two-dimensional example shown in Figure 2.2. The mathematics of the SBA is described in detail in R.E. Grove's work [3] [4].

While we have used the SBA for this research, our method is applicable to most spatial discretization methods with the restriction on the spatial grid that the interface between spatial regions that may have different quadrature sets must be perpendicular to the coordinate axes. This forces a clean separation between incoming and outgoing portions of the direction domain and thereby allows accurate mapping from one quadrature region to the next. However, the interior cells can be of arbitrary size, shape, and complexity.

2.5 Iterative Solution

In order to solve this discretized system, we utilize a nested iterative approach. An outer iteration is a loop over energy groups from the highest-energy group to the

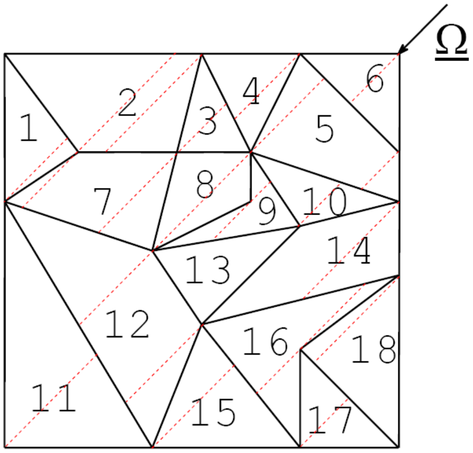

Fig. 2.2. Decomposition of an arbitrary 2D mesh into slices.

lowest-energy group. In each energy group, we perform inner iterations. A single inner iteration includes a transport sweep, which is the solution of Equation 2.40 using the previous inner iteration's scalar flux for the within-group scattering source and the latest available scalar fluxes for the scattering source from other energy groups. The within-group scalar flux is then updated using Equation 2.33. The latest and previous within-group scalar flux are compared on each cell. If they are "close enough" in every cell, the inner iteration for the current group is considered converged and we move to the next group and update its source using the latest scalar fluxes from other groups. After the inner iterations for all groups have converged, the scalar fluxes on each cell and each group are compared against those from the previous outer iteration. If they are "close enough" for each cell and each group, we consider the outer iteration to be converged.

2.6 Summary

Using the discretization techniques described in this section, the neutral-particle transport equation can be simplified into a system of algebraic equations. This allows a solution to be found using finite computational resources in an iterative process. However, these approximations create discretization errors in the solution. For example, Discrete Ordinate (D-O) methods suffer from discretization errors including "ray effects" [5]. Figure 2.3 shows the scalar flux as a function of spatial location for a point source in a low-scattering region for a two-dimensional problem. We show a pseudo-color plot of scalar flux using eight discrete ordinates and the "ray effects" are quite clear. The errors arise from the inability of a fixed quadrature set to accurately integrate functions that are not smooth. The more "peaky" the angular flux, the more ordinates are needed to accurately integrate this function. Even in seemingly simple transport problems, the angular flux is not smooth. We illustrate this non-smooth behavior with an infinite square lattice of circular fuel pins in water. Implementing a multi-group D-O method, we see that the fast-group angular flux at a certain point in the infinite lattice as function of azimuthal angle is quite complex as shown in Figure 2.4 [6]. Over the years, there have been numerous schemes for mitigating ray effects using specific quadrature sets [7] [8] [9] [10] [11] [12] [13] [14]. In the next section, we discuss methods devised for reducing or controlling the errors introduced by this D-O approximation.

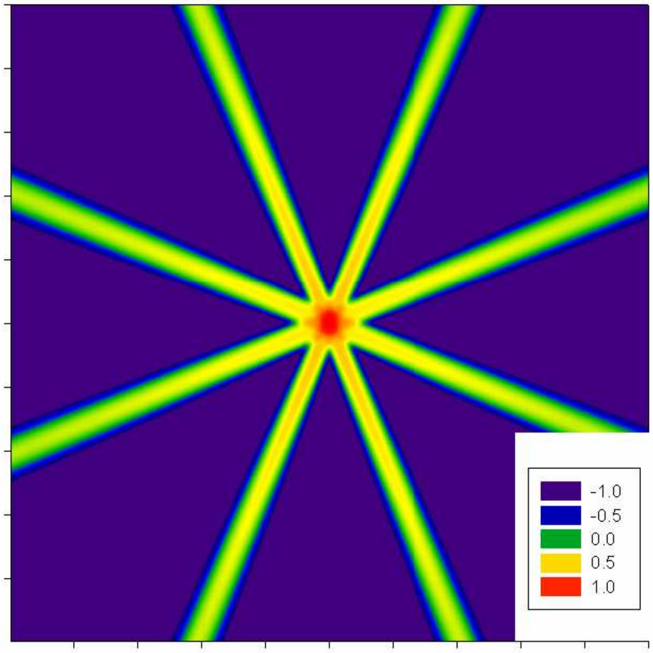

Fig. 2.3. Ray effects of simple 2D problem with 8 discrete ordinates. The values in the legend are powers of ten.

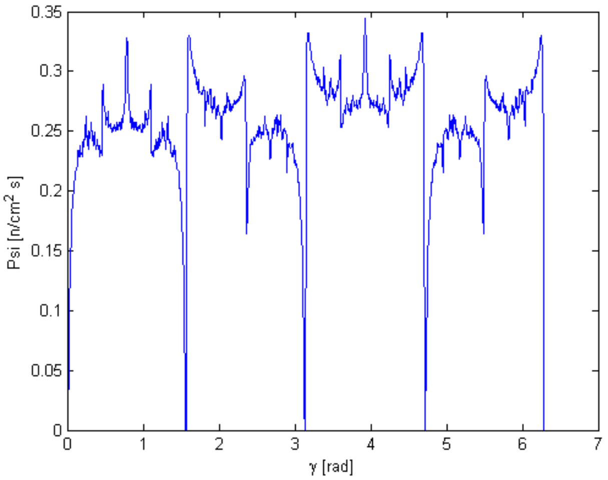

Fig. 2.4. Fast-group angular flux at a certain point in an infinite square lattice of circular fuel pins in water, as a function of azimuthal angle. The polar angle is 90°, so the particles move in the xy plane.

3. STATE OF THE ART IN DISCRETE-ORDINATE QUADRATURE METHODS

3.1 Static Quadrature Sets

To our knowledge, all discrete-ordinates codes that are widely used today use static quadrature sets. These sets are chosen before the problem is run and kept constant throughout the solution process. In this section we describe several kinds of quadrature sets that are used in multi-dimensional discrete-ordinates calculations. We will see that a common guiding principle in the creation of such quadrature sets is to integrate spherical-harmonics functions as accurately as possible. Normally, quadrature set directions are described using the three variables μ, η and ξ which are derived from the angles θ and γ which completely describe the direction as illustrated in Figure 3.1 and shown Equation 3.1:

$$\Omega_x = \mu = cos(\gamma)sin(\theta) = cos(\gamma)\sqrt{1-\xi^2} \ , \tag{3.1a}$$

$$\Omega_y = \eta = sin(\gamma)sin(\theta) = sin(\gamma)\sqrt{1-\xi^2} \ , \tag{3.1b}$$

and

$$\Omega_z = \xi = cos(\theta) \ . \tag{3.1c}$$

The direction cosines are related because the squares of the cosines must sum to one as shown below:

$$\mu^2 + \eta^2 + \xi^2 = 1 \ . \tag{3.2}$$

In most quadrature sets, the ordinates are defined on given ξ values which we call a polar level (because ξ is dependent only on the polar angle θ). In many cases, “triangular” quadrature sets are utilized. A “triangular” quadrature set refers to the shape of the location of the ordinates in a given octant. The triangular quadrature set’s largest ξ level will have one ordinate, the next largest ξ level will have two

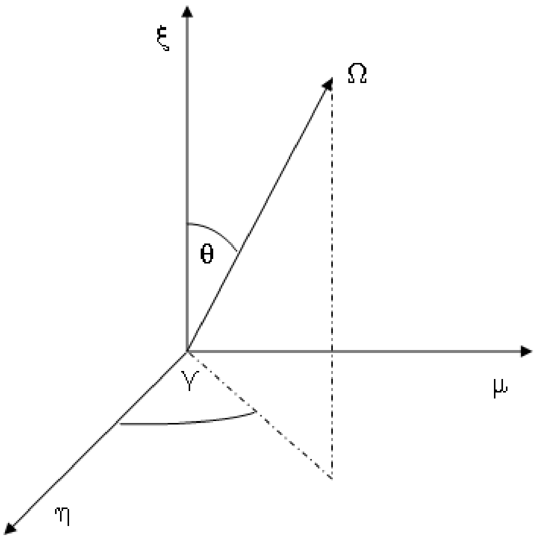

Fig. 3.1. Coordinate system for direction Ω.

ordinates, the next level will have three ordinates, etc. This gives the appearance of a triangle. There are also "square" quadrature sets which have a given number of levels, N_l, and N_l ordinates on each level, as well as, "rectangular" quadrature sets which have a given number of levels, N_l, and $2N_l$ ordinates on each level.

3.1.1 Level Symmetric

The Level Symmetric (S_n) quadratures require that the set of ordinates be rotationally symmetric about the three axes for a given octant. Because of this constraint, it can be shown that once the location of the first ordinate has been chosen, all other ordinates are determined [1]. This single degree of freedom in location of the ordinate along with the weights of each ordinate are determined by requiring that the set be able to integrate the highest order spherical-harmonic function over the unit sphere. This set is generally considered an extension of a Gaussian set from one-dimension. The S_n quadrature set is triangular and has $\frac{n}{2}$ ξ levels in each octant. This creates $\frac{n(n+2)}{8}$ directions in each octant. As n is increased, some of the weights of this set decrease and eventually go negative when the quadrature order n is equal to or larger than 20. This can lead to unphysical solutions when the angular flux is not smooth. The Level Symmetric S_{16} quadrature set is shown in Figure 3.2.

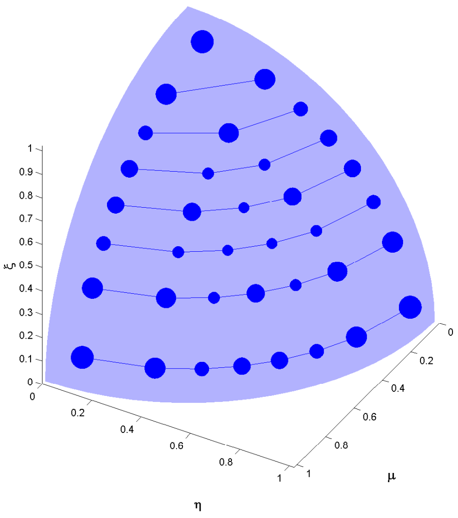

Fig. 3.2. Level symmetric S_{16} quadrature set.

3.1.2 Gauss-Chebyshev

If the symmetry constraints imposed by level-symmetric sets are eased, then, other quadrature sets can be devised. Some of these alternative sets are considered “product” sets because they combine a one-dimensional quadrature set along the polar axis with a one-dimensional set in the azimuthal variable, in a way that involves products of the weights from the two sets. The Gauss-Chebyshev set utilizes a one-dimensional Gaussian polar quadrature and an equally-weighted Chebyshev quadrature in the azimuthal variable [15]. Different azimuthal quadrature orders may be associated with different polar ordinates. A benefit of these sets is the ability to have an arbitrary number of angles without limitations from potentially problematic negative weights. The Gauss-Chebyshev quadrature set with a level-symmetric-like appearance is shown in Figure 3.3.

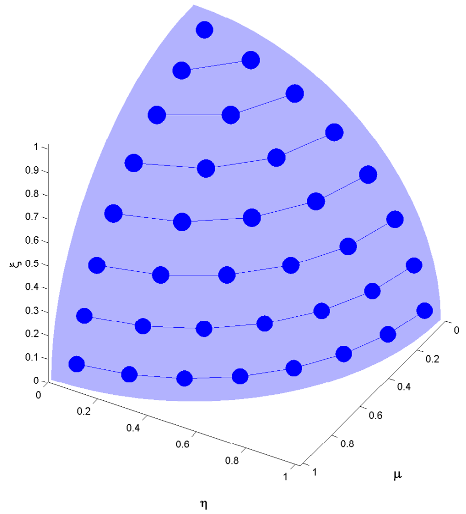

Fig. 3.3. Gauss-Chebyshev S_{16}-like quadrature set.

3.1.3 Quadruple Range

Abu-Shumays developed the Quadruple Range (QR) quadrature sets to accurately integrate functions that are discontinuous across octant boundaries [12] [13] [14]. They are product sets in which the polar and azimuthal quadrature sets are determined by requiring exact integration of certain spherical-harmonic functions over each individual octant instead of the entire unit sphere. An example of a QR S_{16}-like quadrature set that Abu-Shumays developed is seen in Figure 3.4, which illustrates that the ordinates are closer to the axes and the central weights are larger, compared to either the level-symmetric or triangular Gauss-Chebychev sets. These sets are limited in an octant to eighteen polar levels and thirty-six azimuthal directions. This limitation arises from the incredibly difficult set of nonlinear algebraic equations that must be solved to arrive at the QR sets.

3.2 Oak Ridge National Lab's DORT/ TORT Code

To address ray effects and other quadrature-induced errors, methods have been developed that use different quadrature sets in different spatial regions, user-defined problem-specific quadrature sets, or algorithms that attempt to adapt the quadrature to the solution. Oak Ridge National Laboratory maintains two- and three-dimensional neutral-particle transport codes (DORT and TORT, respectively) that can utilize different quadrature sets in different spatial regions [16]. The user selects each quadrature set. These codes map the angular flux exiting a region in a given quadrature direction for that region to the nearest ordinate (quadrature direction) in the region on the other side of the boundary, and then use a multiplicative factor to ensure conservation of the particle flow rate across the boundary.

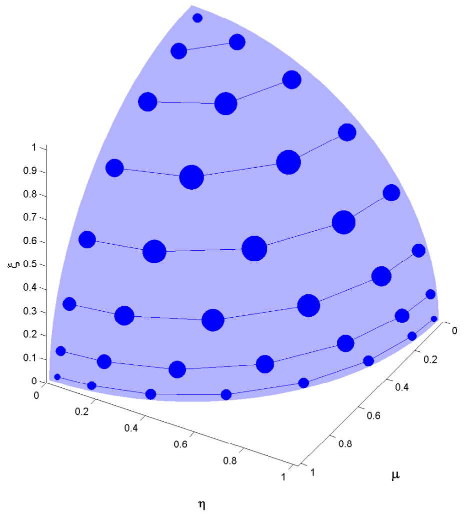

Fig. 3.4. Abu-Shumays' S_{16}-like quadrature set.

3.3 University of Florida's PENTRAN Code

G. Longoni and A. Haghighat developed quadratures based on ordinate splitting which they called Regional Angular Refinement (RAR) [17] [18] [19]. Their method begins with an existing quadrature set and refines a given ordinate that is chosen by the user. They used product quadrature sets in which each ordinate was associated with an area on the unit sphere ($\Delta\gamma \times \Delta\xi$). Their method replaced the chosen quadrature point and weight with a local product set on the associated patch of area. A single point can be split into an $n{\times}n$ set of points in this way, with n chosen by the user. The user selects an initial quadrature set and selects which ordinates to split. The user also decides how many new ordinates to create from the original ordinate. The method makes no attempt to retain the ability of the original quadrature set to integrate high-order spherical-harmonics functions, with the exception that it does require that if the user refines one patch of direction space, the corresponding patch in the opposite direction must also be refined. The method did not introduce different quadrature sets in different regions, but used a single modified set throughout the problem domain.

3.4 P.N. Brown's Locally Refined Quadrature Rules

Brown, Chang, and Clouse developed a user-defined locally refined quadrature set on triangular elements on the sphere [20]. A key difference between Brown, Chang, and Clouse's quadrature set and Longoni and Haghighat's quadrature set is that Brown and Chang chose to determine the weights by a solving a "Quadratic Programming Problem," which ensures that locally refined sets retain the ability to integrate spherical-harmonics functions through a specified order. They also start with a user-defined cone of refinement, as performed in Longoni and Haghighat's RAR work, and thus do not introduce a method that refines based on the behavior of the solution.

3.5 J.C. Stone's 2D Adaptive Strategies

Stone and Adams developed a two-dimensional D-O method that is truly adaptive (not user-defined) [21] [22] [23]. It allows different quadrature sets in different spatial regions and employs a simple algorithm to map angular fluxes across region boundaries. It uses a "finite element view" of the angular flux to guide the definition of weights, interpolations, and mapping methods. To make refinement decisions, the method solves for the angular flux at certain directions and compares those results with the interpolated values that come from the basis functions and the angular fluxes at the already-existing quadrature points. The method we propose in the next section directly builds upon this work and extends it to three dimensions.

3.6 Extension to Three Dimensions

There are a number of difficulties that arise when the phase space is expanded from two to three spatial dimensions. The definition of the basis functions has more possibilities and the integration of those basis functions on the surface of the sphere is difficult. The limits of integration for these basis functions are based on patches (such as spherical triangles) on the unit sphere and therefore the method for refinement is a two-dimensional refinement versus a one-dimensional refinement. Another difficulty is passing information from one quadrature region to the next. The conservation of scalar flux and current can be difficult to apply without significantly altering the angular flux's angular-dependent shape. We describe these challenges and how we have addressed these issues in the following section.

4. ADAPTIVE QUADRATURE METHOD

4.1 Overview

Previous work indicates that local adaptation of the quadrature set can be useful for solving particle-transport problems. In this work, we extend previous methods from two to three dimensions and also introduce some new ideas, with the goal of developing a powerful tool to address difficult problems. Our method includes:

- Division of the spatial domain into "quadrature regions," each of which may have a different quadrature set;
- A linear-discontinuous finite-element (LDFE) approximation of the angular flux on sub-polygons (initially triangles) on the surface of the unit sphere;
- Use of the LDFE basis functions to create interpolated angular fluxes for comparison against calculated fluxes at certain "test" directions to determine if coarsening or refinement is necessary;
- Each quadrature weight defined as the integral of the quadrature point's basis function;
- Local refinement and local coarsening on sub-polygons;
- A mapping scheme that conserves the discrete-ordinate version of partial current normal to each interface between quadrature regions;
- An attempt to estimate the errors in the scalar flux and current introduced by the D-O method and its use of quadrature sums to approximate angular integration.

4.2 Timing of Refinement and Coarsening Testing

In our method, refinement and coarsening testing in a steady-state problem is performed after every n sweeps, where n is a user-defined number. In time-dependent problems, which we do not solve in this work, there could be several time steps between refinement and coarsening testing. The frequency of this testing would initially be user-defined. At some point, this frequency could also be determined by an adaptive method. This would enable the code to refine and coarsen more often where necessary and less often when conditions permit.

4.3 Location of Refining and Coarsening Testing

We define contiguous or periodic spatial regions that have the same quadrature set and we call these "quadrature" regions. The testing for refinement and coarsening occurs only on the boundaries of these quadrature regions. If the boundary is composed of many cell faces, each cell face is tested. If on ALL cell faces of all boundaries of a region, coarsening is found to be permitted, the quadrature points are removed from the quadrature set. If on ANY cell face on any boundary of the region, refinement is found to be needed, the test directions are added to the quadrature set and no further cell faces are tested for those directions.

4.4 Method for Defining Refined Quadrature

Our method for refining our quadrature mesh is based on a polyhedron's projection onto the unit sphere. Our base case is the regular octahedron (made up of eight equilateral flat triangles), which corresponds to the octant-based view normally used in transport theory, as seen in Figure 4.1. A point on each flat triangle can be defined using two variables, which we call $\tilde{u}$ and v as shown in Figure 4.2, and each such point uniquely defines a point on the surface of the sphere, or equivalently a direction vector from the sphere center to the point. We refine each equilateral triangle by dividing it into 4 equilateral sub-triangles as shown in Figure 4.3. Each triangle contains four discrete ordinate points. The center point lies at the centroid of the triangle. The other three points can be defined in a variety of manners. The locations of the four points in a triangle are the locations where basis functions are defined to have values of unity or zero; thus, given that the quadrature weights are integrals of basis functions, the locations of the four points completely determine the four quadrature weights. Our initial method insisted that the three non-central

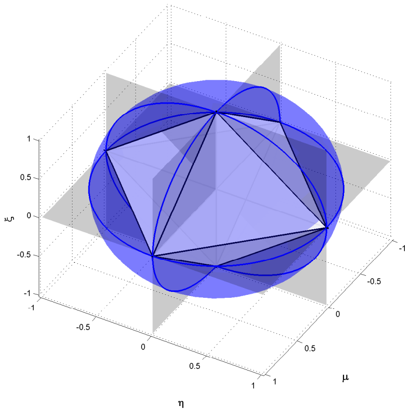

Fig. 4.1. Basic octahedron view of discrete ordinates.

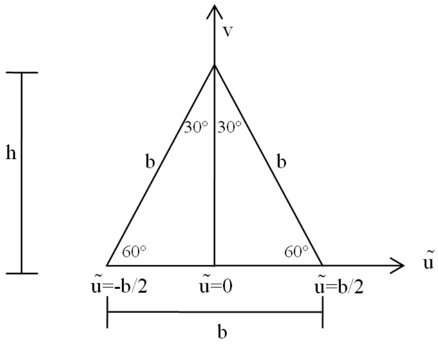

Fig. 4.2. Flat view of triangle using $\tilde{u}$ and v.

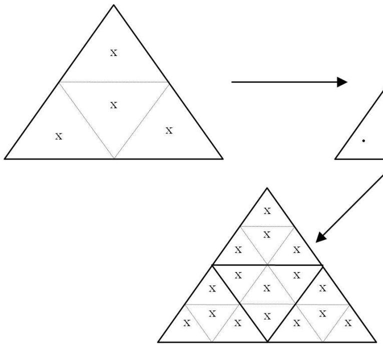

Fig. 4.3. One triangle refinement.

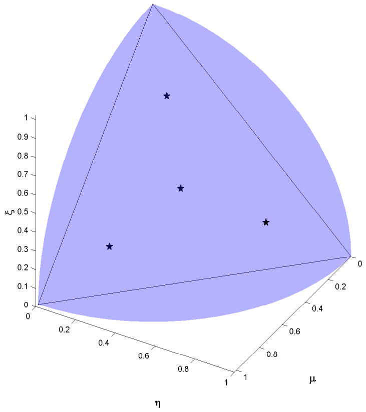

Fig. 4.4. Original octant-based triangle.

points be at the centroids of their respective sub-triangles, as shown in Figure 4.4. This choice leads to negative weights and the problems associated with them, as we discuss below. We have devised alternative locations that ensure that the center point's weight on any triangle always equals the spherical surface area of its sub-triangle. One variant ensures that each sub-triangle spherical surface area is equal to the weight of its associated point.

Figure 4.5 illustrates the method for defining discrete ordinates, where the black stars represent the points on each of the four flat triangles and the blue spheres represents their projections onto the unit sphere, where the sphere's volume represents the point's weight. This refinement process can continue for as many steps as necessary to achieve a given accuracy of solution. Several uniform refinements are shown in Figure 4.6, where each triangle contains four ordinates. We define the first refinement, shown in Figure 4.6, as the base triangle. A localized refinement with four levels of refinement for the base-top, -left, and -right triangles, five levels of refinement for the base-center triangle's top, left, and right sub-triangles, and 6 levels of refinement for the base-center triangle's center sub-triangle is shown in Figure 4.7. Because of this hierarchy, our method of refinement is relatively simple to implement. Because we use the flat triangle to determine our refinement methodology, the surface area of the spherical triangles for a given refinement level are different. Because the weights of the four points in any triangle sum to the associate spherical surface area, these surface areas give some indication of the magnitude of our quadrature weights. A comparison of the surface area as a function of the level of uniform refinement can be seen in Table 4.1. Even with seven levels of uniform refinement (16384 triangles and 65536 directions per octant), the maximum ratio of surface areas is 5.12.

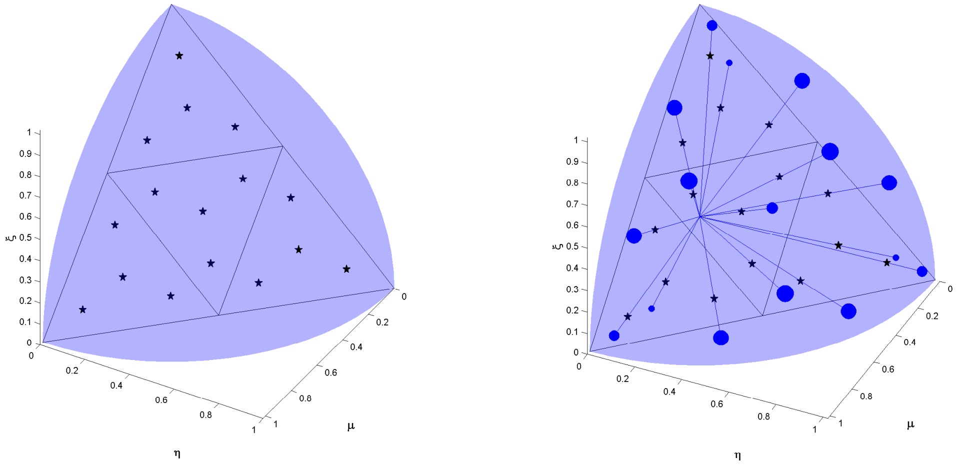

Fig. 4.5. Projection from the flat triangle to the unit sphere to produce discrete ordinates.

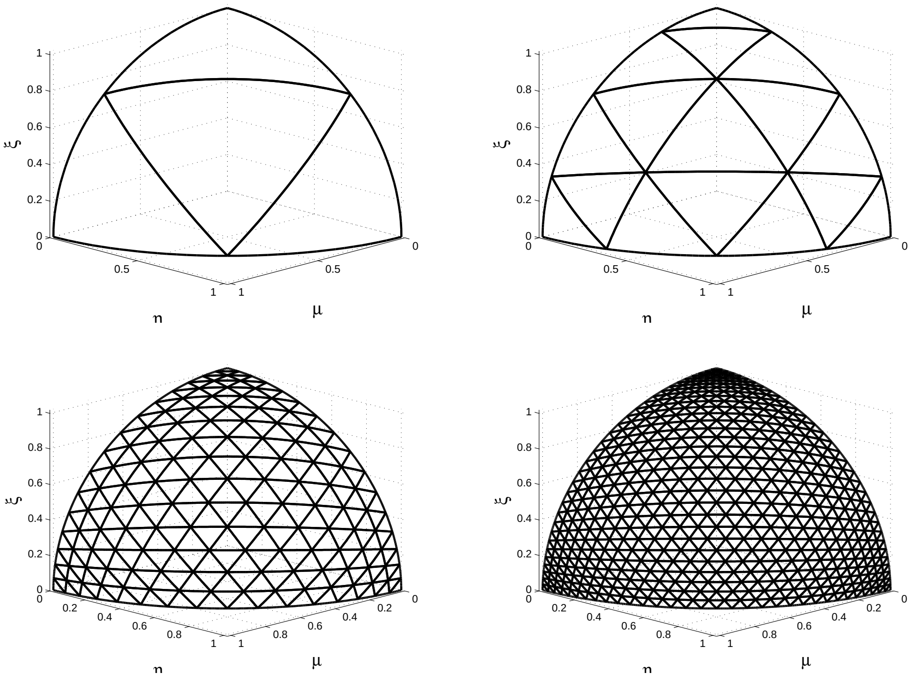

Fig. 4.6. Refinement of quadrature triangles: First, Second, Fourth, and Fifth Refinements.

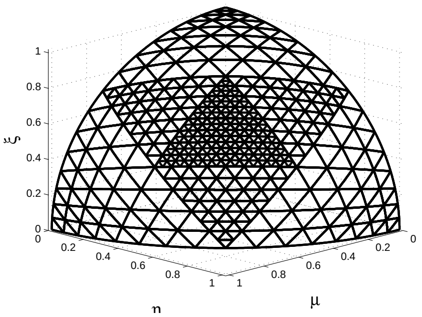

Fig. 4.7. An example of local quadrature mesh refinement. The base-top, -left, and -right triangles have been refined four times. The base-center triangle's top, left, and right sub-triangles have been refined five times, and the base-center triangle's center sub-triangle has been refined six times.

4.5 Basis Function Definitions

Each flat triangle contains four points, corresponding to the discrete ordinates used in the quadrature set. The coordinate system utilized can be seen in Figure 4.8. The weights are determined using a discontinuous linear finite element approximation

Table 4.1
Comparison of largest and smallest surface area.

Refinement Step	Minimum SA	Maximum SA	Ratio
1	3.34e-001	5.51e-001	1.62
2	5.27e-002	1.55e-001	2.95
3	1.01e-002	4.01e-002	3.97
4	2.22e-003	1.01e-002	4.56
5	5.20e-004	2.54e-003	4.88
6	1.26e-004	6.34e-004	5.03
7	3.10e-005	1.59e-004	5.12

in the direction cosines $(\Omega_x, \Omega_y, \Omega_z)$, which are defined as

$$\Omega_x = \mu = cos(\gamma)sin(\theta) = cos(\gamma)\sqrt{1-\xi^2}\ , \tag{4.1a}$$

$$\Omega_y = \eta = sin(\gamma)sin(\theta) = sin(\gamma)\sqrt{1-\xi^2}\ , \tag{4.1b}$$

and

$$\Omega_z = \xi = cos(\theta)\ . \tag{4.1c}$$

On each triangle, we define four basis functions, b_m, where $m = 1, 2, 3, 4$:

$$b_m(\overrightarrow{\Omega}) = c_{c,m} + c_{\mu,m}\mu + c_{\eta,m}\eta + c_{\xi,m}\xi\ . \tag{4.2}$$

On our coarsest quadrature mesh, the four basis functions are plotted in Figure 4.9. Here we convert the basis functions from the (μ, η, ξ) space to the (θ, γ) space using Equation 4.1. The basis functions as a function of $\tilde{u}$ and v can be seen in Figure 4.10. The constants of Equation 4.2 are determined by specifying that the basis functions are "cardinal" functions at the four ordinates, meaning that each has a value of unity at its quadrature point and zero at the other three points. The constants are found

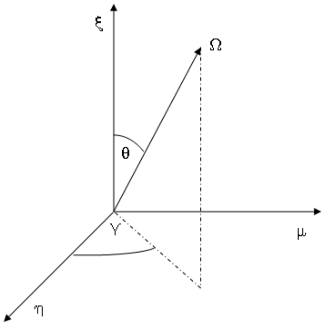

Fig. 4.8. Spherical coordinate system.

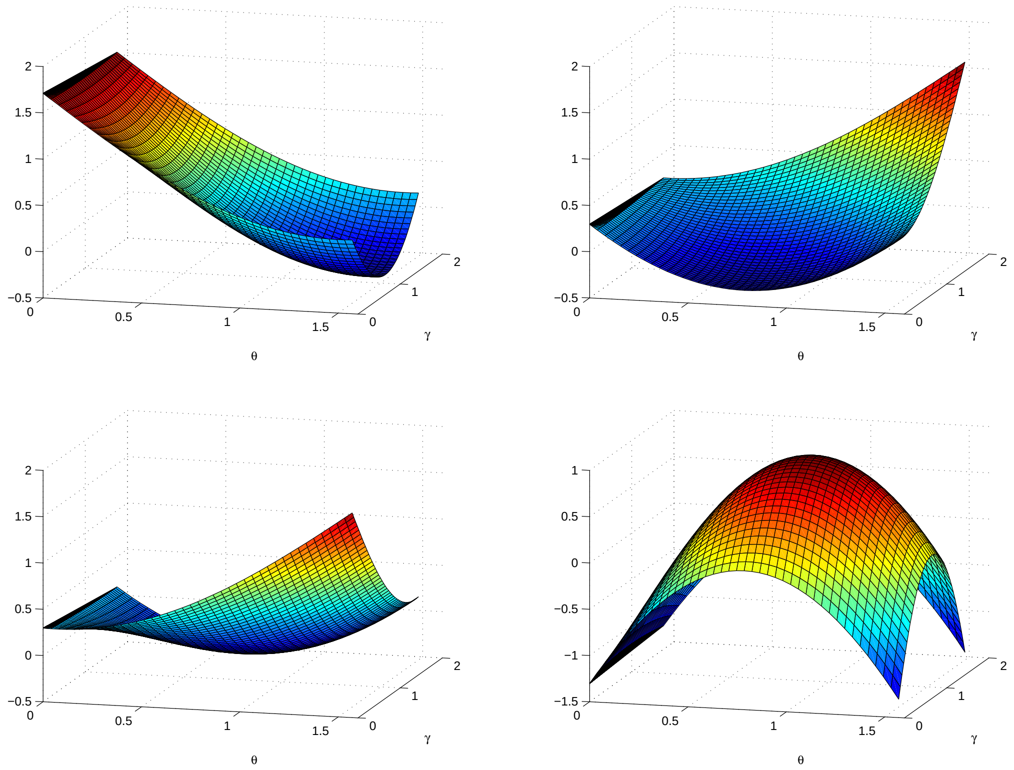

Fig. 4.9. On the coarsest octant triangle, the top, left, right, and center basis functions plotted along $\gamma(0, \frac{\pi}{2})$ and $\theta(0, \frac{\pi}{2})$.

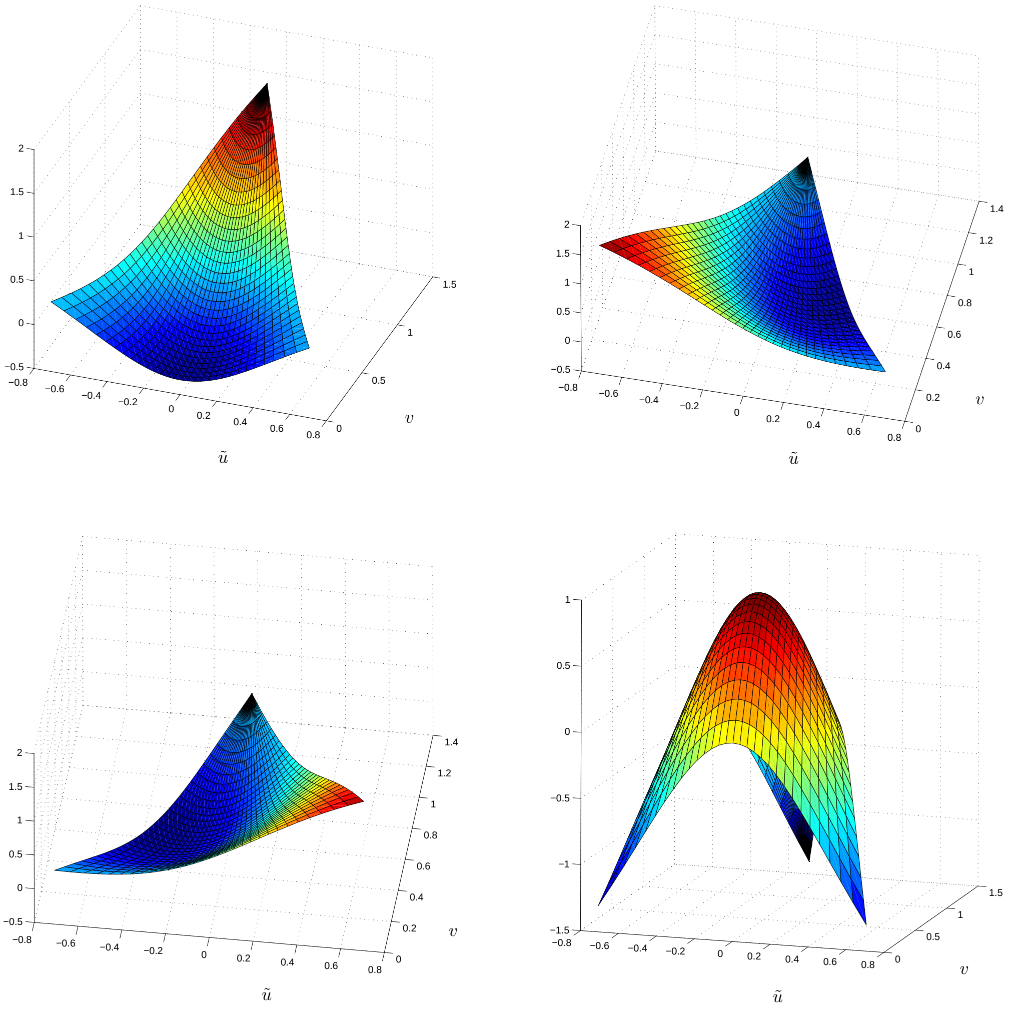

Fig. 4.10. On the coarsest octant triangle, the top, left, right, and center basis functions plotted along $\tilde{u}$ and v.

by solving the following equation:

$$\begin{bmatrix} 1 & \mu_1 & \eta_1 & \xi_1 \\ 1 & \mu_2 & \eta_2 & \xi_2 \\ 1 & \mu_3 & \eta_3 & \xi_3 \\ 1 & \mu_4 & \eta_4 & \xi_4 \end{bmatrix} \begin{bmatrix} c_{c,1} & c_{c,2} & c_{c,3} & c_{c,4} \\ c_{\mu,1} & c_{\mu,2} & c_{\mu,3} & c_{\mu,4} \\ c_{\eta,1} & c_{\eta,2} & c_{\eta,3} & c_{\eta,4} \\ c_{\xi,1} & c_{\xi,2} & c_{\xi,3} & c_{\xi,4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4.3}$$

Here $\{\mu_i, \eta_i, \xi_i\}$ are the components of the four ordinates in the triangle.

4.6 Weight and Ordinate Determination

We define the weight of each ordinate as

$$w_m = \int_{\Omega_{\text{triangle}}} d\Omega b_m(\overrightarrow{\Omega}) . \tag{4.4}$$

We have been unable to perform this integral analytically, largely because the boundaries of our spherical triangles are not simple functions of θ and γ. We have utilized a numerical method for this integration, as shown in Appendix A.

We have developed four methods for defining the points on the triangle, which in turn define the directions and weights in the quadrature set. We describe each of these methods below.

4.6.1 Linear Discontinuous Finite Element Quadrature Sets with Ordinates at the Centroids of the Sub-Triangles (LDFE-Center)

This method specifies that the four points lie at the centroids of the flat subtriangles. This method re-uses the four points from the parent triangle and introduces 12 new ordinates as shown in Figure 4.3. Because the triangle is defined recursively, the center ordinate of each sub-triangle has the same ordinate as the original parent triangle. Unfortunately, this method creates large differences between the surface area of a given spherical sub-triangle and the weight associated with the correspond-

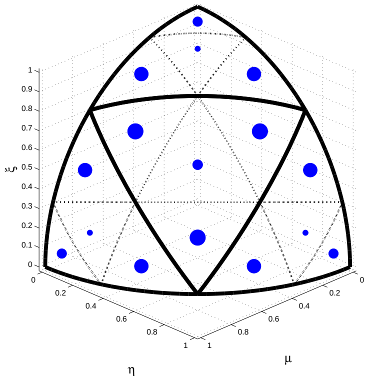

Fig. 4.11. Surface area versus associated discrete-ordinate weights for the first LDFE-Center refinement in the first octant. Quadrature weights are indicated by the size of the blue dots. Note that they bear little relation to the areas of the sub-triangles in which they reside.

ing point. Figure 4.11 graphically shows the surface area of a given spherical triangle and the associated weight produced using this centroid method. In this figure there are four main triangles, each with four sub-triangles. The figure shows that the center point of each main triangle has a strikingly small weight relative to the other points. In fact, as refinement continues, the center weights in some triangles become negative. This caused significant difficulties in our mapping algorithms and led us to seek alternative locations for the quadrature points.

4.6.2 Linear Discontinuous Finite Element Quadrature Sets Specifying That the Center Sub-Triangle's Surface Area Equals Its Weight, Using Θ on the Unit Sphere (LDFE-Θ)

We noted that the center weight of each triangle was increasingly small and in some situations negative if we specified the three outer points (top, left, and right) lie at the centroids of the their respective sub-triangles. In order to alleviate this

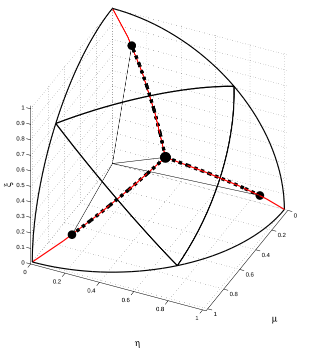

Fig. 4.12. Example of Θ method for determining three outer ordinates on a triangle. The ratio of the black arc length to the red arc length represents the "Θ" ratio. This ratio is determined on the surface of the sphere.

problem, we developed a method in which the outer locations were chosen to force the center point's weight to equal the surface area (SA) of the center triangle:

$$w_{m,\text{center}} = SA_{\text{center}} \ . \tag{4.5}$$

There are many ways to choose the outer points to achieve this outcome. Here we describe a method that uses a "Θ ratio." This method requires the center point to be at the centroid of the triangle and the three outer points to lie along the great circles between the center point and the associated corners of the spherical triangle as shown in Figure 4.12. For example, the top point's ratio along this path can be described as a ratio of the arc length from the center point to the top point divided by the arc length from the center point to the top corner of the spherical triangle, as shown in Equation 4.6:

$$\text{ratio}_{\Theta,top} \equiv \frac{s_{\text{center-top}}}{s_{\text{center-topCorner}}} \equiv \frac{\arccos(\Omega_{\text{center}} \cdot \Omega_{\text{top}})}{\arccos(\Omega_{\text{center}} \cdot \Omega_{\text{topCorner}})} \ . \tag{4.6}$$

We then require all ratios to be equal as follows:

$$\text{ratio}_{\Theta,top} = \text{ratio}_{\Theta,left} = \text{ratio}_{\Theta,right} = \text{ratio}_{\Theta} \ . \tag{4.7}$$

We define a variable r_T such that

$$r_T = \Omega_{\text{center}} \cdot \Omega_{\text{top}} \ . \tag{4.8}$$

Letting $\overrightarrow{\Omega}_{CT}$, $\overrightarrow{\Omega}_C$, and $\overrightarrow{\Omega}_T$ stand as the ordinates of the top corner point, the center point, and the top point, respectively allows the definition of the top point to be

$$\overrightarrow{\Omega}_T = r_T \overrightarrow{\Omega}_c + (1 - r_T^2)^{1/2} \frac{\overrightarrow{\Omega}_{CT} - (\overrightarrow{\Omega}_c \cdot \overrightarrow{\Omega}_{CT})\overrightarrow{\Omega}_c}{|\overrightarrow{\Omega}_{CT} - (\overrightarrow{\Omega}_c \cdot \overrightarrow{\Omega}_{CT})\overrightarrow{\Omega}_c|} \ . \tag{4.9}$$

Equations 4.5, 4.7, and 4.9 completely define the location of the four points. However, these equation must be solved iteratively. We use a secant method to approximate the derivative of the relative error (RE) with respect to ratio_Θ:

$$RE_{\text{center}} \equiv \frac{SA_{\text{center}} - w_{\text{center}}}{SA_{\text{center}}} \tag{4.10}$$

and

$$\left(\frac{\partial RE_{\text{center}}}{\partial \text{ratio}_\Theta}\right)^{[1]} \approx \frac{RE^{[1]}_{\text{center}} - RE^{[0]}_{\text{center}}}{\text{ratio}^{[1]}_\Theta - \text{ratio}^{[0]}_\Theta} \ . \tag{4.11}$$

We utilize Equation 4.11 to approximate the derivative and the equation for the next ratio becomes

$$\text{ratio}^{[n]}_\Theta = \text{ratio}^{[n-1]}_\Theta - \frac{RE^{[n-1]}_{\text{center}}}{\left(\frac{\partial RE_{\text{center}}}{\partial \text{ratio}_\Theta}\right)^{[n-1]}} \ . \tag{4.12}$$

While the secant method is not guaranteed to converge in all situations, our function of relative error verses ratio is a smooth, monotonically decreasing function. If we start with two bounding guesses, we can guarantee convergence.

4.6.3 Linear Discontinuous Finite Element Quadrature Sets That Specify the Center Sub-Triangle's Surface Area Equals Its Weight, Using L in the Flat Triangle (LDFE-L)

The Θ method is computationally difficult because of multiple $tan, cos, arccos$, and $arctan$ used in evaluating ratio_Θ. We developed an alternative method that

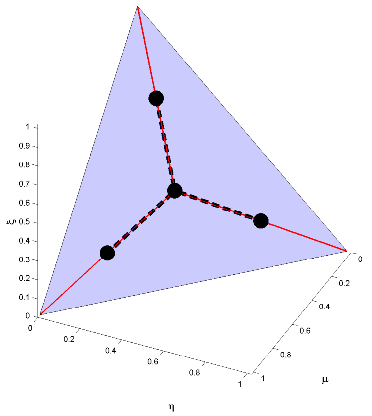

Fig. 4.13. Example of L method for determining three outer ordinates on a triangle. The ratio of the black line to the red line represents the "L" ratio. This ratio is determined on the flat triangle.

also specifies the surface area of the center triangle be equal to its weight but uses a different constant ratio. For the top point, we define this "L" ratio as the ratio of the distance on the original flat triangle from the center point to the top point compared to the distance on the flat triangle from the center point to the top corner point as illustrated in Figure 4.13.

$$\text{ratio}_{L,top} \equiv \frac{d_{\text{center-top}}}{d_{\text{center-topCorner}}} \equiv \frac{\left((\tilde{u}_c - \tilde{u}_t)^2 + (v_c - v_t)^2\right)^{1/2}}{\left((\tilde{u}_c - \tilde{u}_{\text{cornerTop}})^2 + (v_c - v_{\text{cornerTop}})^2\right)^{1/2}} \ . \tag{4.13}$$

We then specify that all three ratios must be equal:

$$\text{ratio}_{L,top} = \text{ratio}_{L,left} = \text{ratio}_{L,right} = \text{ratio}_L \ . \tag{4.14}$$

We therefore can do all the mathematics on the flat triangle, which eliminates the inverse trigonometric functions. The $(\tilde{u}, v)$ of the top point is

$$(\tilde{u}, v)_{\text{top}} = \text{ratio}_L * (\tilde{u}, v)_{\text{center}} + (1 - \text{ratio}_L) * (\tilde{u}, v)_{\text{topCorner}} \ . \tag{4.15}$$

The left and right points can be described using equations similar to Equation 4.15. Equations 4.5, 4.14, and 4.15 completely define the location of the four points. However, this set of equations must also be solved in an iterative manner. We use a secant method to approximate the derivative of the relative error (RE) with respect to ratio$_L$:

$$\left(\frac{\partial RE_{\text{center}}}{\partial \text{ratio}_L}\right)^{[1]} \approx \frac{RE^{[1]}_{\text{center}} - RE^{[0]}_{\text{center}}}{\text{ratio}^{[1]}_L - \text{ratio}^{[0]}_L} \ . \tag{4.16}$$

In the same manner as we used the secant method for the Θ method, we utilize the secant method for the L method:

$$\text{ratio}^{[n]}_L = \text{ratio}^{[n-1]}_L - \frac{RE^{[n-1]}_{\text{center}}}{\left(\frac{\partial RE_{\text{center}}}{\partial \text{ratio}_L}\right)^{[n-1]}} \ . \tag{4.17}$$

While the secant method is not guaranteed to converge in all situations, our function of relative error verses ratio is again a smooth, monotonically decreasing function. If we start with two bounding guesses, we can guarantee convergence.

4.6.4 Linear Discontinuous Finite Element Quadrature Sets That Require All Sub-Triangles' Surface Areas to be Equal to Their Associated Ordinate Weights (LDFE-SA)

Both of the previous methods were devised in order to alleviate the problem of the center ordinate weight being drastically different than its associated surface area and in some cases going negative. Once we established that we could force the center surface area equal to the weight, we realized that we could ensure all surface areas were equivalent to their weights on a given spherical triangle. Our method begins with an initial guess for the location of the four points generated by either the L or Θ method. We will label these weights as w^0_t, w^0_l, w^0_r, and w^0_c, where t=top, l=left, r=right, and c=center. Our initial weight establishes that $w^0_c = SA_c$, where $SA =$ surface area. These weights have a certain ratio associated with them ρ^0_t, ρ^0_l, and ρ^0_r which are currently equal. We now allow the ratios to change independently by an amount δ:

$$\rho^n_t = \rho^{[n-1]}_t + \delta^{[n-1]}_t \ , \tag{4.18a}$$

$$\rho^n_l = \rho^{[n-1]}_l + \delta^{[n-1]}_l \ , \tag{4.18b}$$

and

$$\rho_r^n = \rho_r^{[n-1]} + \delta_r^{[n-1]} \ . \tag{4.18c}$$

We want a set of equations in which $\overrightarrow{\delta}^{[n]} \equiv \left[\delta_t^{[n]}\ , \delta_r^{[n]}\ , \delta_l^{[n]}\ \right]^t$ converges to 0. We will use the following equation to find $\overrightarrow{\delta}^{[n-1]}$:

$$A^{[n-1]}\overrightarrow{\delta}^{[n-1]} = \overrightarrow{b}^{[n-1]} \ , \tag{4.19}$$

where

$$A^{[n-1]} = \begin{pmatrix} \frac{\partial w_t}{\partial \rho_t} & \frac{\partial w_t}{\partial \rho_l} & \frac{\partial w_t}{\partial \rho_r} \\ \frac{\partial w_l}{\partial \rho_t} & \frac{\partial w_l}{\partial \rho_l} & \frac{\partial w_l}{\partial \rho_r} \\ \frac{\partial w_r}{\partial \rho_t} & \frac{\partial w_r}{\partial \rho_l} & \frac{\partial w_r}{\partial \rho_r} \end{pmatrix}^{[n-1]} \ , \tag{4.20}$$

and

$$\overrightarrow{b}^{[n-1]} = \begin{pmatrix} SA_t - w_t^{[n-1]} \\ SA_l - w_l^{[n-1]} \\ SA_r - w_r^{[n-1]} \end{pmatrix} \ . \tag{4.21}$$

We again use a secant two point method to evaluate the partial derivative terms in the A matrix:

$$\left(\frac{\partial w_x}{\partial \rho_y}\right)^{[n-1]} \equiv \frac{w_x^{[n-1]} - w_x^{[n-2]}}{\rho_y^{[n-1]} - \rho_y^{[n-2]}} \ . \tag{4.22}$$

We invert A and solve for $\overrightarrow{\delta}^{[n-1]}$ to determine the next ratio points using Equation 4.18, and evaluate the relative error of each of the discrete ordinate weights verse the surface areas. If the errors are not less than a given tolerance, we go through this process again. We require two points to initially evaluate the partial derivative so we have arbitrarily chosen the first guess for the ratios as

$$\rho_y = \frac{\rho_y^0 + 1}{2} \ , \tag{4.23}$$

for $y = l, r, t$. This is equivalent to setting the first guess equal to the average of the current ratio and the corner of the triangle (ratio=1). Again, this method is based on the secant method and a linear perturbation method that are not guaranteed to converge. However, the relative error functions as a function of the ratios are smooth and generally decrease monotonically. We note that this quadrature set is a “geometric” quadrature set where the weight of each point corresponds to a surface area on the unit sphere. Another interesting result is that, as the number of refinements increase, the spherical triangle gets more and more spherically equilateral. This means that the three arcs defining the triangle approach the same value. This forces the surface areas of the three outer sub-triangles to converge to the same value.

The four choices of point locations described above lead to four quadrature sets that share much in common but differ in detail. In our numerical testing we will compare and contrast the four methods.

4.7 Mapping across Quadrature Boundaries

Because there can be different numbers and/or locations of quadrature points in adjacent quadrature regions, a simple one-to-one mapping is not always possible or desirable. The mapping of one quadrature set to another requires compromises. An optimal mapping technique would ensure conservation of as many moments of the spherical harmonics functions as possible, would not create negative fluxes, and would preserve the angular shape of the flux. Difficulties and compromises arise because no mapping can accommodate all of these requirements. We will first discuss coarse-to-fine mapping algorithms and then discuss fine-to-coarse.

4.7.1 Coarse-to-fine Mapping

Consider a boundary between two regions that have different quadrature sets, and consider an octant of directions for which the upstream region has fewer quadrature points than the downstream region. In this case a coarse-to-fine mapping must be performed as illustrated in Figure 4.14. This moves information from a coarse triangle with four ordinates and fluxes to a fine triangle with sixteen ordinates and fluxes.

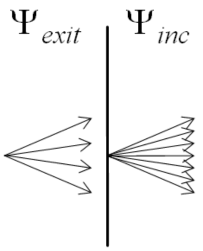

Fig. 4.14. Example of coarse-to-fine mapping.

(There are many other cases, such as going from four ordinates to 64 ordinates, but these can be handled by a series of four-to-sixteen mappings.) We therefore have sixteen degrees of freedom to determine the optimal way to pass information across a boundary.

Conservation of ϕ, J_x, J_y, and J_z

The first mapping technique was determined to ensure conservation of four quantities judged to be important to an accurate transport solution: face scalar fluxes and face currents. On each sub-triangle on the coarse triangle, there is one coarse point and four fine points. On each sub-triangle, we conserve these four moments using the following equation:

$$A\Psi = b \ , \tag{4.24}$$

where

$$A \equiv \begin{bmatrix} w_0 & w_1 & w_2 & w_3 \\ w_0\mu_0 & w_1\mu_1 & w_2\mu_2 & w_3\mu_3 \\ w_0\eta_0 & w_1\eta_1 & w_2\eta_2 & w_3\eta_3 \\ w_0\xi_0 & w_1\xi_1 & w_2\xi_2 & w_3\xi_3 \end{bmatrix} \tag{4.25}$$

and

$$b \equiv \begin{bmatrix} w_{\text{coarse}}\Psi_{\text{coarse}} \\ w_{\text{coarse}}\mu_{\text{coarse}}\Psi_{\text{coarse}} \\ w_{\text{coarse}}\eta_{\text{coarse}}\Psi_{\text{coarse}} \\ w_{\text{coarse}}\xi_{\text{coarse}}\Psi_{\text{coarse}} \end{bmatrix} . \tag{4.26}$$

We invert A and solve for Ψ at the four fine points on each sub-triangle. Unfortunately, this method skews the angular flux and can introduce unphysical negative fluxes.

Conservation of ϕ using linear discontinuous finite element basis functions

Another method is to use linear basis functions on the coarse triangle to interpolate/extrapolate the flux values at the sixteen fine points and then ensure conservation of the scalar flux across the face. This method maps from the coarse region N with 4 directions to the fine region M with 16 directions using the following equations:

$$\Psi_m(\overrightarrow{\Omega}_m) = \sum_{n=1}^{4} \Psi(\overrightarrow{\Omega}_n) b_n(\overrightarrow{\Omega}_m) \text{ for } m = 1:16, \tag{4.27}$$

$$F_{\text{conserve}} = \frac{\sum_{n=1}^{4} w_n \Psi_n}{\sum_{m=1}^{16} w_m \Psi_m}, \tag{4.28}$$

and

$$\Psi_M(\overrightarrow{\Omega}_m) \leftarrow F_{\text{conserve}} \Psi_M(\overrightarrow{\Omega}_m) \text{ for } m = 1:16 . \tag{4.29}$$

This algorithm ensures conservation of the discrete-ordinate scalar flux on a face. However, this scalar flux on the face is an unphysical quantity and does not ensure conservation of particles. Also, the linear basis functions have to extrapolate at some points which can introduce negative flux values at those points.

Conservation of partial current normal to the face using linear discontinuous finite element basis functions

The next method is to use linear basis functions on the coarse triangle to interpolate/extrapolate the flux values at the sixteen fine points and then ensure conservation of the current normal to the face. This method maps from the coarse region N with 4 directions to the fine region M with 16 directions using the following equations:

$$\Psi_m(\overrightarrow{\Omega}_m) = \sum_{n=1}^{4} \Psi(\overrightarrow{\Omega}_n) b_n(\overrightarrow{\Omega}_m) \text{ for } m = 1:16, \tag{4.30}$$

$$F_{\text{conserve}} = \frac{\sum_{n=1}^{4}(\overrightarrow{\Omega}_n \cdot \overrightarrow{n}) w_n \Psi_n}{\sum_{m=1}^{16}(\overrightarrow{\Omega}_m \cdot \overrightarrow{n}) w_m \Psi_m}, \tag{4.31}$$

and

$$\Psi_M(\overrightarrow{\Omega}_m) \leftarrow F_{\text{conserve}} \Psi_M(\overrightarrow{\Omega}_m) \text{ for } m = 1:16\ . \tag{4.32}$$

This routine will ensure conservation of the discrete-ordinate current normal to the face, which conserves particle flow rate across a boundary. However, the linear basis functions will have to extrapolate at some points which can introduce negative flux values at those points.

Conservation of partial current normal to the face using constant discontinuous finite element basis functions

The last method is to use a constant basis function on the sub-triangles of the coarse triangle to interpolate/extrapolate the flux values at the four fine points on each sub-triangle and then ensure conservation of the current normal to the face. This makes each of the four points on the coarse sub-triangle equal to $\Psi_{\text{fine,st}}$ which is defined as:

$$\Psi_{\text{fine,st}} = \frac{\Psi_{\text{coarse}}(\overrightarrow{\Omega}_{\text{coarse}} \cdot \overrightarrow{n}) w_{\text{coarse}}}{\sum_{i=1}^{I=4}(\overrightarrow{\Omega}_{\text{fine},i} \cdot \overrightarrow{n}) w_{\text{fine},i}} \text{ for each sub-triangle } st = 1:4\ . \tag{4.33}$$

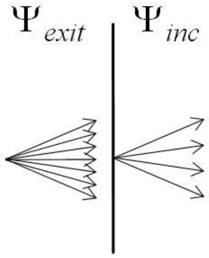

Fig. 4.15. Example of fine-to-coarse mapping.

This method loses information about the variation of the angular flux within a subtriangle, but will not introduce negative fluxes if the weights are positive, and will still conserve partial current across the face.

Implementation

After testing many options, we settled on implementing conservation of partial current normal to the face using linear finite element basis functions. This high-order method conserves particles and uses the finite element basis functions. However, this method does introduce negative fluxes in some situations. If negative fluxes are introduced, we move to the method of conserving partial current normal to the face using constant basis functions. The idea of moving from a higher order approximation to a lower order approximation if negative values are introduced is common in numerical methods.

4.7.2 Fine-to-coarse Mapping

If particles flow from a region with a more refined quadrature set to one with a less refined set, coarsening must be performed as illustrated in Figure 4.15. This moves information from a fine triangle with sixteen ordinates and fluxes to a coarse triangle with four ordinates and fluxes. (Again there are many possible cases, but all can be handled by recursive application of sixteen-to-four mappings.) We therefore have four degrees of freedom to determine the optimal way to pass information across a boundary.

Conservation of ϕ, J_x, J_y, and J_z

The first mapping technique we developed ensured conservation of face scalar fluxes and face currents. Because we have four degrees of freedom, we utilize four equations to solve for Ψ as follows

$$A\Psi_{\text{coarse}} = b \ , \tag{4.34}$$

where

$$A \equiv \begin{bmatrix} w_0 & w_1 & w_2 & w_3 \\ w_0\mu_0 & w_1\mu_1 & w_2\mu_2 & w_3\mu_3 \\ w_0\eta_0 & w_1\eta_1 & w_2\eta_2 & w_3\eta_3 \\ w_0\xi_0 & w_1\xi_1 & w_2\xi_2 & w_3\xi_3 \end{bmatrix} \tag{4.35}$$

and

$$b \equiv \begin{bmatrix} \sum_{i=1}^{I=16} w_{\text{fine},i}\Psi_{\text{fine},i} \\ \sum_{i=1}^{I=16} w_{\text{fine},i}\mu_{\text{fine},i}\Psi_{\text{fine},i} \\ \sum_{i=1}^{I=16} w_{\text{fine},i}\eta_{\text{fine},i}\Psi_{\text{fine},i} \\ \sum_{i=1}^{I=16} w_{\text{fine},i}\xi_{\text{fine},i}\Psi_{\text{fine},i} \end{bmatrix} . \tag{4.36}$$

We invert A and solve for Ψ_{coarse} at the four coarse points on the triangle. Unfortunately, this method skews the angular flux and can introduce unphysical negative fluxes.

Conservation of ϕ using linear discontinuous finite element basis functions

Another method is to use the fine triangle linear basis functions on each of the sub-triangles to interpolate the flux value at the coarse point in that sub-triangle and then ensure conservation of the scalar flux across the face. This method maps from the fine region N with 16 directions to the coarse region M with 4 directions using the following equations:

$$\Psi_M(\overrightarrow{\Omega}_m) = \sum_{n=1}^{4} \Psi(\overrightarrow{\Omega}_n) b_n(\overrightarrow{\Omega}_m) \text{ for each sub-triangle } , \tag{4.37}$$

$$F_{\text{conserve}} = \frac{\sum_{n=1}^{16} w_n \Psi_n}{\sum_{m=1}^{4} w_m \Psi_m} , \tag{4.38}$$

and

$$\Psi_M(\overrightarrow{\Omega}_m) \leftarrow F_{\text{conserve}} \Psi_M(\overrightarrow{\Omega}_m) \text{ for } m = 1:4 . \tag{4.39}$$

This algorithm will ensure conservation of the discrete-ordinate scalar flux on a face. However, this scalar flux on the face is an unphysical quantity and does not ensure conservation of particles. Also, interpolating the linear basis functions can introduce negative flux values at the coarse points.

Conservation of current normal to the face using linear discontinuous finite element basis functions

Another method is to use the fine triangle linear basis functions on each of the sub-triangles to interpolate the flux value at the coarse point in that sub-triangle and then ensure conservation of the partial current normal to the face. This method maps from the fine region N with 16 directions to the coarse region M with 4 directions using the following equations:

$$\Psi_M(\overrightarrow{\Omega}_m) = \sum_{n=1}^{4} \Psi(\overrightarrow{\Omega}_n) b_n(\overrightarrow{\Omega}_m) \text{ for each sub-triangle } , \tag{4.40}$$

$$F_{\text{conserve}} = \frac{\sum_{n=1}^{16} (\overrightarrow{\Omega}_n \cdot \overrightarrow{n}) w_n \Psi_n}{\sum_{m=1}^{4} (\overrightarrow{\Omega}_m \cdot \overrightarrow{n}) w_m \Psi_m} , \tag{4.41}$$

and

$$\Psi_M(\overrightarrow{\Omega}_m) \leftarrow F_{\text{conserve}} \Psi_M(\overrightarrow{\Omega}_m) \text{ for } m = 1:4 . \tag{4.42}$$

This algorithm will ensure conservation of the partial current normal to the face, which ensures particle conservation. However, interpolating the linear basis functions can introduce negative flux values at the coarse points.

Even though we are not extrapolating in this mapping, it is possible to introduce negative flux values. The linear basis functions have regions of the triangle where their values are negative. If the angular flux associated with one basis functions was positive and the other three basis function's angular fluxes were zero, the basis function associated with the non-zero flux would interpolate to a negative value in certain regions on the triangle.

Conservation of current normal to the face using constant discontinuous finite element functions on each sub-triangle

The last method we consider is to use conservation of current normal to the face on each sub-triangle. This makes each coarse point equal to

$$\Psi_{\text{coarse,j}} = \frac{\sum_{i=1}^{I=4} \Psi_{\text{fine},i} (\overrightarrow{\Omega}_{\text{fine},i} \cdot \overrightarrow{n}) w_{\text{fine},i}}{(\overrightarrow{\Omega}_{\text{coarse,j}} \cdot \overrightarrow{n}) w_{\text{coarse,j}}} \quad \text{for} \quad j = 1:4\;. \tag{4.43}$$

This method loses detailed information about the variation of the angular flux within a sub-triangle but will not introduce negative fluxes if the weights are positive and will still conserve partial current across the face.

Implementation

We chose to implement conservation of partial current normal to the face using linear finite element basis functions. This high order method conserves particles and still uses the finite element basis functions. However, this method will introduce negative fluxes in some situations. If negative fluxes are introduced, we move to the method of conserving partial current normal to the face on each sub-triangle. Again, the concept of moving from a high order interpolation scheme to a low order scheme if negativities are introduced is not new.

4.7.3 Same Level Mapping

If the same four directions are contained in each quadrature set, we simply do a one-to-one mapping to determine the angular flux in the mapped quadrature. Mapping from region N with 4 directions to region M with 4 directions takes the form of

$$\Psi_M(\overrightarrow{\Omega}_m) = \Psi_N(\overrightarrow{\Omega}_n) \text{ for } \overrightarrow{\Omega}_m = \overrightarrow{\Omega}_n \text{ for } m = 1:4\ . \tag{4.44}$$

4.8 Method for Determining if Coarsening Is Necessary

We specify that coarsening of triangles can only occur on a parent triangle. Each parent triangle contains four daughter triangles. In order for coarsening to occur, all four daughter triangles in a parent triangle must be determined to be unnecessary. The basic steps for coarsening on a parent triangle can be seen in Table 4.2. Because the four coarse (parent) points are not necessarily in the fine (daughter) quadrature set, we must determine appropriate angular flux values to use at the parent points.

Table 4.2
Coarsening steps for the triangle.

1	Use basis functions on parent triangle to determine Ψ_{interp} at 16 ordinates
2	Compare Ψ_{interp} and Ψ_{calc}
3	If on all faces and for all Ψ's, the values are “close enough,” remove the 16 daughter points from the quadrature set and add the 4 parent points to the quadrature set

It would be cumbersome to initiate a new transport sweep to obtain these values, and it should not be necessary since there will always be daughter points that are very close to the parent points. We therefore obtain the parent-point values by using the fine linear basis functions on each sub-triangle to interpolate the coarse Ψ as follows:

$$\Psi_{\text{coarse},j} = \sum_{i=1}^{I=4} b_{\text{fine},i}(\overrightarrow{\Omega}_{\text{coarse},j})\Psi_{\text{fine},i} \text{ for } j = 1:4 \; . \tag{4.45}$$

We then use the coarse basis functions with the calculated coarse Ψs to determine the interpolated values at the 16 fine directions:

$$\Psi_{\text{interp}}(\overrightarrow{\Omega}_m) = \sum_{i=1}^{I=4} b_{\text{coarse},i}(\Omega_m)\Psi_{\text{coarse},i} \text{ for } m = 1:16 \; . \tag{4.46}$$

These values, interpolated from parent-point values, are then compared against the actual values (denoted Ψ_{calc}). If they are within the specified tolerance, then the coarse (parent) points are considered sufficient and coarsening is permitted.

4.9 Method for Determining if Refinement Is Necessary

The basic steps for testing for refinement on each triangle can be seen in Table 4.3 and graphically in Figure 4.3. We provide below a more detailed description of each step in our refinement process.

4.9.1 Solving Ψ_{calc}

We execute a transport sweep to determine the angular fluxes at the test ordinates on all faces of a given region using the latest scattering and fixed source as well as the same boundary conditions in the previously executed transport sweep. Note that the boundary conditions for a region include exiting fluxes from neighboring regions, calculated during the previous iteration.

4.9.2 Determination of $\Psi_{\text{interpolated}}$

We use the basis functions on the parent triangle to determine an interpolated angular flux at the test (daughter) ordinates. There are four basis functions on the triangle and the interpolated angular flux on each face for each of the directions is shown in Equation 4.47.

$$\Psi_{\text{interpolated}}(\overrightarrow{\Omega}_m) = \sum_{i=1}^{I=4} b_{\text{coarse},i}(\overrightarrow{\Omega}_m)\Psi_{\text{coarse},i} \text{ for } m = 1 : 16 \; . \tag{4.47}$$

4.10 Comparison of Angular Fluxes

We compare the angular fluxes on each face using a user-defined point-wise error tolerance, ϵ_{PW}, and a user-defined octant current tolerance, ϵ_{OC}. ϵ_{OC} is used to ensure that over-adaptation does not occur if the angular fluxes are dramatically

Table 4.3
Refinement steps for the triangle.

1	Solve Ψ_{calc} at test ordinates (candidate daughter points) using scattering source and boundary conditions
2	Use basis functions of parent triangle to determine Ψ_{interp} at candidate daughter ordinates
3a	If Ψ_{calc} and Ψ_{interp} are “close enough” for each of the test points on each cell face on the boundary of the quadrature region, do nothing
3b	Else, add all sixteen new daughter points to the quadrature set Remove four coarse (parent) points from the quadrature set Solve four basis functions on each sub-triangle Solve new weights for the four points in each sub-triangle Sub-triangles are now defined as standard triangles

smaller in one section of the angular domain. We define the test direction difference as

$$\Delta\Psi_{\text{test}} = |\Psi_{\text{calc}} - \Psi_{\text{interp}}|_{\text{test}} \ . \tag{4.48}$$

In order to refine, we must pass BOTH the following tests for ANY test ordinate on ANY face

$$\Delta\Psi_{\text{test}} > \epsilon_{PW}\Psi_{\text{calc}} \tag{4.49a}$$

and

$$\Delta\Psi_{\text{test}} > \epsilon_{OC}\frac{\bar{j}_{\text{octant}}}{(\overrightarrow{\Omega}_{\text{test}} \cdot \overrightarrow{n})} \ . \tag{4.49b}$$

We define $\bar{j}_{\text{octant}}$ for a given octant as

$$\bar{j}_{\text{octant}} \equiv \frac{\sum_{m=1}^{M} w_m(\overrightarrow{\Omega}_m \cdot \overrightarrow{n})\Psi_m}{\sum_{m=1}^{M} w_m} \ , \tag{4.50}$$

where the directions $m = 1..M$ are all of the directions in the same octant as the test direction. We emphasize that refinement will take place even if only one test direction on one cell face satisfies the inequalities defined above.

In order to coarsen, we must pass BOTH the following tests for ALL test ordinates on ALL faces

$$\Delta\Psi_{\text{test}} \leq \epsilon_{PW}\Psi_{\text{calc}} \tag{4.51a}$$

and

$$\Delta\Psi_{\text{test}} \leq \epsilon_{OC}\frac{\bar{j}_{\text{octant}}}{(\overrightarrow{\Omega}_{\text{test}} \cdot \overrightarrow{n})} \ . \tag{4.51b}$$

These adaptive tolerances relate to the cell-average scalar flux in the following manner. If we insist ϵ_{OC} is 1%, we are insisting that the addition of more directions will not impact the surface flow rate in the specific octant by more than 1%. The region's particle balance equation is

$$\begin{aligned}\sum_{f=1}^{FACES} (J_{f,\text{exit}})A_f + \sigma_a\Phi_{\text{reg}}V_{\text{reg}} = \\ \sum_{f=1}^{FACES} (J_{f,\text{inc}})A_f + Q_{\text{Fixed,reg}}V_{\text{reg}} + Q_{\text{Fission,reg}}V_{\text{reg}}\end{aligned} \tag{4.52}$$

where A_f is the area of the face, V_{reg} is the volume of the region, $J_{f,\text{exit}}$ is the particle flow rate per unit area leaving the region through face f, Φ_{reg} is the scalar flux in the region, $J_{f,\text{exit}}$ is the particle flow rate per unit area entering the region through face f, $Q_{\text{Fixed,reg}}$ is the fixed source in the region, and $Q_{\text{Fission,reg}}$ is the fission source in the region. If we solve for Φ_{reg}, we see:

$$\Phi_{\text{reg}} = \frac{\sum_{f=1}^{FACES} (J_{f,\text{inc}} - J_{f,\text{exit}}) A_{\text{f}} + Q_{\text{Fixed,reg}} + Q_{\text{Fission,reg}}}{\sigma_a V_{\text{reg}}} . \tag{4.53}$$

4.11 Updating the Quadrature Set

When refinement or coarsening is determined to be necessary, the quadrature set must be updated. If refinement needs to occur, we add the sixteen new directions to the quadrature set and remove the four old directions from the set. If coarsening needs to occur, we remove the sixteen directions from the quadrature set and add the four new directions to the set.

4.12 Implementation of Method

Our method was implemented in the Knolls Atomic Power Laboratory's slice-balance neutral-particle transport code, Jaguar [24] [25] [26].

4.12.1 Jaguar

Jaguar is a three-dimensional neutral-particle transport code. It solves the neutral-particle transport equation using the Slice Balance Approach (SBA) [2] [3] [4]. It is written in C++ with certain subroutines written in FORTRAN. It is uses an arbitrary polygonal mesh, multiple groups in energy, and discrete ordinates in direction. It is currently not time dependent; our version is not parallel. The code has the ability to define spatial quadrature regions made up of cells. All cells in given region have the same quadrature set. For description, we assume that the code has a vector of angular flux values, $\overrightarrow{V}_{\Psi}$, a vector of scalar flux values, $\overrightarrow{V}_{\Phi}$, and a vector of discrete ordinates, $\overrightarrow{V}_{\text{quadrature}}$ for each spatial quadrature region, which is not energy dependent.

4.12.2 Code Structure and Implementation

We have created an adaptive quadrature class. This class contains a tree structure of "triangles." Each triangle has data associated with it, including geometric descriptions, basis functions, weights, and knowledge of which directions in the $\overrightarrow{V}_{\text{quadrature}}$ belong to the triangle. Because Jaguar only has one quadrature set for each region regardless of energy, we test for refinement and coarsening only in the highest energy group. We chose the first group because the angular flux in the group tends to be the most peaky and unsmooth. In general neutronics problems, as particles slow down their angular dependence becomes more smooth.

Our algorithm can be outlined as follows:

- Input:
  - problem description including geometry, cross-sections, and boundary conditions;
  - initial quadrature set;
  - adaptive user-defined constants including number of sweeps between testing, how many sweeps to perform before the first adaptation sweep, maximum number of adaptation tests, and adaptive tolerances;
  - scalar-flux convergence criteria.
- Non-Adaptive Group Transport Sweep:
  - uses the most recent quadrature set, scattering source, and fixed source;
  - uses quadrature mapping algorithms to conserve partial current normal to the boundary faces;

  - performs SBA transport sweeps, which solves for $\overrightarrow{V}_{\Psi}$ on the region boundaries and $\overrightarrow{V}_{\Phi}$ on the cells in inner iterations;
  - tests if $\overrightarrow{V}_{\Psi}$ and $\overrightarrow{V}_{\Phi}$ of this iteration satisfy the scalar-flux convergence requirements.

- Adaptive Group Transport Sweep:
  - creates a list of test directions for each region;
  - uses the last boundary conditions, scattering source, and fixed source to solve an SBA transport sweep to determine the angular flux values on the boundary faces of the region for these test directions;
  - compares the swept values on the boundary faces to the interpolated values on the faces using the basis functions described in Section 4.5.

- Updating Scalar Flux, Angular Flux, and Quadrature Set ($\overrightarrow{V}_{\Phi}$, $\overrightarrow{V}_{\Psi}$, $\overrightarrow{V}_{\text{quadrature}}$)
  - If Refinement criteria is not satisfied, add directions to $\overrightarrow{V}_{\text{quadrature}}$ and update $\overrightarrow{V}_{\Phi}$ and $\overrightarrow{V}_{\Psi}$;
  - If Coarsening criteria is satisfied, remove directions from $\overrightarrow{V}_{\text{quadrature}}$ and update $\overrightarrow{V}_{\Phi}$ and $\overrightarrow{V}_{\Psi}$.

4.12.3 Memory Requirements

Because our test code is meant to evaluate algorithms (not to solve a large number of large problems), we did not constrain ourselves to make highly efficient use of computer memory. However, the current general memory requirements for a triangle are as follows:

- three sets of two variables to describe the corners of the triangle in u, v space;
- four pointers to the directions in the $\overrightarrow{V}_{\text{quadrature}}$ associated with this triangle;

- four pointers to the sub-triangle objects;
- one pointer to the parent triangle object.

On the triangle, we also must employ methods to

- Determine the weights and basis functions associated with the triangle, which includes integrating a function of the form $c_0 + c_1\mu + c_2\eta + c_3\xi$ over a spherical triangle on the unit sphere;S
- Determine the coordinates (μ, η, ξ) of the associated directions;
- Determine the values of interpolated angular fluxes for the triangle;
- Determine the number of refinements it took to get to the triangle (depth);
- Map the triangle's angular flux values to its sub-triangle's angular flux values;
- Map the triangle's angular flux values to its parent-triangle's angular flux values;
- Find the same triangle or its parents in a different region's quadrature set.

Generally, we must also be able to

- Add directions to $\overrightarrow{V}_{\text{quadrature}}$ and values associated with those directions to $\overrightarrow{V}_{\Psi}$;
- Remove directions from $\overrightarrow{V}_{\text{quadrature}}$ and values associated with those directions from $\overrightarrow{V}_{\Psi}$;
- Sweep a region for a given number of test directions which includes finding correct incoming angular fluxes, finding correct scattering source and fixed source values, and performing the SBA on these directions;
- Compare swept values to interpolated values;
- Determine if Coarsening or Refinement is necessary.

4.13 Method Drawbacks

While this method has the capability to give accurate results with fewer directions and control the error introduced by the discrete ordinate method, we compromised in some areas.

- Because we test for refinement only on the region boundary, certain self-shielding effects and spatial mesh effects that cause peaked interior angular fluxes and smooth boundary angular fluxes would not be captured in our adaptation scheme.
- If the tests for refinement occur only after n sweeps, we could pass the scattering-source convergence test before the refinement test was activated. This would create a false sense of accuracy.
- If the refinement adds directions to the quadrature and the associated flux values to the solution vector, we could meet the user-defined scattering-source convergence tolerance (normally on the order of 1e-8) without ensuring that the angular tolerance was satisfied.
- If the center point method is employed, the weights at the centers of the triangles are reduced in such a way that they can become negative. This can cause unphysical negative fluxes and major difficulties in the mapping scheme.
- From a code analysis perspective, there is a large amount of data associated with each triangle compared to the relatively little data associated with the standard quadrature set. However, because adaptation occurs in the same hierarchical manner that originated from the unit sphere, all adaptations could be performed through a given number of refinement steps and stored in a different file. This would allow the code to use previously computed basis values, weight values, and ordinates without having to recompute these values each time adaptation was determined to be necessary.

5. RESULTS

In this section, we discuss the ability of our quadrature sets to integrate different functions in the direction cosines, μ, η and ξ. We also present results using our different quadrature sets. We begin with simple problems and move to more challenging problems.

5.1 Order of Accuracy of New Quadrature Sets

As discussed in the previous section, four new quadrature sets have been developed. The first is the centroid set, discussed in Section 4.6.1, which requires each point in a triangle to be at the centroid of its sub triangle (LDFE-Center). This set begins to introduce negative weights after only a few refinement steps. The next is the Θ-weighted quadrature set, discussed in Section 4.6.2, which ensures that the center weight and center sub-triangle's surface area are equal (LDFE-Θ). It establishes a Θ ratio which defines the location of the other three points in the triangle. The third set is the L weighted quadrature set, discussed in Section 4.6.3, which also ensures that the center weight and center sub-triangle's surface area was equal (LDFE-L). It establishes a different L ratio which defines the location of the other three points in the triangle. The final quadrature set is the same-surface-area set, discussed in Section 4.6.4, which ensures that all weights (not just the center weights) are equal to their associated sub-triangles' surface areas (LDFE-SA). Because the quadrature sets are based on the integration of linear functions of the cosines, we expect the quadrature sets to exactly integrate linear combinations of the direction cosines, μ, η, and ξ. However, the ability of these sets to integrate higher order polynomials in the cosines needs to be explored.

5.1.1 Uniform Quadrature Sets

We first examine uniform quadrature sets without localized adaptation. Because the quadrature sets do not treat any direction cosine differently from any other, each octant is a rotation of the first octant. Because of this symmetry, the integration analysis need be performed only on the first octant and only for one permutation of each polynomial in the direction cosines (i.e., $\mu^2\eta$ and $\eta^2\xi$ give the same result). Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6 show the relative error of the integrations of $\mu\eta$, μ^2, μ^3, μ^4, $\mu^3\eta\xi$, and $\mu^2\eta^2\xi^2$ versus the angular mesh length for the uniform quadrature sets. This angular mesh length (h) can be thought of as the "h" term normally seen in a truncation-error analysis. In this case, because the direction domain is two-dimensional, the mesh length is proportional to $\frac{1}{\sqrt{\text{number of directions}}}$. However, if there are multiple cells with different quadrature sets, we use the following equation to get the cell average angular mesh length for a problem:

$$h_{\text{cell average}} \equiv \frac{1}{\sqrt{\frac{\sum_{c=1}^{numCells} N_{\text{directions,c}}}{numCells}}} \, . \tag{5.1}$$

We can also define an angular mesh length for a given region, which computes $h_{\text{cell average}}$ for just the cells in a given region. Because all cells in a given region have the same number of directions, this reduces the region's angular mesh length to

$$h_{\text{region}} \equiv \frac{1}{\sqrt{\text{number of directions in a cell}}} \, . \tag{5.2}$$

The uniform sets exactly integrate μ^2, η^2, and ξ^2 but fail to exactly integrate $\mu\eta$ cross-terms or any moments higher than μ^2. However, as the quadrature sets mesh length is reduced by a given factor, the error goes down by approximately that factor to the fourth power for all polynomial functions in the cosines that we tested, which include functions up through sixth order. Thus we call these uniform sets fourth-order accurate for integration of high-order polynomials in the cosines.

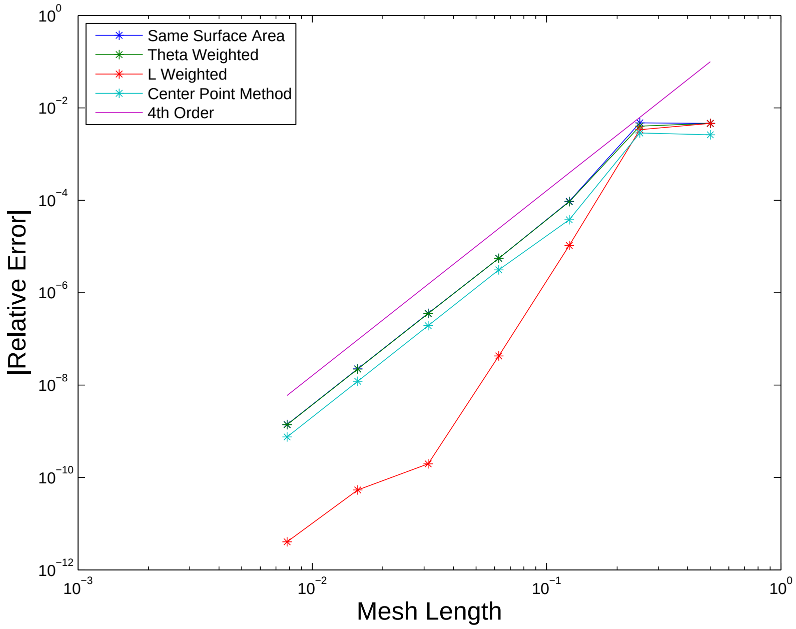

Fig. 5.1. Relative error of $\mu\eta$ integration on the first octant versus mesh length for the uniform quadrature sets.

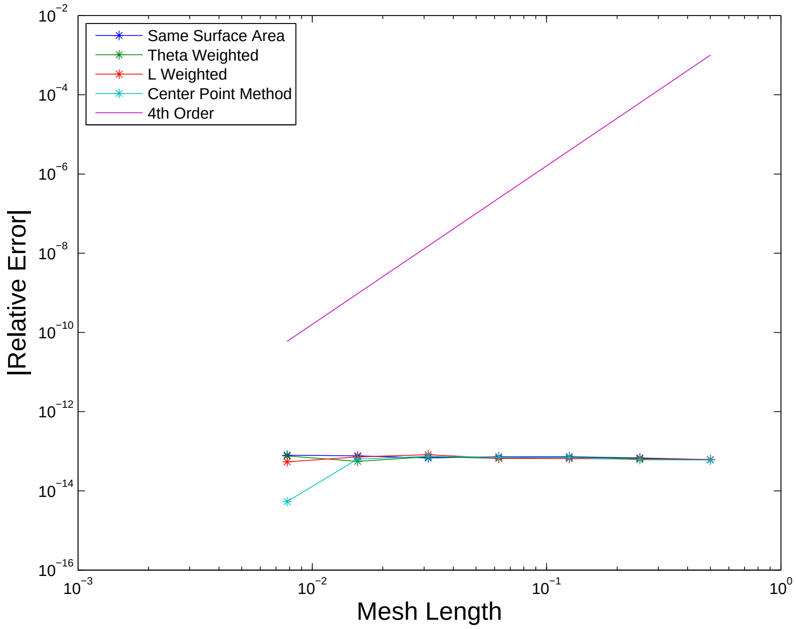

Fig. 5.2. Relative error of μ^2 integration on the first octant versus mesh length for the uniform quadrature sets.

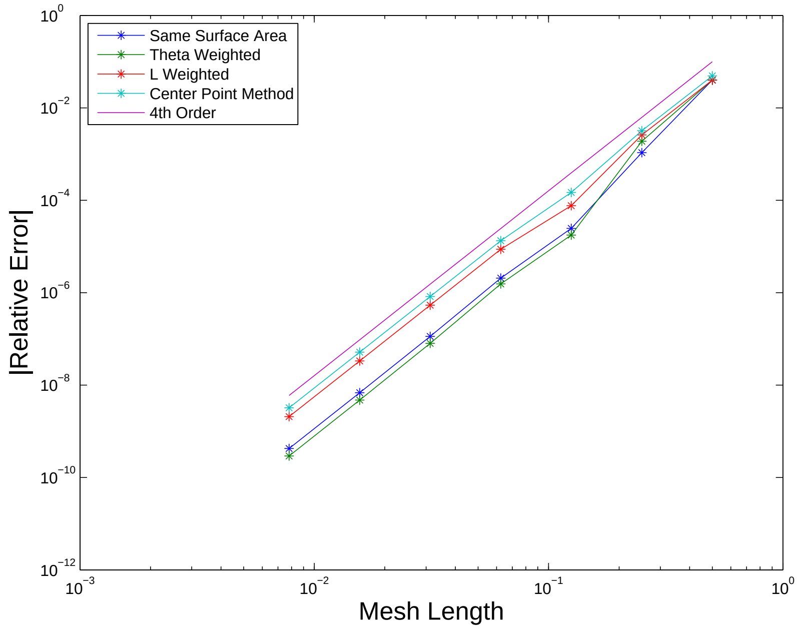

Fig. 5.3. Relative error of μ^3 integration on the first octant versus mesh length for the uniform quadrature sets.

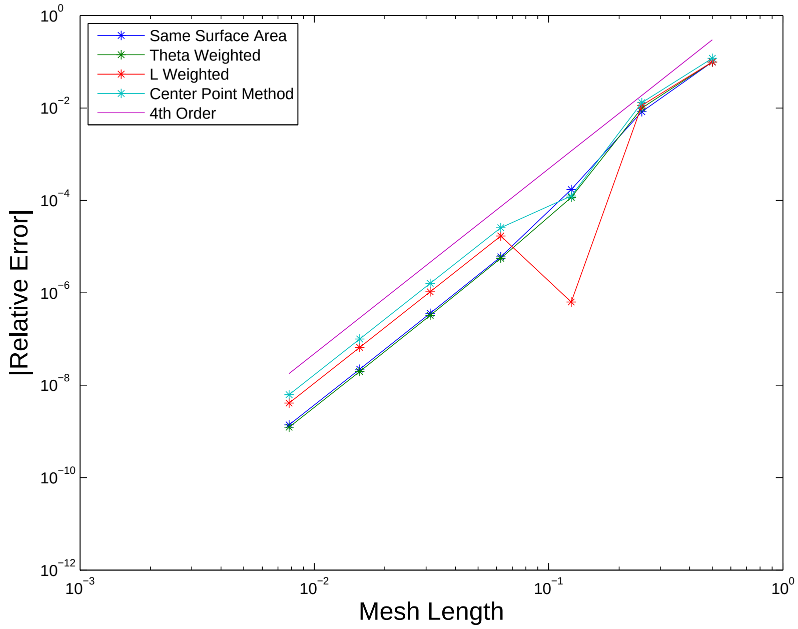

Fig. 5.4. Relative error of μ^4 integration on the first octant versus mesh length for the uniform quadrature sets.

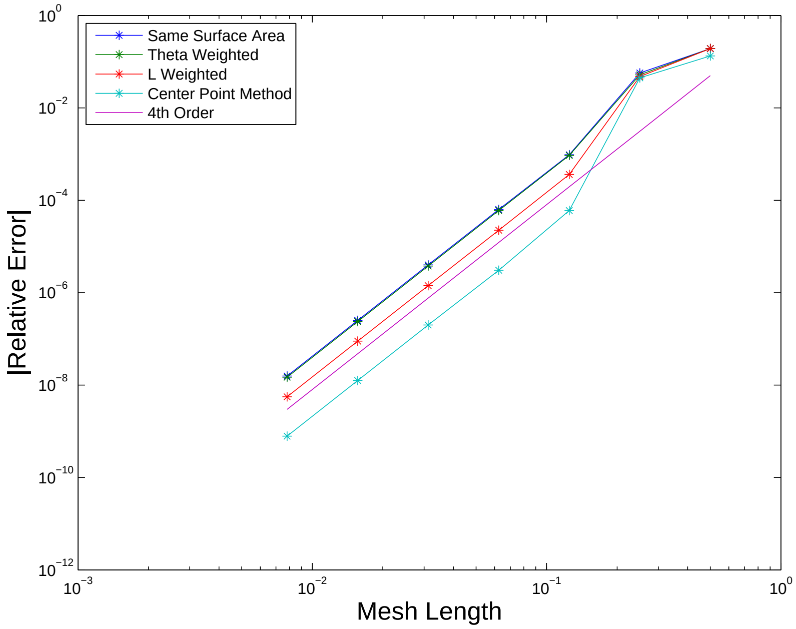

Fig. 5.5. Relative error of $\mu^3\eta\xi$ integration on the first octant versus mesh length for the uniform quadrature sets.

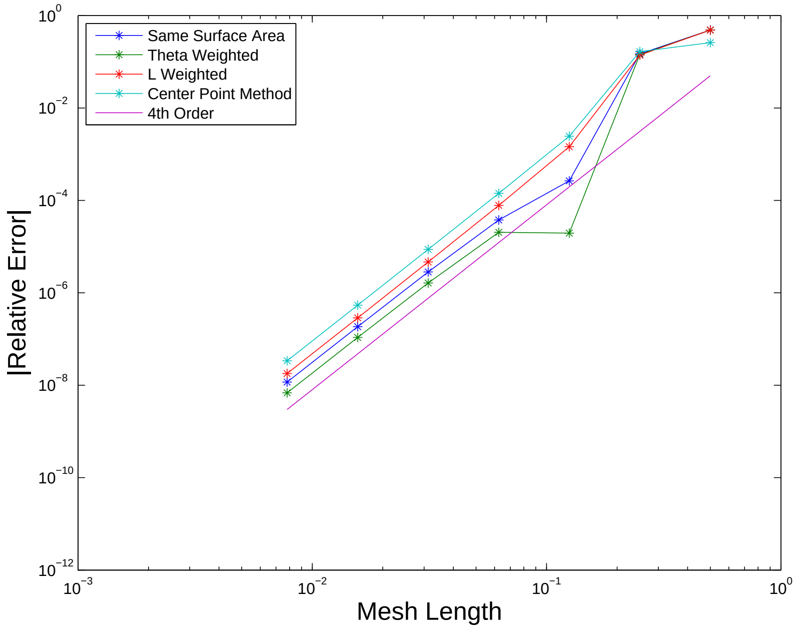

Fig. 5.6. Relative error of $\mu^2\eta^2\xi^2$ integration on the first octant versus mesh length for the uniform quadrature sets.

5.1.2 Locally-refined Quadrature Sets

Once local refinement occurs, the behavior of the quadrature sets' integrations of high-order polynomials in the cosines changes dramatically. Our locally refined sets are produced using a simple one-region, one-cell, fixed-source, purely-absorbing, one-group problem as illustrated in Figure 5.7. A more detailed description of this problem can be seen in Section 5.2. The adaptive sets use a Piece-Wise tolerance of 0.0 and a Octant-Current tolerance of 0.001. We analyze the locally refined set after each refinement iteration. The exact integration of μ^2, η^2, and ξ^2 is not guaranteed and the fourth-order reduction in error as a function of mesh length is lost. Figures 5.8, 5.9, 5.10, 5.11, 5.12, and 5.13 show the error in integrating $\mu\eta$, μ^2, μ^3, μ^4, $\mu^3\eta\xi$, and $\mu^2\eta^2\xi^2$ versus the angular mesh length using a locally refined quadrature set that approaches the uniform sets. For this problem, the locally refined quadrature set is almost uniform; this allows integration of μ^2 to continue to be almost exact. In fact, each octant is a rotation of the first octant. The almost uniform quadrature set with 808 points, which corresponds to a mesh length of 3.52e-2, can be seen in Figure 5.14.

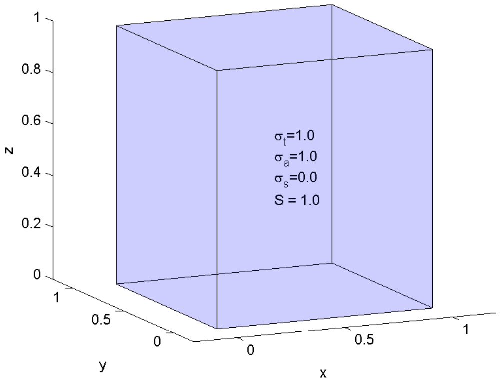

Fig. 5.7. Test problem with one-region, one-cell, one-group, fixed source, isotropic scattering, vacuum boundary conditions, and a pure absorber.

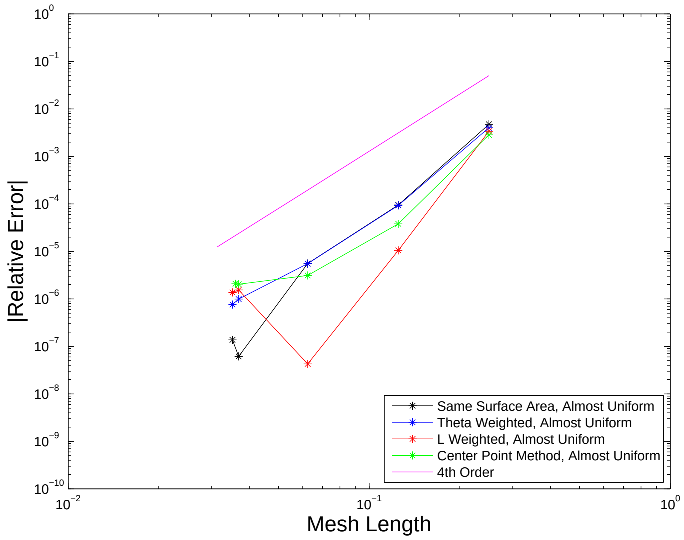

Fig. 5.8. Relative error of $\mu\eta$ integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

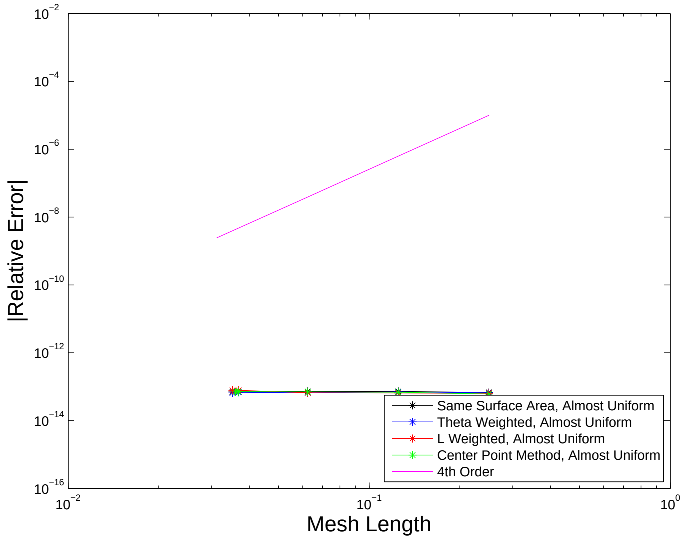

Fig. 5.9. Relative error of μ^2 integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

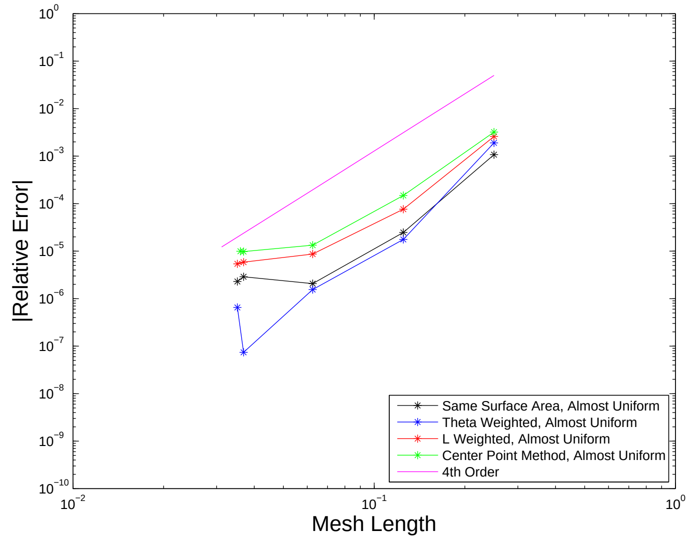

Fig. 5.10. Relative error of μ^3 integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

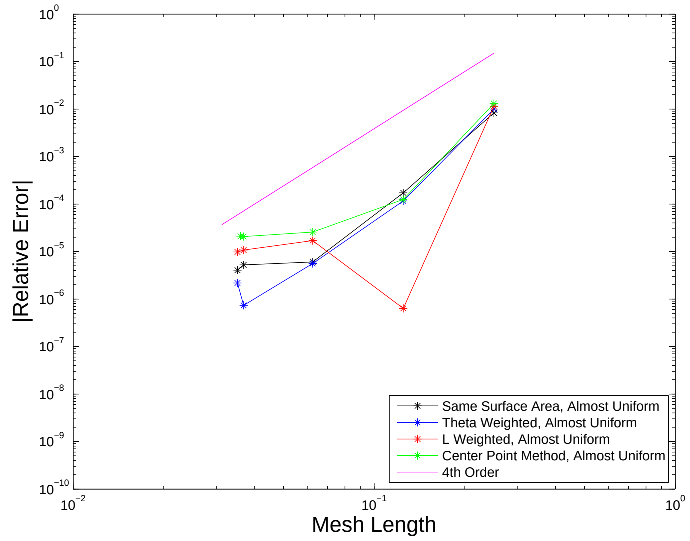

Fig. 5.11. Relative error of μ^4 integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

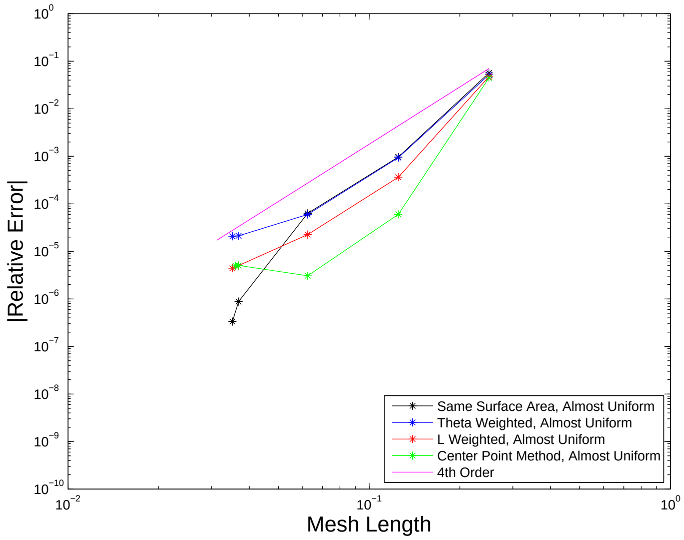

Fig. 5.12. Relative error of $\mu^3\eta\xi$ integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

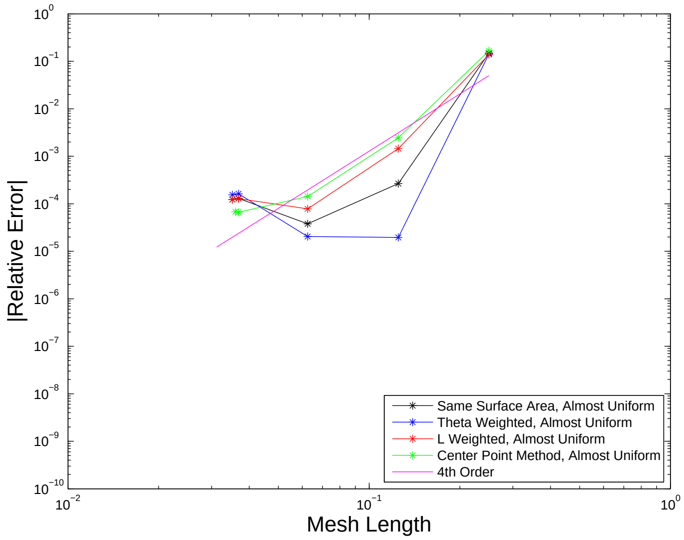

Fig. 5.13. Relative error of $\mu^2\eta^2\xi^2$ integration on the first octant versus mesh length in locally refined quadrature sets for a simple one-region, one-cell problem.

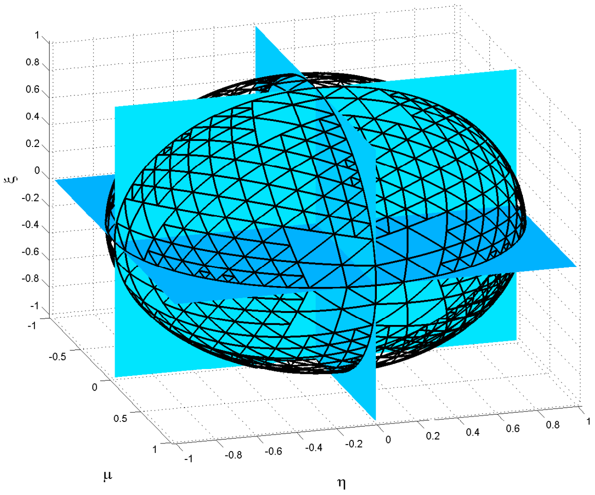

Fig. 5.14. Almost uniform quadrature set for 808 directions for the one-region problem.

The three points with the largest angular mesh length in these figures are actually truly uniform sets. However, the two points with the smallest angular mesh length are locally refined sets. From these points, one can easily see the polynomial-integration behavior changing.

We now look at a problem where the angular flux is more peaked. It is a three-cell, one-region, fixed-source in the first cell, purely-absorbing, one-group problem illustrated in Figure 5.15. We give a more detailed description of this problem in Section 5.4. Our locally refined quadrature sets are produced using a Piece-Wise tolerance of 0.0 and a Octant-Current tolerance of 0.3. We show the integration properties of the quadrature set after each refinement step. Figures 5.16, 5.17, 5.18, 5.19, 5.20, and 5.21 show the error in integration of $\mu\eta$, μ^2, μ^3, ,μ^4, $\mu^3\eta\xi$, and $\mu^2\eta^2\xi^2$ using a locally refined quadrature set that is highly refined in a localized direction. The highly localized quadrature set for 1012 directions, which corresponds to a mesh length of 3.14e$-$2, can be seen in Figure 5.22.

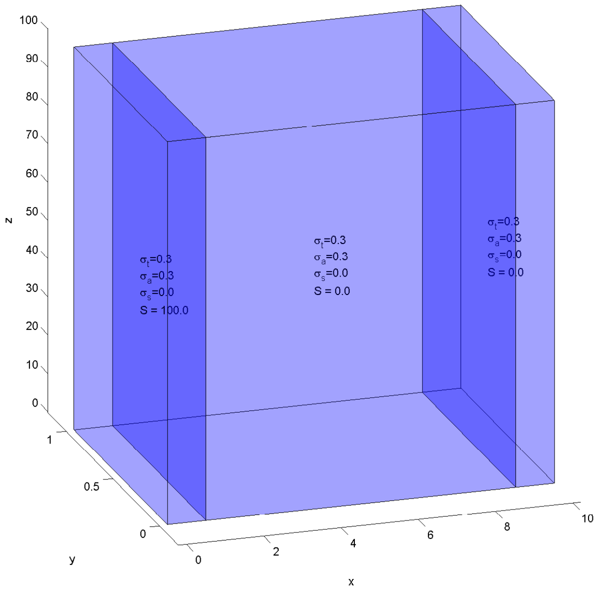

Fig. 5.15. Three-cell, one-group, pure absorbing problem with vacuum boundary conditions.

These figures show that there is no reduction in polynomial-integration error when directions are added locally and, in some cases, an increase in error occurs. For integration of higher-order polynomials in the cosines, the locally adapted quadrature sets are not guaranteed to be more accurate as more local refinement occurs, whereas the uniform quadrature sets become much more accurate as refinement occurs. The quadrature sets that approach the uniform sets still retain some ability to decrease the error in integration as the number of directions is increased, but even that cannot be guaranteed, especially for the higher-order polynomials. Moreover, in the highly localized quadrature sets, the integrations can get dramatically worse as directions are added locally. We state simply that as the quadrature sets get more locally refined the integration error of polynomials in the cosines cannot be guaranteed to improve and with highly localized refinement may become substantially worse.

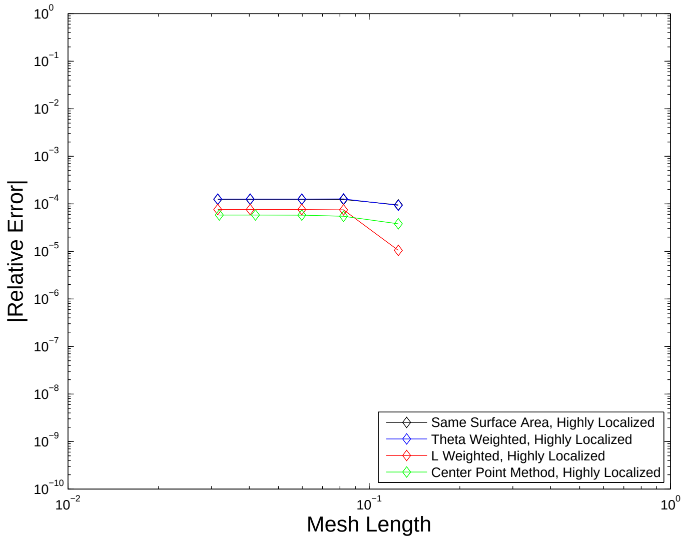

Fig. 5.16. Relative error of $\mu\eta$ integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

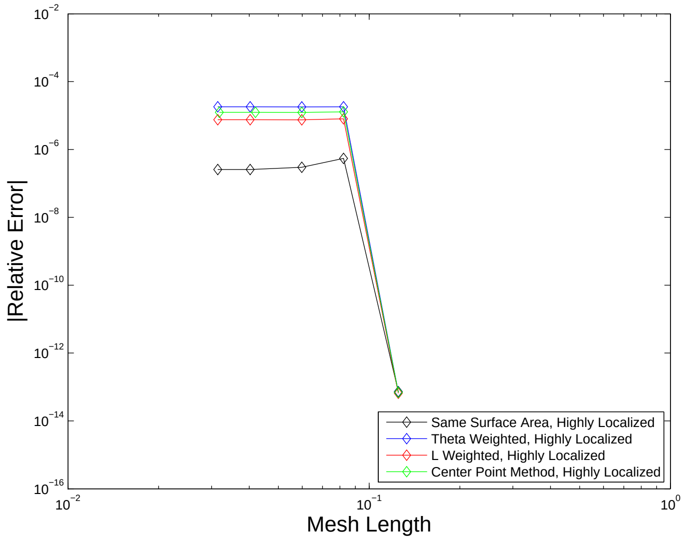

Fig. 5.17. Relative error of μ^2 integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

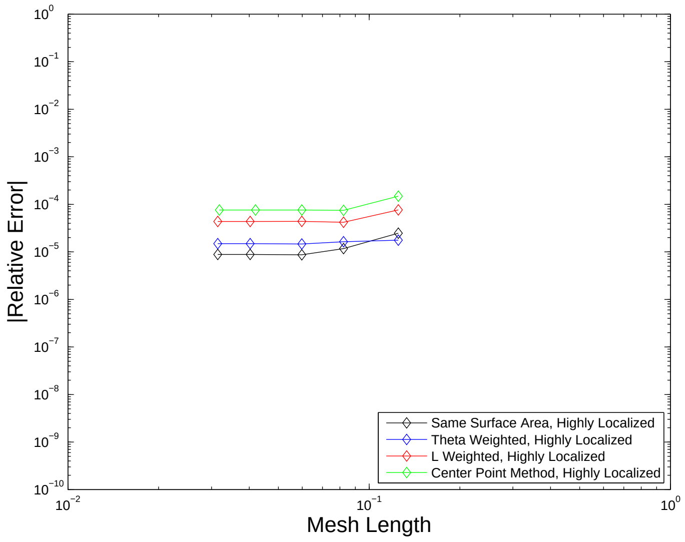

Fig. 5.18. Relative error of μ^3 integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

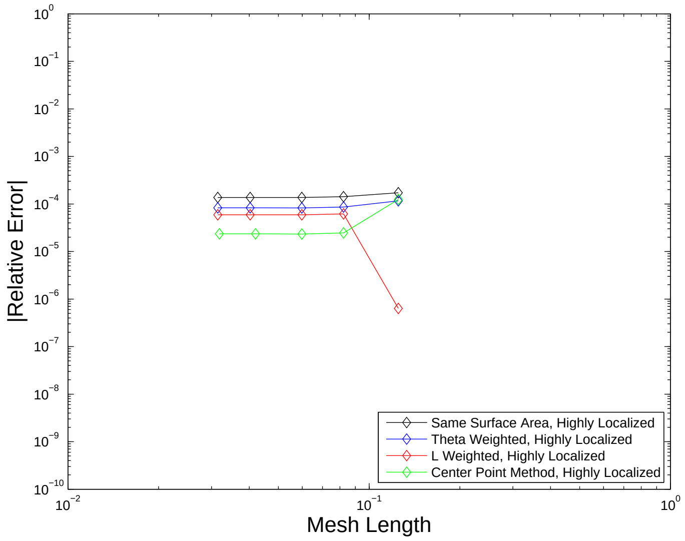

Fig. 5.19. Relative error of μ^4 integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

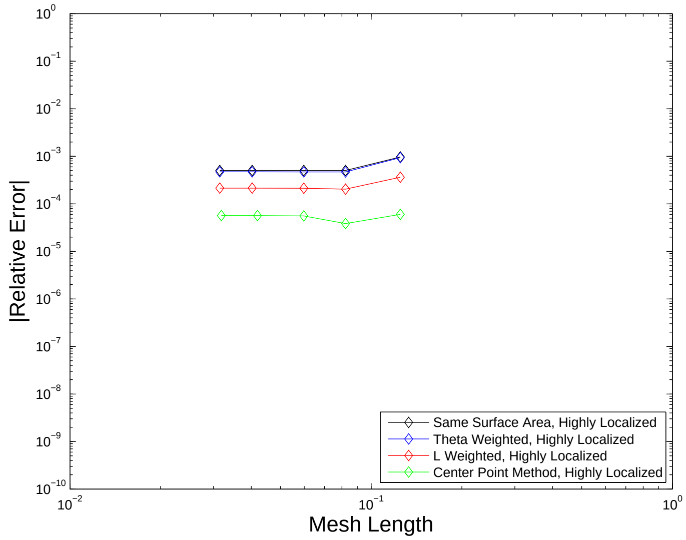

Fig. 5.20. Relative error of $\mu^3\eta\xi$ integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

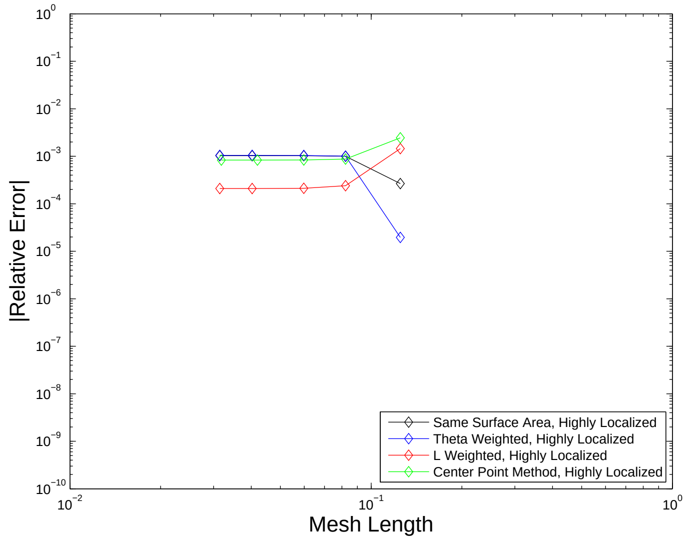

Fig. 5.21. Relative error of $\mu^2\eta^2\xi^2$ integration on the first octant versus mesh length for locally refined quadrature sets for a long one-region, three-cell problem.

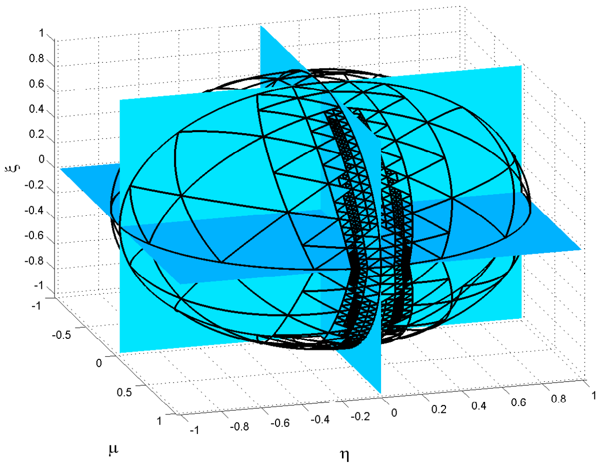

Fig. 5.22. Localized quadrature set for 1012 directions for the one-region, three-cell problem.

5.2 One-region Problem

We turn now from integrating polynomials to solving transport problems (which requires quadrature integrations of angular fluxes). The first problem analyzed was a one quadrature region, one-cell, one-group, isotropic scattering, fixed-source problem. We utilized a step-characteristic spatial method and applied vacuum boundary conditions. The region was a 1 cm × 1 cm × 1 cm cube with a constant isotropic source of 1.0 $\frac{neutron}{cm^3 s{*}ster}$. As shown in Figures 5.7 and 5.23, we initially analyzed a purely absorbing problem with a total macroscopic cross section of $\Sigma_t = 1.0\ cm^{-1}$, which ensures that the cell is 1×1×1 mean-free paths (mfp), and a purely scattering problem with a total macroscopic cross section of $\Sigma_t = 5.0\ cm^{-1}$, which ensures that the cell is 5x5x5 mfps. This simple one-region problem was utilized to debug the adaptation method as well as to guide the adaptive tolerance descriptions. The first step was to ensure that the new quadrature sets (LDFE-Center, LDFE-Θ, LDFE-L, and LDFE-SA) were converging to the same solution as established quadrature sets such as QR, LS, or GC. Once this was established, the reference solution scalar flux

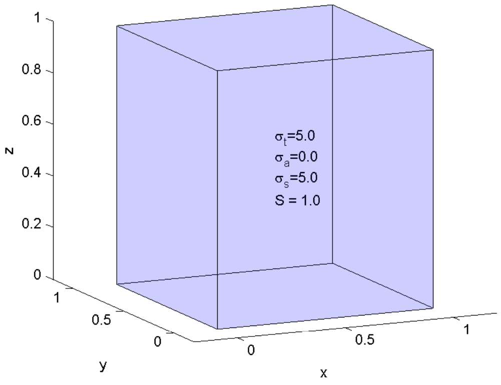

Fig. 5.23. Test problem with one region, one cell, one group, fixed source, isotropic scattering, vacuum boundary conditions, and a pure scatterer.

was defined as the scalar flux from a very fine uniform LDFE-SA set (524288 directions) with a very tight convergence criterion that forced the L2 norm of the residual divided by the L2 norm of the source vector to be less than of 1e-13. As an aside, the purely absorbing problem converges to computer precision in one sweep because the directions are independent of each other.

5.2.1 Pure Absorber

We first compared the uniform quadrature sets (LDFE-SA, LDFE-Θ, LDFE-L, and LDFE-Center) to the Level-Symmetric (LS), Gauss-Chebyshev (GC), Double Gauss-Chebyshev (DGC), and Quadruple Range (QR) sets [1] [15] [12] [13] [14]. The LS sets were run at all orders through 16 and also at order 20. Order 18 and orders greater than 22 yield negative weights. The GC sets were run as “square” product sets from 2x2 through 100x100 and as triangular LS-like sets from order 2 through 300. The DGC sets were also run as “square” product sets from 2x2 through 100x100 and as triangular LS-like sets from order 2 through 300. The QR sets were run as “square” sets from 2x2 through 18x18, as “rectangular” sets from 2x4 through

18x36, and as triangular LS-like sets from order 2 through 36. The four LDFE sets were run for refinement numbers 0 through 7. The initial 0 set contains 32 directions and each set after that contains four times the number of directions as the previous set, i.e: the 0 set contains $8 * 4^1 = 32$ directions, the 1 set contains $8 * 4^2 = 128$ directions, ..., the 7 set contains $8 * 4^8 = 524288$ directions. The relative error in the cell-centered scalar flux for uniform LDFE sets, QR sets, DGC, GC, and LS sets as a function of angular mesh length can be seen in Figure 5.24. (The reference solution was taken as the LDFE-SA solution with 524288 directions.) As evident in

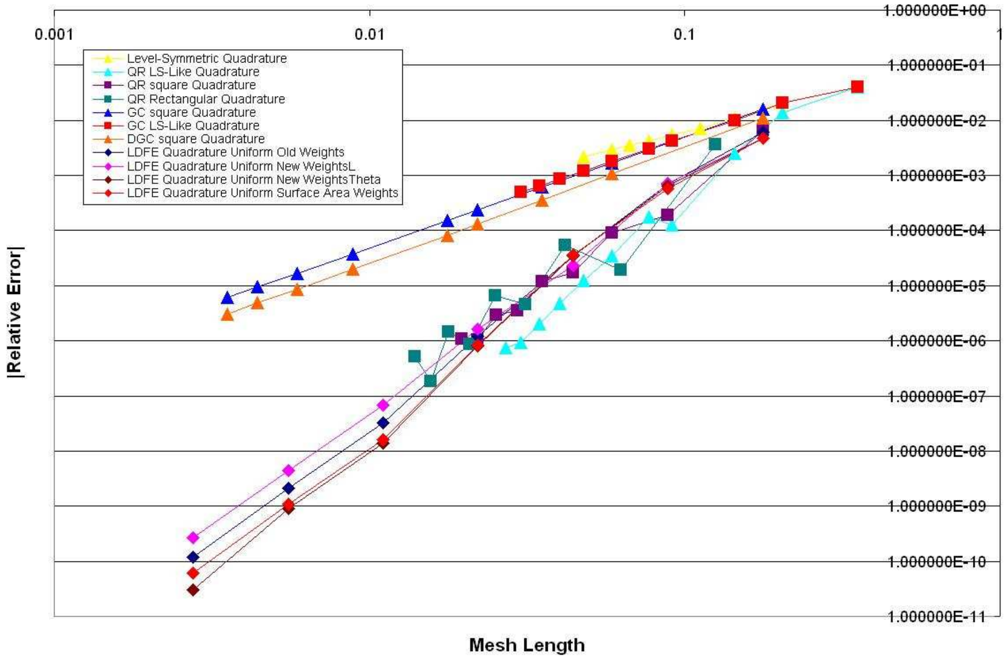

Fig. 5.24. Relative error for LDFE, QR, DGC, GC, and LS uniform quadrature sets versus mesh length for a one-region, one-cell, one-group, purely absorbing, fixed-source problem with vacuum boundary conditions.

Figure 5.24, all LDFE Quadrature sets converge quickly to the reference solution. We computed the order of convergence using the "Microsoft Excel" built-in, least-squares approach to find the best p in the equation $c * h^p$. We used all calculated points for

Table 5.1
Convergence order of uniform quadrature sets' cell-centered scalar fluxes as a function of mesh length for a pure absorbing, one-region problem.

Type of Quadrature	Convergence Order
LDFE Quadrature Uniform Surface Area Weights	4.57
LDFE Quadrature Uniform Old Weights	4.41
LDFE Quadrature Uniform New Weights L	4.12
LDFE Quadrature Uniform New Weights Θ	4.71
Level-Symmetric Quadrature	1.50
QR LS-Like Quadrature	4.52
QR Square Quadrature	3.95
QR Rectangular Quadrature	3.94
GC Square Quadrature	2.01
GC LS-Like Quadrature	1.93
DGC Square Quadrature	2.09
DGC LS-Like Quadrature	1.84

this evaluation. The QR sets and LDFE quadrature sets converge to the correct scalar flux at approximately 4th order, the GC and DGC sets at approximately 2nd order, and the LS sets at approximately 1.5 order as shown in Table 5.1.

We employ tests using two different tolerances when deciding whether or not to refine a quadrature set as discussed in Section 4.10. The Point-Wise Tolerance (PW Tol) compares values using the linear finite element basis functions. The Octant Current Tolerance (OC Tol) compares the normal component of the angular current density ($\overrightarrow{n} \cdot \overrightarrow{\Omega}\Psi$) at the test direction versus the average normal component of the angular current density over the entire octant. In order for refinement to occur, the test points must fail both tolerances. Therefore, setting one of the tolerances equal to zero allows the refinement to be governed solely by the other tolerance. This one-region problem was first analyzed using a OC Tol of 0 and varying the PW Tol from 30% to 0.01% for the LDFE-SA sets, LDFE-L sets, LDFE-Θ sets, and the LDFE-Center. Labels for these results in the figure include "PW Tol." It was then analyzed using a PW Tol of 0 and varying the OC Tol from 30% to 0.01% for each quadrature set. Labels for these results include "Octant Current." The results can be seen in Figure 5.25. Figure 5.25 shows that the locally adaptive sets approach

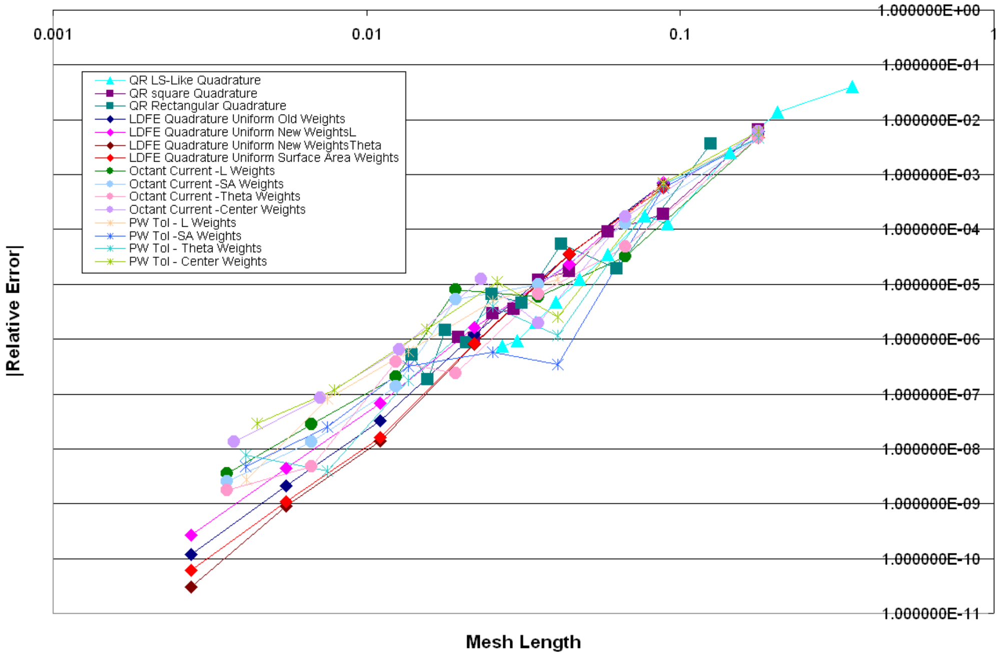

Fig. 5.25. Relative error for QR uniform sets, LDFE-L, LDFE-Θ, LDFE-SA, and LDFE-Center adaptive and uniform quadrature sets versus mesh length for a one-region, one-cell, one-group, purely absorbing, fixed-source problem with vacuum boundary conditions.

Table 5.2
Convergence order of locally refined quadrature sets' cell-centered scalar fluxes as a function of mesh length for a pure absorbing, one-region problem.

Type of Quadrature	Convergence Order
PW Tol - Center Weights	3.37
PW Tol - Θ Weights	3.90
PW Tol - SA Weights	3.74
PW Tol - L Weights	3.70
Octant Current - Center Weights	3.38
Octant Current - Θ Weights	3.88
Octant Current - SA Weights	3.76
Octant Current - L Weights	3.52

the correct solution but are not producing better region-averaged, scalar-flux results than the uniform sets. The order of convergence of each locally adaptive case can be seen in Table 5.2. We see that the locally adaptive sets approach the true solution at orders between 3.3 and 3.9 for each tolerance method. It should be noted that we are able to obtain more accurate solutions with the new sets than is possible with the QR sets, because the QR sets are limited to eighteen polar levels and thirty-seven azimuthal directions in a given octant.

5.2.2 Pure Scatterer

The purely scattering problem was analyzed in a similar fashion. The relative errors in cell-averaged scalar flux for the uniform LDFE quadrature sets, QR sets, DGC sets, GC sets, and the LS sets can be seen in Figure 5.26 and the order of convergence can be seen in Table 5.3. Again, the LDFE uniform sets have approximately 4th-order convergence to the reference solution. In this problem, the QR sets exhibit approximately 5th order convergence.

The performances of LDFE quadrature sets with localized adaptation were also analyzed. The first locally refined sets were generated using a OC Tol of 0.0 and varying the PW Tol from 30% to 0.01% for the LDFE-SA sets, the LDFE-L sets,

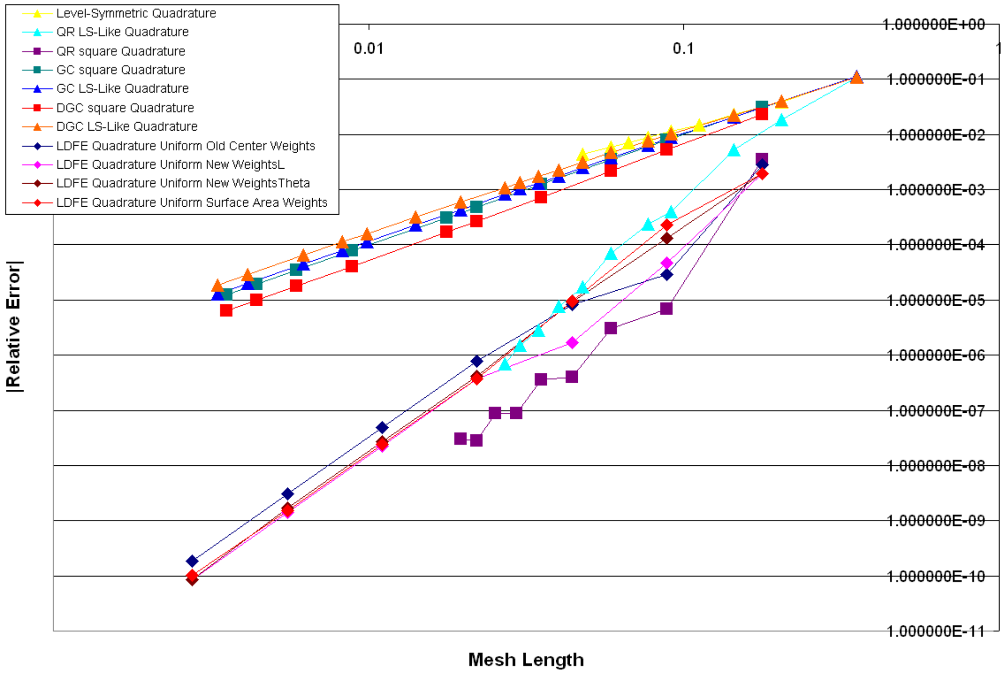

Fig. 5.26. Relative error in the cell-centered scalar flux for LDFE, QR, DGC, GC, and LS uniform quadrature sets versus mesh length for a one-region, one-cell, one-group, purely scattering, fixed-source problem with vacuum boundary conditions.

Table 5.3
Convergence order of uniform quadrature sets' cell-centered scalar fluxes as a function of mesh length for a purely scattering, one-region problem.

Type of Quadrature	Convergence Order
LDFE Quadrature Uniform Surface Area Weights	4.13
LDFE Quadrature Uniform Old Center Weights	3.76
LDFE Quadrature Uniform New WeightsL	3.91
LDFE Quadrature Uniform New WeightsTheta	4.08
Level-Symmetric Quadrature	1.59
QR LS-Like Quadrature	4.76
QR Square Quadrature	5.03
GC Square Quadrature	2.00
GC LS-Like Quadrature	1.95
DGC Square Quadrature	2.09
DGC LS-Like Quadrature	1.86

Table 5.4
Convergence order of locally adaptive quadrature sets' cell-centered scalar fluxes as a function of mesh length for a purely scattering, one-region problem.

Type of Quadrature	Convergence Order
PW Tol - Center Weights	3.68
PW Tol - Θ Weights	3.46
PW Tol - SA Weights	3.54
PW Tol - L Weights	3.57
Octant Current - Center Weights	3.70
Octant Current - Θ Weights	3.67
Octant Current - SA Weights	4.05
Octant Current - L Weights	3.56

the LDFE-Θ sets, and the LDFE-Center sets. Labels for these results in the figure include "PW Tol." The next locally refined sets were generated using a PW Tol of 0 and varying the OC Tol from 30% to 0.01% for each quadrature set. Labels for these results in the figure include "Octant Current." The results can be seen in Figure 5.27 and the order of convergence can be seen in Table 5.4. The localized sets generally agree with the uniform LDFE sets but rarely produce a more accurate solution than the uniform LDFE sets. We hypothesize that this occurs because the locally adaptive sets are less accurate in their integrations of high-order polynomials in the cosines, as we showed in a previous section.

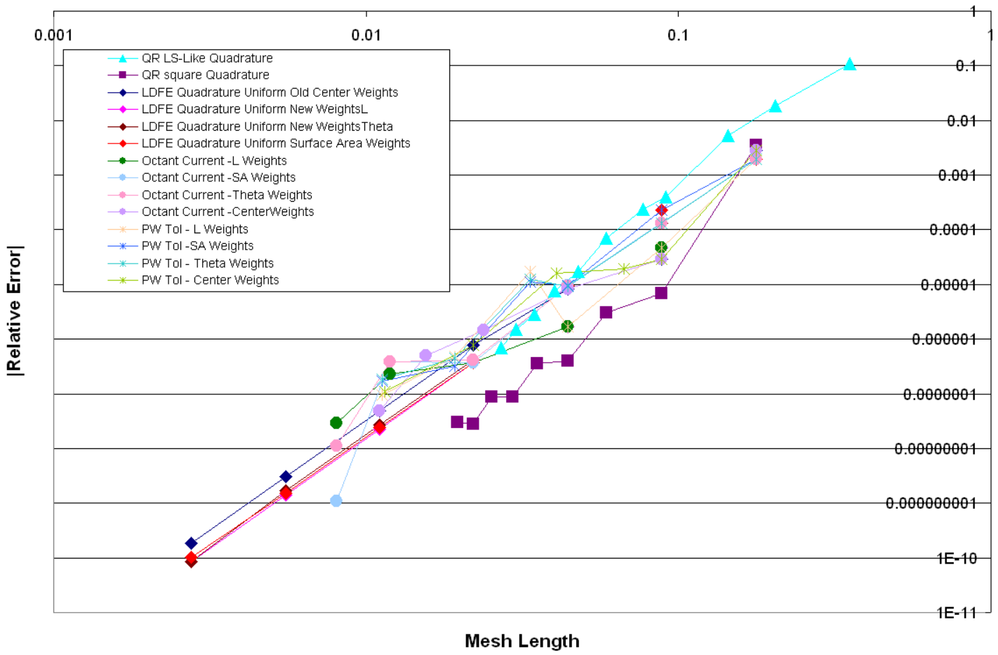

Fig. 5.27. Relative error for QR uniform sets, LDFE-L, LDFE-Θ, LDFE-SA, and LDFE-Center adaptive and uniform quadrature sets versus mesh length for a one-region, one-cell, one-group, purely scattering, fixed-source problem with vacuum boundary conditions.

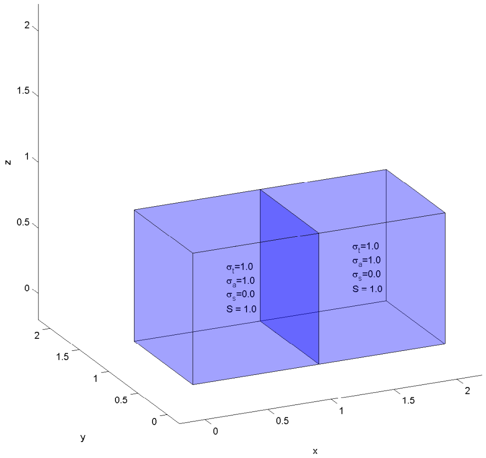

Fig. 5.28. Two-region, one cell per region, one-group, fixed-source, vacuum boundary conditions, purely absorbing problem.

5.3 Two-region Problem

The next problem analyzed was a two-region problem with one cell in each region as illustrated in Figure 5.28. It was a one-group, purely absorbing problem with a constant source in each region and vacuum boundary conditions which was solved using the step-characteristics method. The source was 1.0 $\frac{neutrons}{cm^3 s*ster}$ and the macroscopic total cross section was $\Sigma_t = 1.0\ cm^{-1}$. Each region was a 1 cm × 1 cm × 1 cm cube or a 1 mfp × 1 mfp × 1 mfp cube. This problem was run to test the mapping algorithms developed for these new LDFE quadrature sets. We defined the reference solution to be the uniform LDFE-SA quadrature set solution with 524288 directions.

5.3.1 Uniform Quadrature Sets

The absolute value of the relative errors in the cell-centered scalar flux as a function of angular mesh length for the uniform LDFE sets, the QR sets, the LS sets, the GC sets, and the DGC sets are shown in Figure 5.29. Because all sets have the same quadrature sets in both regions, the mapping on the interior boundary is a simple one-to-one mapping. In reality, this problem is the same as a single region, two-cell problem. The order of convergence is quite similar to the one-region case as shown in Table 5.5. Again, the LDFE sets and the QR sets have 4th-order convergence, the GC and DGC sets have 2nd-order convergence, and the LS sets have a convergence order of about 1.5.

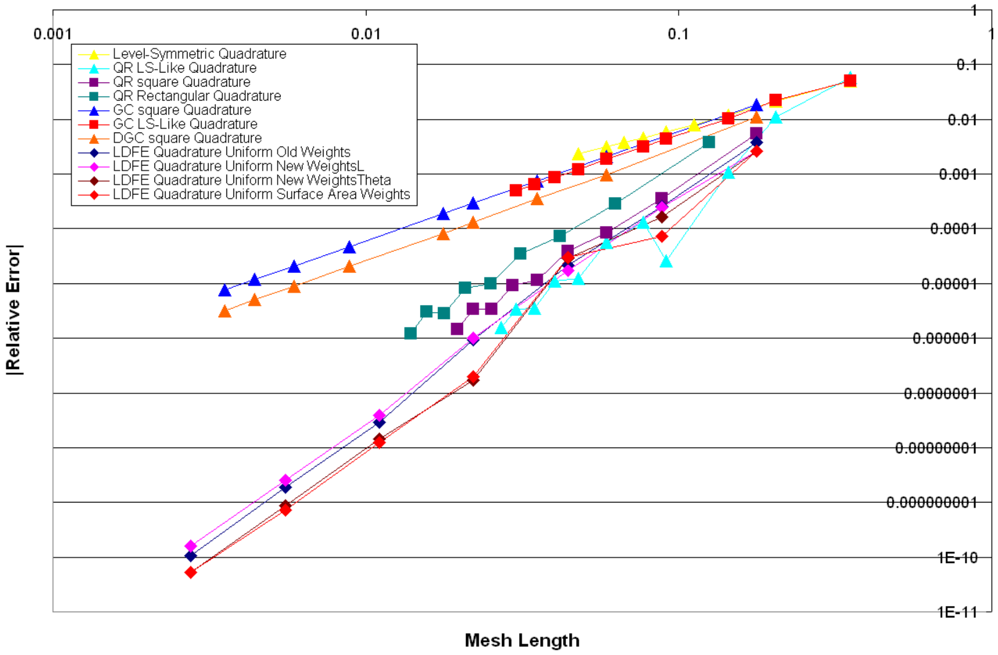

Fig. 5.29. Relative error for LDFE, QR, DGC, GC, and LS uniform quadrature sets versus mesh length for two-region, one cell per region, one-group, purely absorbing, fixed-source problem with vacuum boundary conditions.

Table 5.5

Convergence order of uniform quadrature sets' cell-centered scalar fluxes as a function of mesh length for a purely absorbing, two-region problem.

Type of Quadrature	Convergence Order
LDFE Quadrature Uniform Surface Area Weights	4.33
LDFE Quadrature Uniform Old Weights	4.25
LDFE Quadrature Uniform New WeightsL	4.07
LDFE Quadrature Uniform New WeightsTheta	4.38
Level-Symmetric Quadrature	1.55
QR LS-Like Quadrature	4.03
QR Square Quadrature	3.64
QR Rectangular Quadrature	3.56
GC Square Quadrature	1.99
GC LS-Like Quadrature	1.92
DGC Square Quadrature	2.07
DGC LS-Like Quadrature	1.86

5.3.2 Locally Refined Quadrature Sets

The locally adaptive quadrature sets were analyzed in order to judge the impact of the mapping algorithms. The problem was analyzed using a OC Tol of 0 and varying the PW Tol from 30% to 0.01% for the LDFE-SA sets, LDFE-L sets, LDFE-Θ sets, and the LDFE-Center sets, which are labeled PW Tol. It was then analyzed using a PW Tol of 0 and varying the OC Tol from 30% to 0.01% for each quadrature set, which are labeled OC Tol. The results can be seen in Figure 5.30 and the orders of convergence can be seen in Table 5.6. The locally adaptive order of convergence is lower than the uniform quadrature sets' convergence, which we hypothesize can be attributed to the lack of accuracy when integrating higher order polynomials in the cosines. However, the two-region problem's adaptive convergences were lower than the one-region problem's adaptive convergences for all eight adaptive cases. This drop in convergence can be attributed, we believe, to the mapping scheme, which introduces some distortion in the angular distribution that is transmitted across the boundary between regions with different quadrature sets.

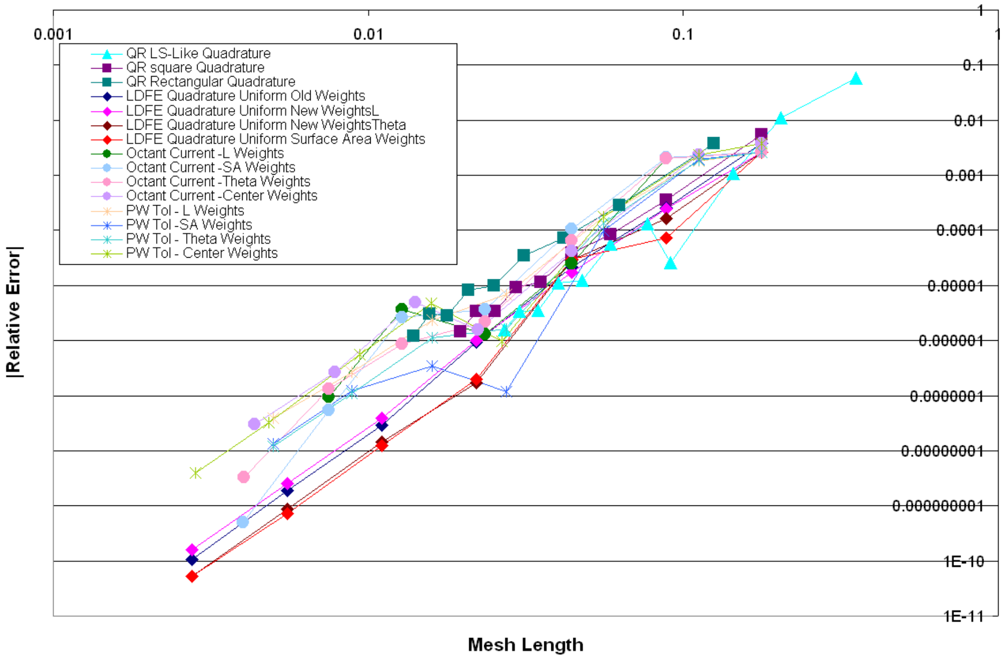

Fig. 5.30. Relative error for QR uniform sets, LDFE-L, LDFE-Θ, LDFE-SA, and LDFE-Center adaptive and uniform quadrature sets versus mesh length for two-region, one cell per region, one-group, purely absorbing, fixed-source problem with vacuum boundary conditions.

Table 5.6
Convergence order of locally adaptive quadrature sets' cell-centered scalar fluxes as a function of mesh length for a purely absorbing, two region problem.

Type of Quadrature	**Convergence Order**
PW Tol - Center Weights	3.35
PW Tol - Θ Weights	3.62
PW Tol - SA Weights	3.66
PW Tol - L Weights	3.27
Octant Current - Center Weights	3.20
Octant Current - Θ Weights	3.67
Octant Current - SA Weights	3.66
Octant Current - L Weights	3.29

5.3.3 Quadrature-to-quadrature Mapping

While this problem has fairly uniform quadrature sets mapping from one region to then next, errors still arise. It is difficult to pass information from a fine to a coarse angular grid or a coarse to a fine angular grid and still preserve an accurate solution in the downstream region. All methods for mapping that were discussed in Section 4.7 have downsides. In our scheme, we have given high priority to conserving the partial current normal to the face; this preserves conservation of particle flow across the boundary. Every mapping technique that we have considered has the potential to artificially induce refinement and coarsening in select problems and skew the angular flux and scalar flux solutions. The quadrature-mapping field is in its infancy and has many opportunities for future work.

5.4 Three-cell Problem

Having established that the LDFE uniform sets yield accurate results for simple problems, we move toward analyzing a more difficult problem. This one-group, purely absorbing problem with vacuum boundary conditions contains three cells and was solved using the step-characteristics method for spatial discretization as shown in Figure 5.15. The first cell is 1 cm × 1 cm × 100 cm and consists of a fixed source of

$100\frac{neutrons}{cm^3 s*ster}$ with a total macroscopic cross section of $\Sigma_t = 0.3\ cm^{-1}$. The second is a 8 cm × 1 cm × 100 cm cell with no fixed source and a total macroscopic cross section of $\Sigma_t = 0.3\ cm^{-1}$. The third cell is 1 cm × 1cm × 100 cm, contains no fixed source, and has a total macroscopic cross section of $\Sigma_t = 0.3\ cm^{-1}$. The reference solution is considered to be the LDFE-SA uniform set solution with 524288 directions. This problem can be analyzed using two different refinement approaches.

5.4.1 One-region, Three-cell Analysis

The first approach uses one quadrature region with all three cells contained in that region. This is important for our adaptive method for the following reasons:

1. Comparisons of the angular fluxes for refinement purposes occur only on the region boundaries, not the cell boundaries.

2. The quadrature-to-quadrature mapping is invoked only if more than one quadrature set exists in the problem.

By encapsulating all three cells in a single region, we can explore the adaptive scheme without having the mapping scheme impact the results. We initially ran this problem with the uniform LDFE quadrature sets, the QR quadrature sets, and the LS quadrature sets. The absolute value of the relative error in the cell-centered scalar flux for each cell versus the cell-averaged angular mesh length can be seen in Figures 5.31, 5.32, and 5.33. We then ran adaptive cases with the PW tolerance at zero and the OC tolerance varying from 30% to 1% for LDFE-SA, LDFE-Θ, LDFE-L, and LDFE-Center set with the absolute value of the relative error in the cell-centered scalar fluxes as a function of cell-averaged angular mesh length shown in Figures 5.34, 5.35, and 5.36. Unfortunately in some cases, the number of directions became so large the memory requirements became impossible to manage with our serial code. Our solution was to restrict the number of directions in a given region. If the number of directions in a region became larger than $100,000$, we stopped

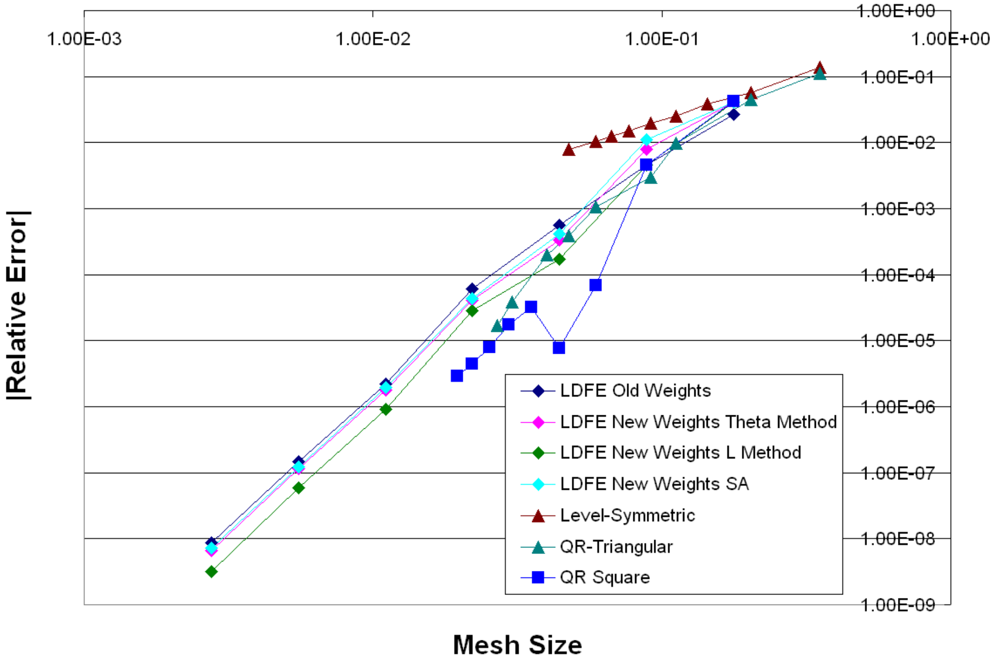

Fig. 5.31. The relative error in the first cell associated with the LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

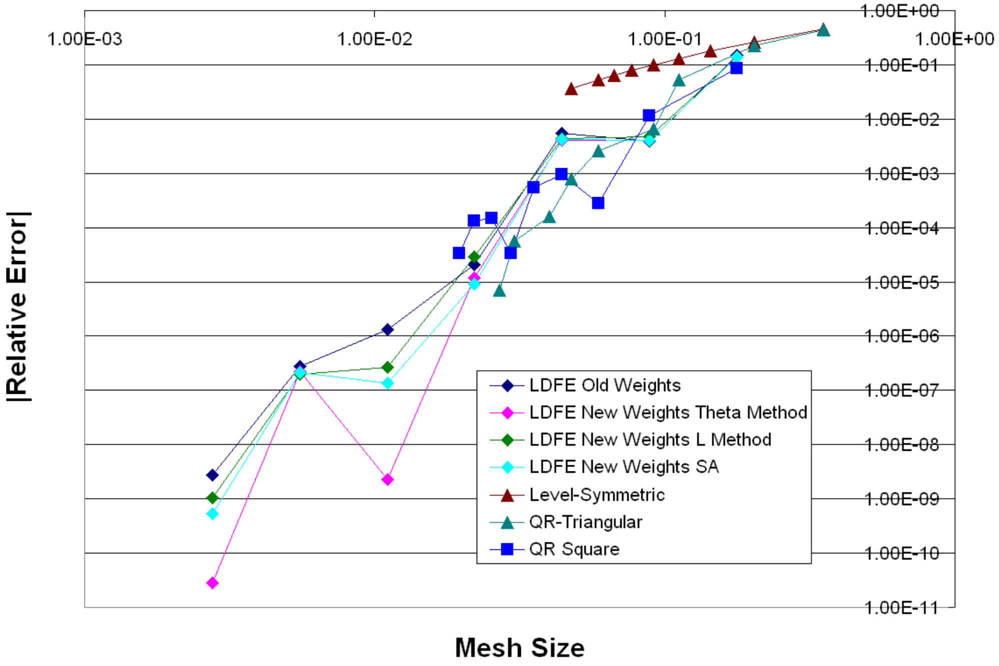

Fig. 5.32. The relative error in the second cell associated with the LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

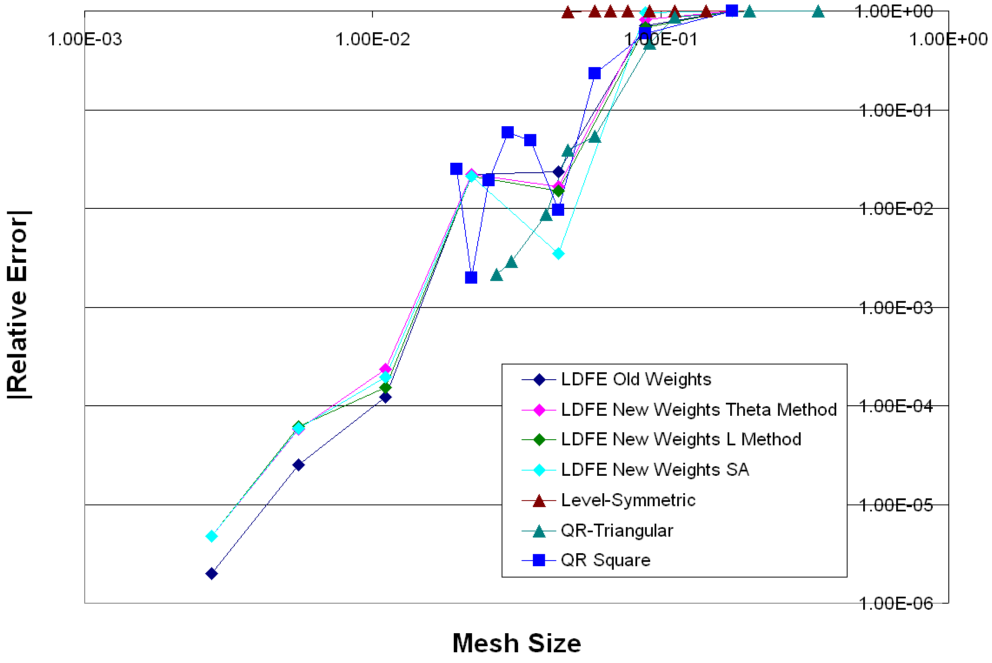

Fig. 5.33. The relative error in the third cell associated with the LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

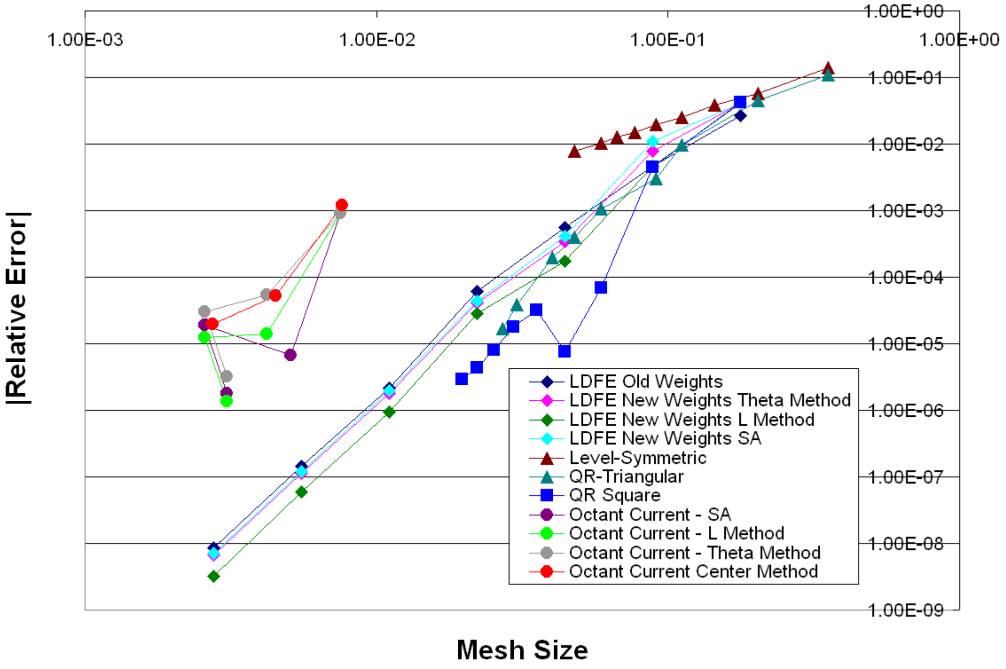

Fig. 5.34. The relative error in the first cell associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

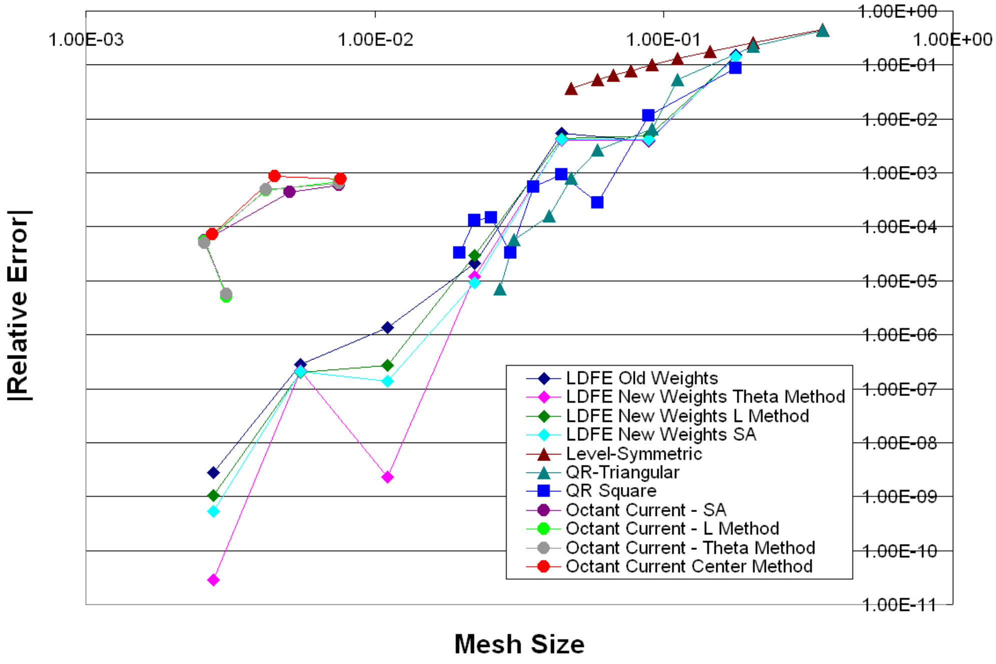

Fig. 5.35. The relative error in the second cell associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

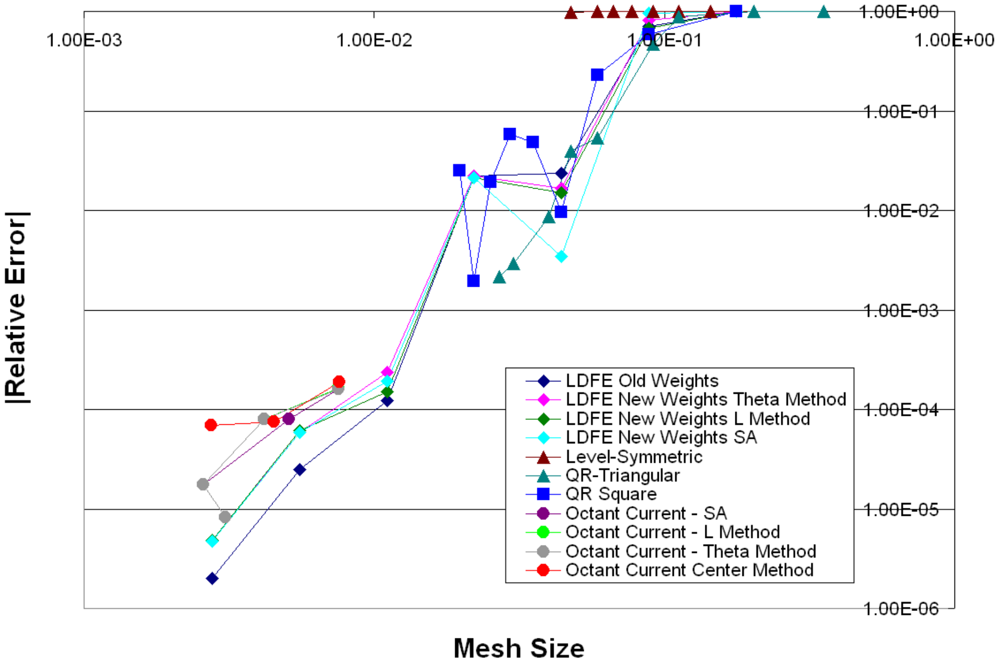

Fig. 5.36. The relative error in the third cell associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the one-region, one-group, three-cell, purely absorbing problem with vacuum boundary conditions.

testing for refinement. This number of directions corresponds to a mesh length of 3.16e-3 in our figures. This led to certain smaller tolerance problems containing fewer directions than those quadrature sets that had a higher tolerance. As these figures show, the scalar flux using a locally refined quadrature set for the first and second cell are not calculated as efficiently as the uniform set's scalar fluxes while the third cell's scalar flux is calculated almost as efficiently. It is not difficult to understand why this occurs. In this problem, with its single quadrature region, the highly peaked angular flux in the third cell causes a large number of directions to be added to the quadrature set to resolve the peak. These added directions do not help much with the scalar-flux integrals in the first two cells; thus the adapted set in this problem is not at all efficient for those two cells. The angular flux on the third cell's outermost boundary can seen in Figure 5.37, which also shows the adapted quadrature mesh (with each triangle containing one quadrature point). Notice that it is strongly peaked near the $\mu = 1$ direction.

5.4.2 Three-region, One Cell per Region

In order to allow each cell to have a different quadrature set, we made each cell its own region. While this allows different quadrature sets, this introduces quadrature-to-quadrature mapping which can limit the accuracy of the solution. The uniform sets give the same scalar flux solutions as the one-region, three-cell problem which we see in Figures 5.31, 5.32, and 5.33. However, the locally adaptive solutions are different. When we plot the cell-centered scalar flux as a function of unknowns for each region (instead of the entire problem), we see behavior as shown in Figures 5.38, 5.39, and 5.40. The first region's adaptive sets produce approximately the same efficiency as the uniform refinement sets. However, the second region has much lower efficiency. We believe that this is due to the quadrature-to-quadrature mapping from cell one to cell two. We are mapping from a region with hundreds of directions to a region with many thousands of directions. This coarse-to-fine mapping significantly

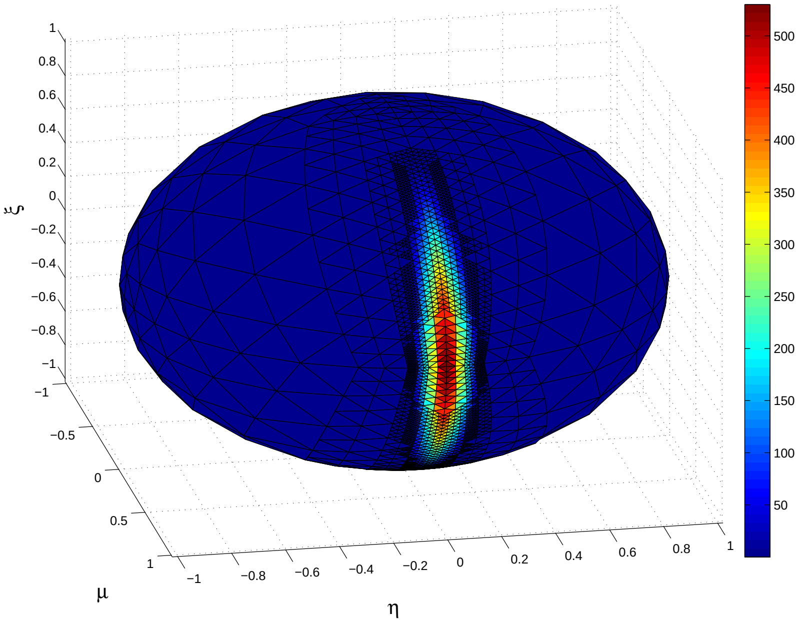

Fig. 5.37. Pseudocolor plot of the angular flux on the boundary face of the third cell as a function of direction.

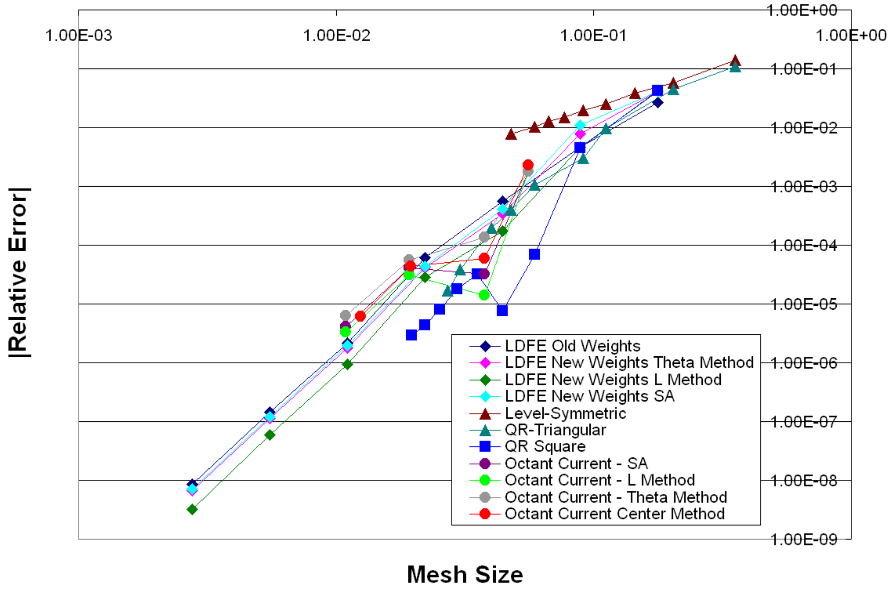

Fig. 5.38. The absolute value of the cell-centered scalar-flux relative error in the first cell versus the first region's angular mesh length associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the three-region, one cell per region, one-group, pure absorbing problem with vacuum boundary conditions.

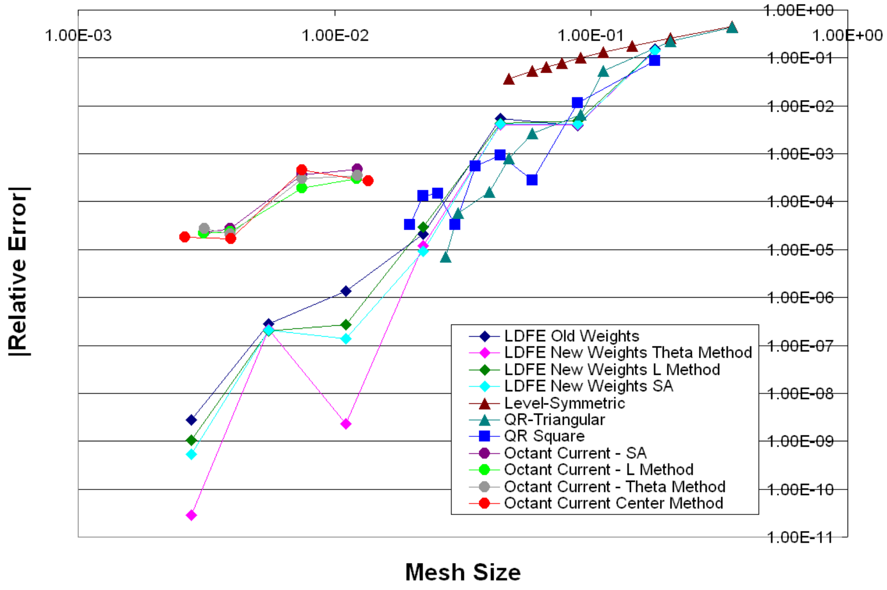

Fig. 5.39. The absolute value of the cell-centered scalar-flux relative error in the second cell versus the second region's angular mesh length associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the three-region, one cell per region, one-group, purely absorbing problem with vacuum boundary conditions.

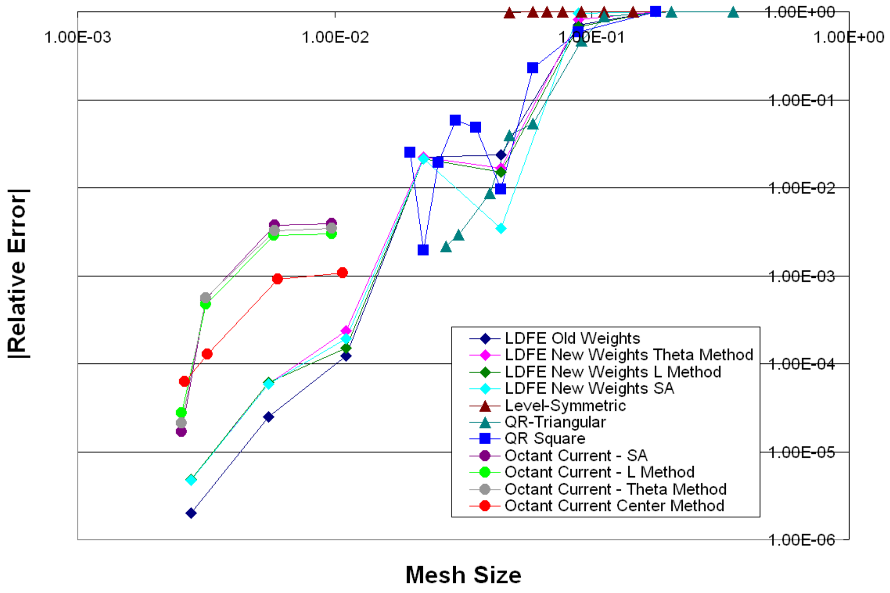

Fig. 5.40. The absolute value of the cell-centered scalar-flux relative error in the third cell versus the third region's angular mesh length associated with the adaptive and uniform LDFE-SA, LDFE-Θ, LDFE-L, LDFE-Center, QR, and LS quadrature sets for the three-region, one cell per region, one-group, purely absorbing problem with vacuum boundary conditions.

impacts the accuracy of the solution. The third region's adaptive sets are almost as efficient as the uniform sets in some cases. If we force the second and third regions to have the same quadrature set, thus eliminating mapping-induced perturbations at the interface, we see much more accurate results. We note that the incident flux on cell three comes from a narrow cone of exiting directions from cell one. The details of the distribution in that cone are not important to the cell-one scalar flux; thus, cell one does not adapt to resolve them. However, they are very important to the cell-three scalar flux, yet cell three has no control over their resolution. No matter how much the cell three quadrature set is refined, it does not improve the accuracy of the distribution that emerges from cell one, given the algorithm that we are using. Again, we believe that the perturbations introduced by the mapping from one quadrature set to the next significantly impacts the accuracy of the solution. In all cases, we observe scalar flux accuracy to one percent in all cells.

In these test problems, the cell-averaged scalar-flux errors are always significantly smaller – usually by orders of magnitude – than the tolerances that go into refinement and coarsening decisions. Thus, if the input imposes a 1% tolerance, the scalar fluxes in these problems have errors that are substantially smaller than 1%. There is a need for an algorithm that estimates the actual cellwise scalar-flux error that will result from a given Point-Wise tolerance and Octant-Current tolerance, but we have not developed a theory or algorithm for this. Perhaps a starting point for such a theory would be the conservation equation, Equation 4.52.

5.5 Skewed Duct Problem

The last problem we analyzed was a three-region problem with a skewed duct going through the second region as shown in Figure 5.41. This problem is a purely absorbing problem with almost all the particles reaching region three reaching it by coming through the skewed duct. All boundary conditions are vacuum and the first region has a constant fixed source of $100\frac{neutrons}{cm^3 s*ster}$. The problem is a two-dimensional

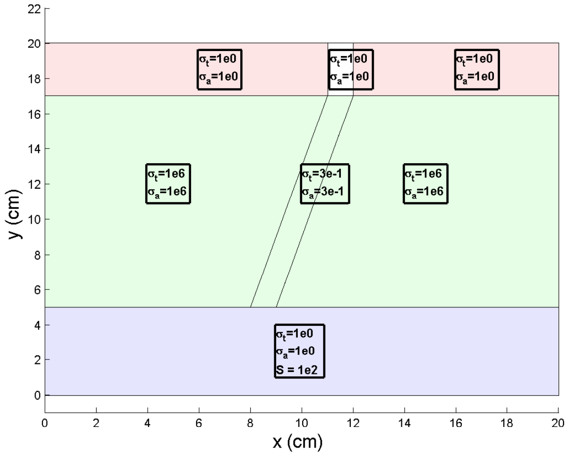

Fig. 5.41. Purely absorbing, three-region, skewed duct problem with vacuum boundary conditions from a two-dimensional view.

figure extruded up the z-axis 20 cm as shown in Figure 5.42 and was solved using the step-characteristics spatial discretization method. The reference solution was considered to be the LDFE-SA solution with 131072 directions. We were restricted to this number of directions because of the memory requirements of the SBA scheme as it is currently implemented in the Jaguar code coupled with the physical memory of the computer we were using. Because of this memory limitation, we also had to limit the number of directions in a region for adaptive refinement to 35000. The purpose of this problem is to assess the performance of existing and new (including adaptive) quadrature sets for a problem that does not have a strictly Cartesian mesh. In this problem, we believe it is appropriate to look at each region and cell separately, so that the source region does not overly impact the accuracy of the adaptive quadrature sets. If we look at a cell-by-cell comparison of the cell-centered scalar flux, we see the benefits of the locally adaptive sets. The cells of interest in this problem are the third region's middle cell and right-most cell. These cells' fluxes arise almost solely from the particles coming up the skewed duct. Therefore, if we can accurately capture the streaming coming up the duct, we should get reasonably

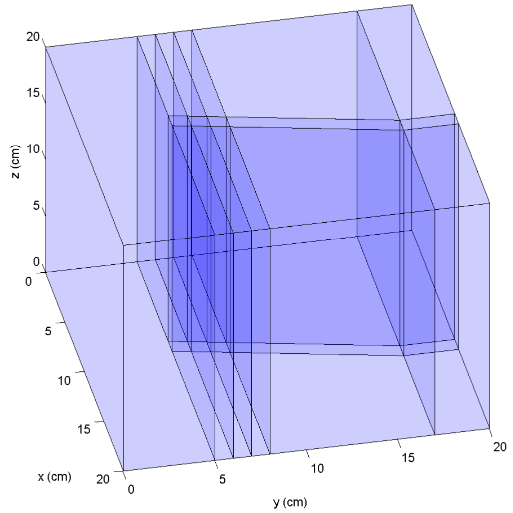

Fig. 5.42. Purely absorbing, three-region, skewed duct problem with vacuum boundary conditions from a three-dimensional view.

accurate solutions for the cell's scalar flux. The third region's middle cell's absolute value of the relative errors in the cell-centered scalar flux as a function of problem cell-averaged angular mesh length for uniform and adaptive quadrature sets can be seen in Figure 5.43 and the third region's last cell's absolute value of the relative errors in the cell-centered scalar flux as a function of problem cell-averaged angular mesh length for uniform and adaptive quadrature sets can be seen in Figure 5.44. The convergence orders from uniform (not adapted) quadrature sets for the third region's middle cell as a function of region's cell-averaged angular mesh length can be seen in Table 5.7.

In this case, the LDFE sets do an excellent job of convergence compared to standard quadrature sets. From Figures 5.43 and 5.44, we can see that the adaptive sets produce more accurate results than the uniform sets for the same total number of unknowns.

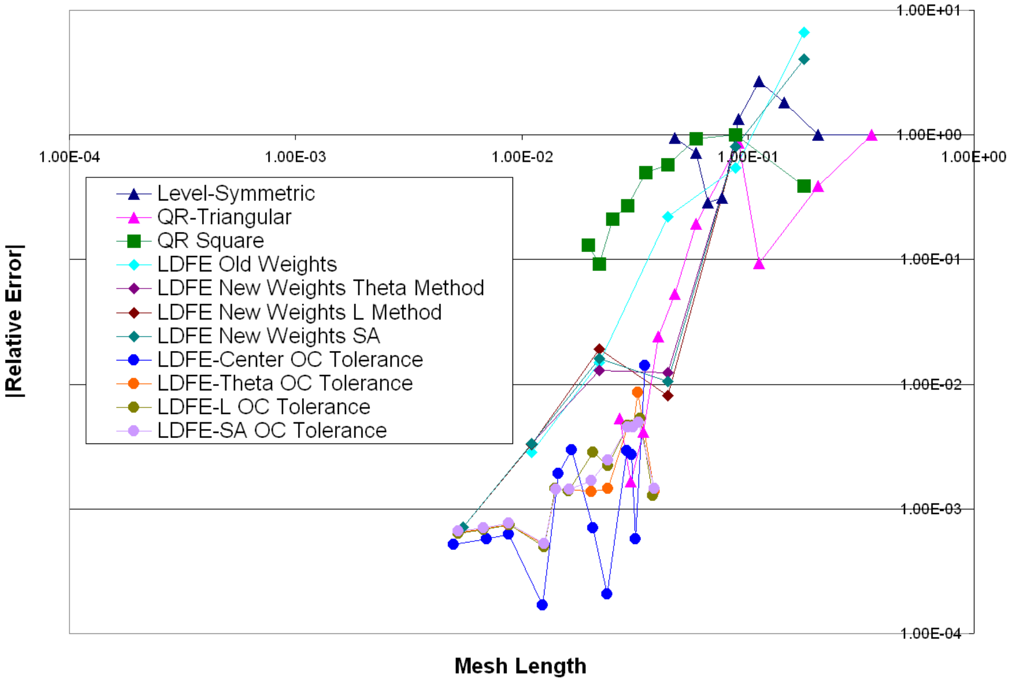

Fig. 5.43. Purely absorbing, three-region, skewed duct problem's third region's middle cell's scalar flux as a function of problem cell-averaged angular mesh length for uniform and adaptive quadrature sets.

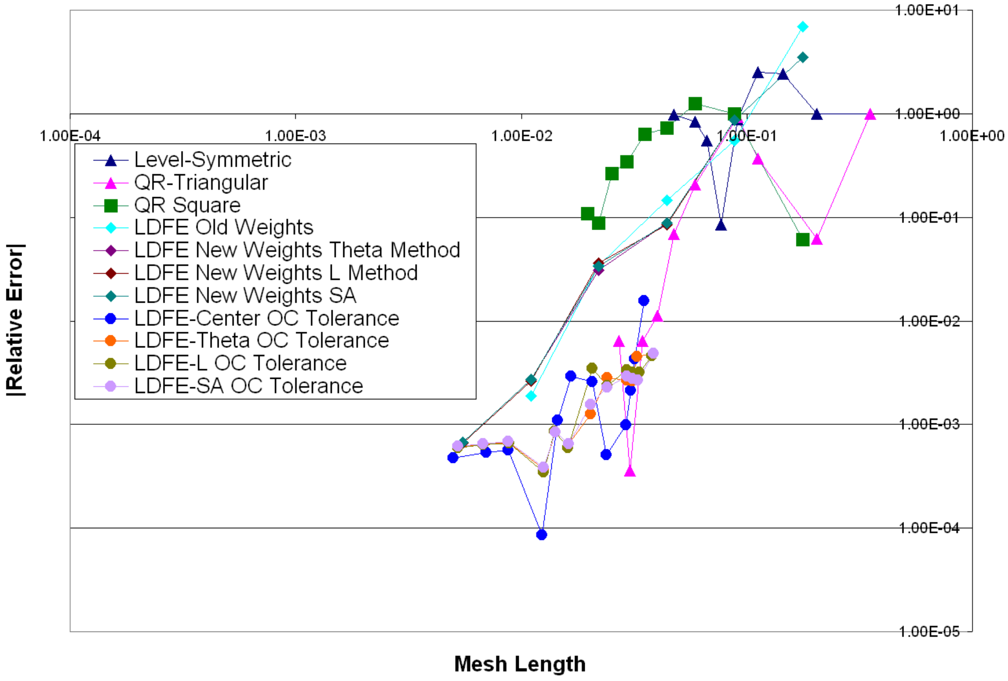

Fig. 5.44. Purely absorbing, three-region, skewed duct problem's third region's last cell's scalar flux as a function of problem cell-averaged angular mesh length for uniform and adaptive quadrature sets.

Table 5.7
Convergence order of uniform quadrature sets' third region's middle cell's scalar flux as a function of problem cell-averaged angular mesh length for the skewed duct problem.

Type of Quadrature	**Convergence Order**
Level-Symmetric Quadrature	0.41
QR-Triangular Quadrature	2.26
QR-Square Quadrature	0.77
LDFE-Center Quadrature	2.76
LDFE-Θ Quadrature	2.46
LDFE-L Quadrature	2.43
LDFE-SA Quadrature	2.44

5.6 Summary

The results shown in this section illustrate the properties of our new quadrature sets. The performance of our uniform sets (no local refinement) on our test problems is dramatically better than that of commonly used quadrature sets (level-symmetric (LS), Gauss-Chebyshev (GC) and variants) and is comparable to that of the Abu-Shumays QR sets. On simple problems the QR sets have slightly better performance and on difficult problems the new sets have better performance. On simple problems the QR sets and the new sets tend to exhibit 4th-order convergence in the scalar flux as the directional mesh is refined, whereas the LS and GC sets exhibit 1.5-order and 2nd-order convergence, respectively. On difficult problems (in which there are near discontinuities in the direction domain along directions that are not perpendicular to coordinate axes), these convergence orders diminish and the new sets outperform the others.

For simple problems the new adaptive sets typically do not outperform the new uniform sets or the QR sets. We have carefully studied this result and believe we understand that it arises from a combination of factors: 1) the new uniform sets have unexpectedly high convergence orders and are thus very accurate; 2) the uniform sets integrate polynomials in the cosines with high-order accuracy, but this high-order behavior is destroyed by local refinement; 3) our refinement testing is purely local and thus does not force an upstream region to resolve directional variations that may be important to a downstream region; 4) mapping from one quadrature set to another introduces perturbations in directional distribution that can be detrimental. Even so, for truly difficult problems (by the above description) the new adaptive sets do outperform all other quadrature sets that we have tested.

We observe that errors in cell-averaged scalar fluxes tend to be smaller (by orders of magnitude) than the tolerances that govern refinement and coarsening decisions. That is, these tolerances have proven to be very conservative upper bounds on scalar-flux errors.

In the next section we draw conclusions from our work and offer suggestions for future work.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusions

In this dissertation, we discussed development of four new quadrature sets derived from the linear discontinuous finite element (LDFE) approach on spherical triangles on the unit sphere. These quadrature sets can be used as standard uniform quadrature sets or can be locally refined. In the adaptive algorithm that we have developed and tested, local refinement of these quadrature sets is driven by a difference between interpolated angular fluxes using the LDFE basis functions and the calculated angular fluxes at test directions that are candidates for addition to the quadrature set. We discussed the decision processes for both adding directions to the quadrature set and removing directions from the quadrature set. We also derived methods for mapping angular fluxes from one quadrature set to a different quadrature set, which is necessary if different spatial regions are allowed to have different quadrature sets, as they are in our implementation. We discussed advantages and disadvantages of both the locally adaptive LDFE quadrature sets and the uniform LDFE quadrature sets.

The uniform LDFE quadrature sets exactly integrate zeroth and first order spherical harmonic functions. Our numerical testing indicates that the integrals calculated by these uniform sets converge to the exact integration of higher-order spherical harmonics at fourth order in what we call the “angular mesh length,” which is inversely proportional to the square root of the numbers of directions. This is based on numerical testing of integrals of polynomials in the direction cosine ranging from second through sixth orders. On simple problems the QR sets and the new sets exhibit approximately 4th-order convergence in the scalar flux as the directional mesh is refined, whereas the LS and GC sets exhibit 1.5-order and 2nd-order convergence, respectively. On more difficult problems, in which there are near discontinuities in the angular flux along directions that are not perpendicular to coordinate axes, the convergence orders diminish for all methods. In such problems, the new quadrature sets outperform the others. We believe that this behavior stems from the new sets' property that their directions and weights are based upon mathematical considerations on local spherical triangles, not on integration over entire octants (as is the case with QR sets) or the entire sphere (as with LS and GC sets). These LDFE sets have the desirable property of having no limitation on refinement, whereas the QR sets are limited by the mathematical complexity of their derivations. Further, three of the LDFE sets (SA, Θ, and L) have strictly positive weights regardless of the level of refinement. The Gauss-Chebyshev and other product-based quadrature sets can be produced with any number of directions but in our test problems they perform far less efficiently than our LDFE sets. We conclude that even without local refinement, our new quadrature sets offer considerable advantages in solving practical discrete-ordinates transport problems.

The new LDFE quadrature sets can be refined locally, one spherical triangle at a time, which opens the possibility of adaptive discrete-ordinates algorithms in which the algorithm adds directions where the angular flux is not smooth. Unfortunately, as the quadrature set becomes more locally refined, which is often what is required to resolve the angular flux shape, the quadrature integration of higher-order spherical harmonics becomes less accurate. This dynamic makes the adaptive quadrature set's accuracy difficult to exactly predict. From our analysis and testing, we find that our adaptive sets can be beneficial in certain problems with streaming gaps that are not perpendicular to the Cartesian axes. Furthermore, we believe that with some extension to our work, which we outline in the next section, the adaptive quadrature sets have the potential to improve the accuracy of transport solutions for an even broader class of problems.

We can give a conservative bound on the scalar flux in a given problem based on the adaptive tolerances but, in general, that bound is orders of magnitude more conservative than the true accuracy of the solution for the problems we analyzed.

6.2 Future Work

6.2.1 Cubic/Linear DFE Quadrature

Our analysis and testing has led us to hypothesize that our locally refined quadrature sets would be much more accurate, especially on "simple" problems, if they could more accurately integrate spherical harmonics of second and higher orders. Here we describe a method in which we can guarantee the exact integration of third-order spherical harmonic functions even if local refinement occurs. We continue to use the octant triangle described in this dissertation and describe a "base" quadrature set that is built from a cubic finite element basis function on each octant. This set would consist of sixteen ordinates defined at the ordinates from the LDFE-SA, LDFE-Θ, LDFE-L, or LDFE-Center ordinates. We would make the cubic finite element basis functions be cardinal functions at each of the sixteen ordinates. (We note that there are exactly sixteen linearly independent third-order polynomials in the direction cosines, or equivalently, exactly sixteen third-order spherical-harmonics functions. Thus there are sixteen linearly independent basis functions that span the space of third-order spherical-harmonics functions.) These basis functions are defined as

$$b^b_m(\overrightarrow{\Omega}) \text{ for } m = 1:16 \ . \tag{6.1}$$

The weight associated with a given point is the octant integral of the point's cubic basis function. This completely defines a base quadrature set as

$$\text{Base Set} \equiv \{w^b_m, \overrightarrow{\Omega}^b_m\} \text{ where } \ m = 1..16 \ . \tag{6.2}$$

If we refined the quadrature set, we want the introduction of new ordinates to maintain the accuracy of integration of the base quadrature set. Suppose that the next set of points to be added to the quadrature set is defined as

$$\text{First Adaptive Set} \equiv \{w^a_n, \overrightarrow{\Omega}^a_n\} \text{ where } \ n = 1..N_a \ . \tag{6.3}$$

We can maintain the accuracy of our integrations by applying the adaptive set only to the difference between the actual angular flux and the third-order interpolant through the base set's points. That is, we define the adapted quadrature set's integration as

$$\int_{\delta\Omega} d\Omega f(\overrightarrow{\Omega}) = \sum_{m=1}^{16} w_m^b f(\overrightarrow{\Omega}_m^b) + \sum_{n=1}^{N_a} w_n^a [f(\overrightarrow{\Omega}_n^a) - f_b^{interp}(\overrightarrow{\Omega}_n^a)] \ , \tag{6.4}$$

where

$$f_b^{interp}(\overrightarrow{\Omega}) \equiv \sum_{m=1}^{16} f(\overrightarrow{\Omega}_m^b) b_m^b(\overrightarrow{\Omega}) \ . \tag{6.5}$$

Note that this adapted quadrature integration remains perfect for any third-order polynomial f, because for any such function the difference in square brackets will be zero. Equation 6.4 can be factored in the following manner:

$$\int_{\delta\Omega} d\Omega f(\overrightarrow{\Omega}) = \sum_{m=1}^{16} [w_m^b - \sum_{n=1}^{N_a} w_n^a b_m^b(\overrightarrow{\Omega}_n^a)] f(\overrightarrow{\Omega}_m^b) + \sum_{n=1}^{N_a} w_n^a f(\overrightarrow{\Omega}_n^a) \ . \tag{6.6}$$

We then define the updated weights of the base set $\tilde{w}_m^b$ as

$$\tilde{w}_m^b \equiv w_m^b - \sum_{n=1}^{N_a} w_n^a b_m^b(\overrightarrow{\Omega}_n^a) \ , \tag{6.7}$$

which simplifies Equation 6.6 to

$$\int_{\delta\Omega} d\Omega f(\overrightarrow{\Omega}) = \sum_{m=1}^{16} \tilde{w}_m^b f(\overrightarrow{\Omega}_m^b) + \sum_{n=1}^{N_a} w_n^a f(\overrightarrow{\Omega}_n^a) \ . \tag{6.8}$$

This method has the potential to guarantee the exact integration of up through third order spherical harmonic functions for both uniform and locally refined quadrature sets. The drawbacks to this method include the possibility of introducing negative weights and the increased complexity of incorporating the method in a discrete ordinate code.

6.2.2 Alternative Tesselations of the Sphere

Another area for that has potential for future work is using different polyhedra as bases for dividing the surface of the sphere into spherical polygons. Our quadrature sets were defined using an octahedron's projection to the unit sphere. The possibility of defining a finite element quadrature set from the unit cube is interesting. In this case the corners of the cube would lie at the centers of the octants and the centers of the faces of the cubes would be pierced by the x, y, and z axes. Each square face of the cube could be refined into four squares, and any square could be further refined into four others, etc. The projection of these squares onto the surface would form spherical quadrilaterals whose edges were great circles. It would be natural to define four quadrature points per quadrilateral with each point having an associated basis function that would be linear in the direction cosines. In many ways this square-based tesselation seems better suited to LDFE quadratures than does the triangle-based tesselation that we explored in this dissertation.

6.2.3 Implementation with Other Spatial Discretizations

The work in this dissertation shows that the adaptive quadrature sets have more accurate solutions while using less directions in certain problems. It would be interesting to add this adaptive method to a standard transport code instead of a Slice Balance Approach (SBA) code. Because Jaguar is based on the SBA, slices are required to be produced and stored for each direction. Once tens and hundreds of thousands of directions are being utilized, a large amount of memory is required. Other types of codes would have less memory restrictions and could therefore add many more directions than we are currently limited to. We could, therefore, analyze more complicated problems. We also note that Jaguar's SBA could be written in a more computationally efficient manner; this includes decomposing the problem into specific "chunks" and recomputing the slices "on the fly" instead of storing them.

6.2.4 Error Bounds

Our numerical results demonstrate that our adaptive sets yield relative errors in volume-averaged scalar fluxes that are far smaller than the tolerances used in our refinement tests. That is, the tolerances that govern refinement decisions provided extremely conservative upper bounds on the error in the scalar flux. This is not difficult to understand if we examine the region-wise conservation equation arranged into an expression for the region-averaged scalar flux:

$$\Phi_{\text{reg}} = \frac{\sum_{f=1}^{FACES}(J_{f,\text{inc}} - J_{f,\text{exit}})A_{\text{f}} + Q_{\text{Fixed,reg}} + Q_{\text{Fission,reg}}}{\sigma_a V_{\text{reg}}} \,. \tag{6.9}$$

We have examined many test problems in which the source (Q) terms in the above equation are zero, which means that the relative error in the scalar flux equals the relative error in the difference between the incident and exiting partial currents. Our refinement algorithm guarantees that each $J_{f,\text{exit}}$ and each $J_{f,\text{inc}}$ is accurate to within the "Octant-Current" tolerance. This does not guarantee that the difference of the two is accurate to within that tolerance and thus does not guarantee that the scalar flux will be accurate to within that tolerance. However, our results indicate that the scalar flux is always much more accurate than the tolerance. This suggests that in the problems we have studied, the errors in angular current on the many triangles that make up an incident or exiting half-space are in large part canceling each other when the contributions from the triangles are summed. This in turn suggests that it may be possible to accumulate an error estimate in partial currents and use this in the balance equation (above) to estimate the error in volume-averaged scalar flux. It would be interesting to explore this in the future.

6.2.5 Other Future Work

The version of Jaguar that we implemented our method in was a serial code. It would be interesting to see how this adaptive quadrature method scales for parallel codes. Load balancing tends to be an issue in any parallel implementation of any adaptive method. Because the code tests for adaptation only on region boundaries, we may miss information on the cell boundaries, and furthermore a given region has no way (in our current algorithm) to tell an upstream region that it needs detailed information in part of the directional domain. One way to address this latter issue is to base refinement on “goal-oriented” tests, which usually take advantage of adjoint solutions to determine where errors arise and thus where refinement should occur [27] [28] [29] . Other interesting areas for future work include: testing for refinement on all cell boundaries not just the region boundaries, analyzing how the timing of refinement testing impacts the solution, and increasing the finite element basis functions from linear functions to quadratic, cubic, or higher order functions.

REFERENCES

[1] E.E. Lewis, W.F. Miller, Computational Methods of Neutron Transport, John Wiley and Sons, New York, 1984, pp. 1-401.

[2] R.E. Grove, A Diamond-Difference-Like SBA Scheme (SBA-DDL) for Polyhedral Meshes, in: American Nuclear Society Winter Meeting, Washington D.C., 2007.

[3] R.E. Grove, The Slice Balance Approach (SBA): A Characteristic-Based, Multiple Balance Sn Approach on Unstructured Polyhedral Meshes, LM-05K094, Knolls Atomic Power Laboratory, Niskayuna, NY, 2005.

[4] R.E. Grove, A Characteristic-Based Multiple Balance Approach for Solving the S_n Equations on Arbitrary Polygonal Meshes, PhD dissertation, University of Michigan, Ann Arbor, MI, 1996.

[5] K.D. Lathrop, Ray Effects in Discrete Ordinates Equations, Nuclear Science and Engineering 32 (1968) 357–369.

[6] M.L. Adams, Angular Dependence of the Fast Flux in Reactor Lattices, Transactions of the American Nuclear Society 81 (2001) 212.

[7] J.E. Morel, T. A. Wareing, Analysis of Ray-effect Mitigation Techniques., Nuclear Science and Engineering 144 (1) (2003) 1–22.

[8] K.D. Lathrop, Remedies for Ray Effects, Nuclear Science and Engineering 45 (1971) 255-268.

[9] K.D. Lathrop, Elimination of Ray Effects in Curved Geometries, Transactions of the American Nuclear Society 15 (1972) 272-273.

[10] W.F. Miller Jr., W.H. Reed, Ray-effect Mitigation Methods for Two-Dimensional Neutron Transport Theory, Nuclear Science and Engineering 62 (10) (1977) 391-411.

[11] L.L Briggs, W.F. Miller, E.E. Lewis, Ray-effect Mitigation in Discrete Ordinate-like Angular Finite Element Approximations in Neutron Transport, Nuclear Science and Engineering 57 (3) (1975) 205-217.

[12] I.K. Abu-Shumays, Compatible Product Angular Quadrature for Neutron Transport in X-Y Geometry, Nuclear Science and Engineering 64 (1977) 299-316.

[13] I.K. Abu-Shumays, Angular Quadratures for Improved Transport Computations, Transport Theory and Statistical Physics 30(2&3) (2001) 169-204.

[14] I.K. Abu-Shumays, C.E. Yehnert, Angular Interpolations and Splice Options for Three-Dimensional Transport Computations, DE-AC11-93PN38195, Bettis Atomic Power Laboratory, Pittsburgh, PA, 1996.

[15] A.H. Stroud, D. Secrest, Gaussian Quadrature Formulas, Prentice-Hall, Englewood Cliffs, NJ, 1966, pp. 1-374.

[16] W.A. Rhoades, Y.Y. Azmy, Three-dimensional SN calculations with Oak Ridge TORT code, in: International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses, Portland, OR, 1995.

[17] G. Longoni, A. Haghighat, Development of New Quadrature Sets with the Ordinate Splitting Technique, in: Proceedings of the 2001 American Nuclear Society International Meeting on Mathematical Methods for Nuclear Applications, Salt Lake City, UT, 2001.

[18] G. Longoni, A. Haghighat, Development and Application of the Regional Angular Refinement Technique and its Applications to Non-Conventional Problems, in: International Conference on the New Frontiers of Nuclear Technology: Reactor Physics, Safety, and High-Performance Computing, Seoul, Korea, 2002.

[19] G. Longoni, Advanced Quadrature Sets, Acceleration, and Preconditioning Techniques for the Discrete Ordinates Method in Parallel Computing Environments, PhD dissertation, University of Florida, 2004.

[20] P.N. Brown, B. Chang, C.J. Clouse, Locally Refined Quadrature Rules for S_n Transport, UCRL-JRNL-220755, Lawrence Livermore National Laboratory, Livermore, CA, 2006.

[21] J.C. Stone, Adaptive Discrete-ordinates Algorithms and Strategies, PhD dissertation, Texas A&M University, College Station, TX, 2007.

[22] J.C. Stone, M.L. Adams, A Piecewise Linear Finite Element Basis with Application to Particle Transport, in: Nuclear Mathematical and Computational Sciences Meeting, Gatlinburg, TN, 2003.

[23] J.C. Stone, M.L. Adams, Progress on Adaptive Discrete-Ordinates Algorithms and Strategies, in: Nuclear Mathematical and Computational Sciences Meeting, Gatlinburg, TN, 2003.

[24] A.M. Watson, The Effectiveness of Krylov Methods Applied to SBA, in: American Nuclear Society, Winter Meeting, Washington D.C., 2007.

[25] A.M. Watson, R.E Grove, M.T. Shearer, Effective Software Design for a Deterministic Transport System, in: International Conference on Advances in Mathetatics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, 2009.

[26] R.E. Grove, M.T. Shearer, Surface Mesh Refinement with the Slice Balance Approach (SBA), in: International Conference on Advances in Mathetatics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, 2009.

[27] Y. Wang, Adaptive Mesh Refinement Solution Techniques for the Multigroup SN Transport Equation Using a Higher-order Discontinuous Finite Element Method, PhD dissertation, Texas A&M University, 2009.

[28] J.C. Ragusa, Y. Wang, A Two-mesh Adaptive Mesh Refinement Technique for SN Neutral-particle Transport Using a Higher-order DGFEM, Journal of Computational and Applied Mathematics 233 (12) (2009) 3178-3188.

[29] Y. Wang, J.C. Ragusa, Application of HP Adaptivity to the Multigroup Diffusion Equations, Nuclear Science and Engineering 161 (1) (2009) 22–48.

APPENDIX A
MATH OF SPHERICAL INTEGRATION

A.1 Background

In order to determine the weights of the discrete ordinates for our method, the basis functions must be integrated over the spherical triangle formed by three great circles. The spherical coordinates, as shown in Figure A.1, are mathematically defined as follows:

$$\xi = cos(\theta) \ , \tag{A.1a}$$

$$\mu = cos(\gamma)sin(\theta) = cos(\gamma)\sqrt{1-\xi^2} \ , \tag{A.1b}$$

and

$$\eta = sin(\gamma)sin(\theta) = sin(\gamma)\sqrt{1-\xi^2} \ . \tag{A.1c}$$

We define the linear basis functions, based on Equations A.1a, A.1b, and A.1c, as

$$b(\overrightarrow{\Omega}) = c_c + c_\mu \mu + c_\eta \eta + c_\xi \xi \ . \tag{A.2}$$

Based on Figures A.1, A.2, A.3, and A.4, the following geometric properties are true:

$$h = \sqrt{R^2 + \left(\frac{R}{\sqrt{2}}\right)^2} = R\sqrt{3/2} \ , \tag{A.3}$$

$$b = \frac{2h}{3} \ , \tag{A.4}$$

$$b = R\sqrt{2} \ , \tag{A.5}$$

$$sin\,(\alpha) = \frac{1}{\sqrt{3}} \ , \tag{A.6}$$

and

$$cos\,(\alpha) = \sqrt{\frac{2}{3}} \ . \tag{A.7}$$

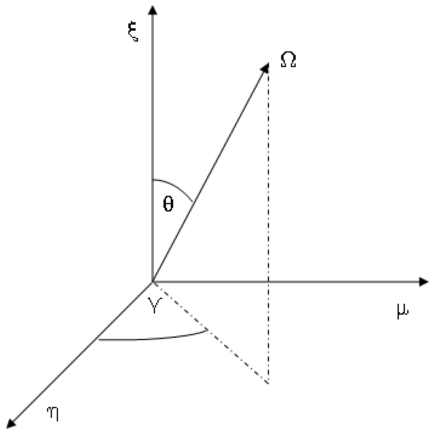

Fig. A.1. Spherical coordinate system.

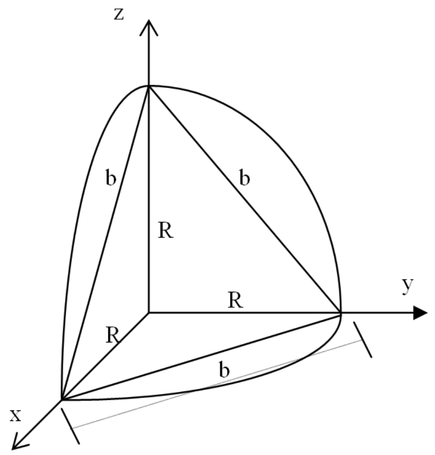

Fig. A.2. Original octant with 45° triangle.

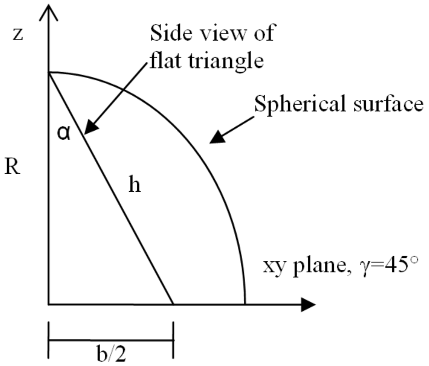

Fig. A.3. Side view of flat triangle.

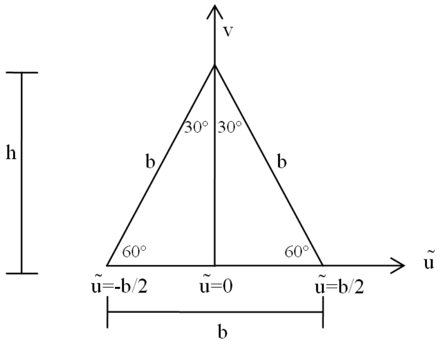

Fig. A.4. Flat view of triangle using $\tilde{u}$ and v.

A.2 Flat Triangle Definition and Change of Variables

From Figures A.2 and A.4, we see that the flat triangle lies in the plane that intersects the xy, yz, and xz planes with an angle of 45°. We perform our calculations on this flat triangle. If we begin with the flat triangle illustrated in Figure A.4, we can transform from the $(\tilde{u}, v)$ variables to the (u, v) variables. On our flat triangle, the vertical range, v, ranges from 0 to h. For a given v, the range of $\tilde{u}$ is

$$\left(-\frac{h-v}{h}\frac{b}{2}, \frac{h-v}{h}\frac{b}{2}\right) \ . \tag{A.8}$$

The area of the triangle is

$$A = \int_0^h dv \int_{-\frac{h-v}{h}\frac{b}{2}}^{\frac{h-v}{h}\frac{b}{2}} \tilde{u} = \int_0^h dv \frac{h-v}{h} b = \frac{1}{2} bh \ . \tag{A.9}$$

We can relate u to $\tilde{u}$ by Equation A.10.

$$u = \frac{\tilde{u}}{\tilde{u}_{max}} = \frac{2h\tilde{u}}{b(h-v)} \tag{A.10}$$

Therefore, the derivative of $\tilde{u}$ is

$$d\tilde{u} = \frac{b(h-v)}{2h} du \ . \tag{A.11}$$

A.3 Relationship between Spherical and Flat Triangle Coordinates

A.3.1 The Relationship between γ and u

Figure A.5 illustrates the view looking down the z-axis towards the xy plane. Projecting v onto the xy plane yields:

$$v_{xy} = v\frac{b}{2h} = \frac{v}{\sqrt{12}} \ , \tag{A.12}$$

$$h_{xy} = \frac{b}{2} \ , \tag{A.13}$$

$$(h-v)_{xy} = \frac{b}{2}\frac{h-v}{h} = h_{xy}\frac{h-v}{h} \ , \tag{A.14}$$

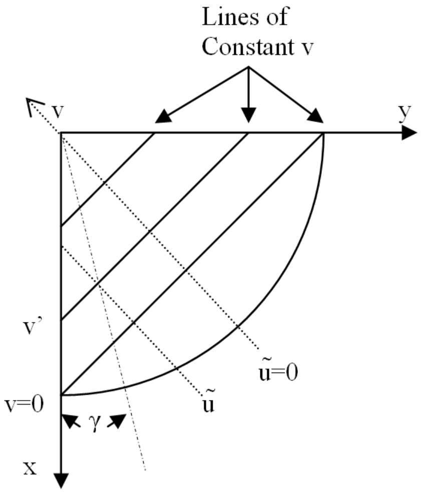

Fig. A.5. View looking down the z-axis towards the xy plane.

and

$$\Rightarrow \frac{(h-v)_{xy}}{h_{xy}} = \left(\frac{h-v}{h}\right) \ . \tag{A.15}$$

Thus, using the trigonometric definitions, we realize that

$$\begin{aligned} tan\left(\frac{\pi}{4}-\gamma\right) &= \frac{|\tilde{u}|}{(h-v)_{xy}} = \frac{|\tilde{u}|/\tilde{u}_{max}}{(h-v)_{xy}/\tilde{u}_{max}} \\ &= \frac{|u|}{(h-v)_{xy}}\frac{b(h-v)}{2h} = |u|\frac{\frac{h-v}{h}}{\frac{(h-v)_{xy}}{h_{xy}}} \\ &= |u| \ . \end{aligned} \tag{A.16}$$

If $u < 0$, $\gamma < \frac{\pi}{4}$ and if $u > 0$, $\gamma > \frac{\pi}{4}$. This allows the transformation of Equation A.16 to

$$tan\left(\frac{\pi}{4}-\gamma\right) = -u \ . \tag{A.17}$$

In the 2nd, 3rd, and 4th quadrants, Equation A.17 becomes, respectively:

$$tan\left(\frac{3\pi}{4}-\gamma\right) = -u \ , \tag{A.18a}$$

$$tan\left(\frac{5\pi}{4}-\gamma\right) = -u \ , \tag{A.18b}$$

and

$$tan\left(\frac{7\pi}{4}-\gamma\right)=-u\ . \tag{A.18c}$$

We take the derivative of Equation A.17:

$$\frac{\partial u}{\partial\gamma}=sec^2\left(\frac{\pi}{4}-\gamma\right)\Rightarrow\frac{\partial\gamma}{\partial u}=cos^2\left(\frac{\pi}{4}-\gamma\right)\ . \tag{A.19}$$

We then solve for $cos^2\left(\frac{\pi}{4}-\gamma\right)$ using geometry:

$$\begin{aligned}cos^2\left(\frac{\pi}{4}-\gamma\right)&=\frac{(h-v)^2_{xy}}{\tilde{u}^2+(h-v)^2_{xy}}=\frac{\left(\frac{b}{2}\right)^2\left(\frac{h-v}{h}\right)^2}{\left(\frac{b}{2}\right)^2\left(\frac{h-v}{h}\right)^2u^2+\left(\frac{b}{2}\right)^2\left(\frac{h-v}{h}\right)^2}\\ \Rightarrow cos^2\left(\frac{\pi}{4}-\gamma\right)&=\frac{1}{1+u^2}\ .\end{aligned} \tag{A.20}$$

Similarly, solving for $sin^2\left(\frac{\pi}{4}-\gamma\right)$ using geometry results in

$$sin^2\left(\frac{\pi}{4}-\gamma\right)=\frac{1}{1+u^{-2}}\ . \tag{A.21}$$

Finally, regardless of the octant, this results in

$$\frac{\partial\gamma}{\partial u}=\frac{1}{1+u^2}\ . \tag{A.22}$$

If we simplify Equation A.17, we get

$$\gamma=\frac{\pi}{4}-tan^{-1}(-u)\ . \tag{A.23}$$

We can then determine $cos(\gamma)$:

$$\begin{aligned}cos(\gamma)&=cos(\frac{\pi}{4}-tan^{-1}(-u))\\&=\frac{\sqrt{2}}{2}\left(cos(tan^{-1}(-u))+sin(tan^{-1}(-u)\right)\\&=\frac{\sqrt{2}}{2}\frac{1-u}{\sqrt{1+u^2}}\ .\end{aligned} \tag{A.24}$$

Thus the relationship between μ,ξ,u, and v is

$$\begin{aligned}\mu&=cos(\gamma)sin(\theta)\\&=\frac{\sqrt{2}}{2}\frac{1-u}{\sqrt{1+u^2}}\sqrt{1-\xi^2}\ .\end{aligned} \tag{A.25}$$

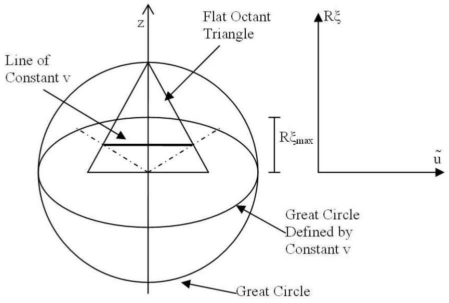

Fig. A.6. View of ellipse from $\gamma = 45°$.

Similarly, sin(γ) is determined as

$$\begin{aligned} sin(\gamma) &= sin(\frac{\pi}{4} - tan^{-1}(-u)) \\ &= \frac{\sqrt{2}}{2}\big(cos(tan^{-1}(-u)) - sin(tan^{-1}(-u)\big) \\ &= \frac{\sqrt{2}}{2}\frac{1+u}{\sqrt{1+u^2}} \ . \end{aligned} \tag{A.26}$$

Therefore, the relationship between η, ξ, u, and v is:

$$\begin{aligned} \eta &= sin(\gamma)sin(\theta) \\ &= \frac{\sqrt{2}}{2}\frac{1+u}{\sqrt{1+u^2}}\sqrt{1-\xi^2} \ . \end{aligned} \tag{A.27}$$

A.3.2 Relationships between ξ, u, and v

Figure A.6 illustrates that a great circle looks like an ellipse when viewed from a perspective not in the plane of the circle. In the $\gamma = 45°$ view, the equation is

$$\left(\frac{h}{v}\right)^2 \frac{\xi^2\tilde{u}^2}{R^2} + \frac{R^2\xi^2}{R^2\xi_{max}^2} = 1 \ . \tag{A.28}$$

This is simplified to

$$\xi^2 = \frac{\left(\frac{v}{h}\right)^2 \xi_{max}^2}{\frac{\tilde{u}^2\xi_{max}^2}{R^2} + \left(\frac{v}{h}\right)^2} \ . \tag{A.29}$$

Equation A.29 shows ξ's dependence on $\tilde{u}$. There is also a dependence on v in ξ_{max}^2 and $\tilde{u}^2$. From Figure A.7, the following geometric relations are derived:

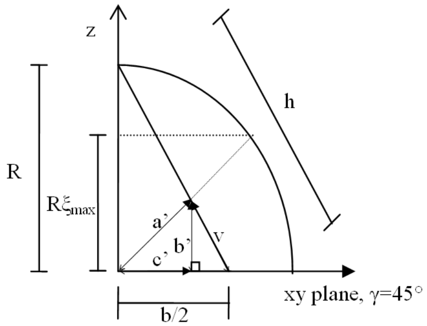

Fig. A.7. A more detailed side view of the triangle.

$$\begin{gathered}\frac{v}{R/\sqrt{2}-c'}=\frac{h}{R/\sqrt{2}}\\ \Rightarrow c'=\frac{R}{\sqrt{2}}\left(\frac{h-v}{h}\right)\ ,\end{gathered}\tag{A.30a}$$

$$\begin{gathered}\frac{b'}{R/\sqrt{2}-c'}=\frac{R}{R/\sqrt{2}}\Rightarrow b'=R-c'\sqrt{2}\\ \Rightarrow b'=R\frac{v}{h}\ ,\end{gathered}\tag{A.30b}$$

$$a'^2=\frac{R^2}{2}\left(\frac{h-v}{h}\right)^2+R^2\left(\frac{v}{h}\right)^2\ ,\tag{A.30c}$$

and

$$\frac{a'}{R}=\frac{b'}{R\xi_{max}}\Rightarrow\xi_{max}=\frac{b'}{a'}\ .\tag{A.30d}$$

For simplicity, we defined

$$d=\left(\frac{v}{h}\right)^2+\frac{1}{2}\left(\frac{h-v}{h}\right)^2\ .\tag{A.31}$$

We solve for ξ^2_{max}:

$$\xi^2_{max}=d^{-1}\left(\frac{v}{h}\right)^2\ .\tag{A.32}$$

The derivative of ξ_{max} with respect to v is

$$\frac{\partial\xi_{max}}{\partial v}=\left(\frac{h-v}{h}\right)\frac{d^{-3/2}}{2h}\ .\tag{A.33}$$

Using Equation A.32, Equation A.29 can be reduced to

$$\xi^2 = \frac{v^2}{(u^2+1)\frac{1}{2}(h-v)^2 + v^2} \; . \tag{A.34}$$

This is then further reduced to

$$\xi = \frac{v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \; . \tag{A.35}$$

Taking the derivative of Equation A.35 with respect to v yields

$$\frac{\partial \xi}{\partial v} = \frac{1}{\sqrt{\frac{u^2+1}{2}(h-v)^2 + v^2}} - \frac{v(v - \frac{u^2+1}{2}(h-v))}{\left(\frac{u^2+1}{2}(h-v)^2 + v^2\right)^{3/2}} \; . \tag{A.36}$$

This is then reduced to

$$\frac{\partial \xi}{\partial v} = \frac{\frac{u^2+1}{2}h(h-v)}{\left(\frac{u^2+1}{2}(h-v)^2 + v^2\right)^{3/2}} \; . \tag{A.37}$$

We also note:

$$1 - \xi^2 = \frac{(u^2+1)\frac{1}{2}(h-v)^2}{(u^2+1)\frac{1}{2}(h-v)^2 + v^2} \tag{A.38}$$

and

$$\sqrt{1-\xi^2} = \frac{\sqrt{\frac{u^2+1}{2}}(h-v)}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \; . \tag{A.39}$$

A.3.3 Representation of μ, η, and ξ in Terms of u and v

Finally, we restate the following derivations for the first octant, where the subscript 1 represents the first octant:

$$\xi_1 = \frac{v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \; , \tag{A.40}$$

$$\eta_1 = \frac{\sqrt{2}}{2}\frac{1-u}{\sqrt{1+u^2}}\frac{\sqrt{\frac{u^2+1}{2}}(h-v)}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \; , \tag{A.41}$$

and

$$\mu_1 = \frac{\sqrt{2}}{2}\frac{1+u}{\sqrt{1+u^2}}\frac{\sqrt{\frac{u^2+1}{2}}(h-v)}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \ . \tag{A.42}$$

After simplification, Equations A.40 , A.41, and A.42 become:

$$\xi_1 = \frac{v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \ , \tag{A.43}$$

$$\mu_1 = \frac{1-u}{2}\frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \ , \tag{A.44}$$

and

$$\eta_1 = \frac{1+u}{2}\frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \ . \tag{A.45}$$

The relationship between the first octant (μ_1, η_1, ξ_1) and the other seven octants can be seen in Table A.1.

Table A.1
Mapping from first octant to other seven octants.

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

A.4 Derivation of the Jacobian

We have explained the relationship between the flat triangle and the spherical coordinates and can now derive the Jacobian relating the two coordinate systems. This is necessary because the following equation describes changing variables from the surface of the sphere to the flat triangle:

$$\int_{\partial\xi} d\xi \int_{\partial\gamma} d\gamma f(\xi,\gamma) = \int_{u_{min}}^{u_{max}} du \int_{v_{min}}^{v_{max}} dv |J| f(u,v) \; , \tag{A.46}$$

where the Jacobian, $|J|$, is defined as

$$|J| = \begin{vmatrix} \frac{\partial\xi}{\partial v} & \frac{\partial\xi}{\partial u} \\ \frac{\partial\gamma}{\partial v} & \frac{\partial\gamma}{\partial u} \end{vmatrix} \; . \tag{A.47}$$

Because of the specific coordinate systems chosen,

$$\frac{\partial\gamma}{\partial v} = 0 \; , \tag{A.48}$$

which simplifies Equation A.47 to

$$|J| = \left| \frac{\partial\xi}{\partial v} \frac{\partial\gamma}{\partial u} \right| \; . \tag{A.49}$$

Inserting Equations A.36 and A.22 into Equation A.49 results in

$$|J| = \frac{h(h-v)}{2\left(\frac{u^2+1}{2}(h-v)^2 + v^2\right)^{3/2}} \tag{A.50}$$

A.5 Putting It All Together

The integration of the linear basis function, which depend on μ, η, and ξ, takes the form of

$$\int_{u_{min}}^{u_{max}} du \int_{v_{min}}^{v_{max}} dv \frac{h(h-v)}{2\left(\frac{u^2+1}{2}(h-v)^2 + v^2\right)^{3/2}} (c_c + c_\mu \mu + c_\eta \eta + c_\xi \xi) \; . \tag{A.51}$$

We insert Equations A.43, A.44, and A.45 into Equation A.51:

$$\begin{aligned}
&\int_{u_{min}}^{u_{max}} du \int_{v_{min}}^{v_{max}} dv \frac{h(h-v)}{2\left(\frac{u^2+1}{2}(h-v)^2+v^2\right)^{3/2}} * (c_c \\
&+ c_\mu \frac{1-u}{2} \frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \\
&+ c_\eta \frac{1+u}{2} \frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}} \\
&+ c_\xi \frac{v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2+v^2}}) \ .
\end{aligned} \tag{A.52}$$

Because we are integrating over a triangle, the limits of integration are not constant. We choose to use a one dimensional Gaussian quadrature in each of the two variables to solve Equation A.52. We combine these sets using in a product quadrature set. On the base flat triangle, we are using the coordinates u and v, where $-1 \leq u \leq 1$ and $0 \leq v \leq h$. On each sub-triangle, we initially divide v using the Gaussian quadrature:

$$\int_u du \int_v dv F(u,v) \approx \int_u du \sum_{j=0}^{J} F(u,v_j) w_{v,j} \ , \tag{A.53}$$

where v_j and $w_{v,j}$ are the one dimensional Gaussian quadrature with J points and $F(u,v)$ is an arbitrary function of u and v. We will do all calculations on a right-side up triangle as shown in Figure A.4. There are examples of sub-triangles being upside down but the nomenclature is the only thing that changes. For each v_j, we determine the range of u (u_{min}, u_{max}):

$$u_{min} = \frac{v_j - \tilde{c_p}}{\tilde{c_p}(1-\frac{v_j}{h})} \ , \tag{A.54a}$$

and

$$u_{max} = \frac{v_j - \tilde{c_n}}{\tilde{c_n}(1-\frac{v_j}{h})} \ , \tag{A.54b}$$

where $\tilde{c_n}$ and $\tilde{c_n}$ were determined from the geometric properties of the flat sub-triangle:

$$\tilde{c_p} \equiv v_{\text{top}} - \frac{2h}{b}\tilde{u}_{\text{top}} \ , \tag{A.55a}$$

and

$$\tilde{c_n} \equiv v_{\text{top}} + \frac{2h}{b}\tilde{u}_{\text{top}}. \tag{A.55b}$$

We then utilize a one dimensional Gaussian quadrature in u over the range (u_{min}, u_{max}):

$$\int_u du \int_v dv F(u,v) \approx \int_v dv \sum_{i=0}^{I} F(u_i, v) w_{u,i} \ , \tag{A.56}$$

where u_i and $w_{u,i}$ are the one dimensional Gaussian quadrature with I points and $F(u,v)$ is an arbitrary function of u and v. Combining Equations A.53 and A.56, we find

$$\int_u du \int_v dv F(u,v) \approx \sum_{j=0}^{J} \sum_{i=0}^{I} F(u_i, v_j) w_{u,i} w_{v,j} \ . \tag{A.57}$$

We use Equation A.57 to evaluate Equation A.52. To a user-determined accuracy, we determine the integration of the linear basis functions over a spherical triangle on the unit sphere:

$$\int_{\Omega_{\text{triangle}}} d\Omega b(\Omega) \approx \sum_{j=0}^{J} \sum_{i=0}^{I} b(u_i, v_j) w_{u,i} w_{v,j} \ , \tag{A.58}$$

where

$$\begin{aligned} b(u,v) = c_c &+ c_\mu \frac{1-u}{2} \frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \\ &+ c_\eta \frac{1+u}{2} \frac{h-v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \\ &+ c_\xi \frac{v}{\sqrt{(u^2+1)\frac{1}{2}(h-v)^2 + v^2}} \ . \end{aligned} \tag{A.59}$$

VITA

Name: Joshua John Jarrell

Address:
Joshua Jarrell
c/o Dr. Marvin Adams
Texas A&M University
3133 TAMU
129 Zachry Engineering Center
College Station, TX 77843-3133

Email Address: joshjarrell@gmail.com, jjarrell@neo.tamu.edu

Education:
Ph.D., Nuclear Engineering, Texas A&M University, 2010

B.S., Nuclear Engineering, Texas A&M University, 2006