A SECOND ORDER LINEAR DISCONTTINUOUS CUT-CELL DISCRETIZATION FOR THE S_N EQUATIONS IN RZ GEOMETRY

A Dissertation

by

MICHAEL SHANE REED

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

Chair of Committee,	Jim Morel
Co-Chair of Committee,	Marvin Adams
Committee Members,	Jean Ragusa
	Jean-Luc Guermond
Head of Department,	Yassin A. Hassan

May 2015

Major Subject: Nuclear Engineering

ABSTRACT

In this dissertation we detail the development, implementation, and testing of a new cut-cell discretization for the discrete ordinates form of the neutron transport equation. This method provides an alternative to homogenization for problems containing material interfaces that do not coincide with mesh boundaries. A line is used to represent the boundary between the two materials in a mixed-cell converting a rectangular mixed-cell into two non-orthogonal, homogeneous cut-cells. The linear-discontinuous Galerkin finite element method (LDGFEM) spatial discretization is used on all of the rectangular cells as well as the non-orthogonal sub-cells. We have implemented our new cut-cell method in a test code which has been used to evaluate its performance relative to homogenization.

We begin by developing the equations and methods associated with the LDGFEM discretization of the transport equation in RZ geometry for a homogenous orthogonal mesh. Next we introduce cut-cell meshes and develop a modification to the LDGFEM equations to account for material interfaces. We also develop methods to account for meshing errors encountered when representing curved material interfaces with linear cell faces.

Finally we present test problems including manufactured solutions, fixed source, and eigenvalue problems for geometries with curvilinear material interfaces. The results of these test problems show the new cut-cell discretization to be second-order convergent for the scalar flux removed from singularities in the solution, as well as significantly more computationally efficient than homogenization.

DEDICATION

For my mother, Lisa Brown

ACKNOWLEDGEMENTS

First of all I would like to thank my family for providing me support over the years. Both my parents and grandparents provided enormous emotional support and guidance as I made my way through adolescence. I would also like to thank my wife Dr. Alexis Reed for supporting me in all that I do. Without all of their support I would not have made it through this long journey.

Next I would like to thank my committee for all of their time and effort. Dr. Jim Morel worked with me tirelessly through the many years it took me to complete this work. His engagement and patience were not always earned but will always be very much appreciated. I would like to thank Dr. Marvin Adams for being a great professor, spending much time with me early in my academic career, and for helping me make contacts for summer jobs that shaped my career. I had the pleasure of taking classes from both Dr. Ragusa and Dr. Guermond and I would like to thank them for serving on my committee as well.

I would also like to thank the mentors I had at Los Alamos National Laboratory over the years that I worked there as a summer student. Dr. Avneet Sood and Dr. Randy Baker were very generous with their time and I learned a tremendous amount about the field of

computational transport from them. Each of these individuals had a significantly positive effect on my professional career.

Finally I would like to thank all of the friends that helped support me in finishing this work after I left school. Completing this work after starting a career proved to be particularly challenging. Without their constant support and encouragement I would have not continued this effort. Dr. Gerry Garino in particular took an enormous interest in ensuring that I completed this work. In addition to being a good friend he has taught me much about life and professionalism. His help has significantly shaped my professional development and for that I am thankful.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vii

LIST OF FIGURES ix

LIST OF TABLES xii

CHAPTER I INTRODUCTION 1

1.1 Use of the transport equation 1
1.2 Solution of the neutron transport equation 5
1.3 Limitations of orthogonal meshes 8
1.4 Summary of previous work 10
1.5 Summary of new cut-cell discretization 13

CHAPTER II DEVELOPMENT OF A 2D RZ TRANSPORT CODE ON ORTHOGONAL MESHES 15

2.1 One-group transport equation in RZ geometry with isotropic scattering 15
2.2 Weighted diamond angular discretization 18
2.3 Linear discontinuous Galerkin finite element spatial discretization 25
2.3.1 Spatial discretization for weighted directions 25
2.3.2 Spatial discretization for non-weighted directions 30
2.4 Angular quadrature 32
2.5 Spatial quadrature 37
2.5.1 Integrations on triangular elements 37
2.5.2 Integration on quadrilateral elements 41
2.5.3 Integration on pentagonal elements 45
2.5.4 Line integrals on polygonal faces 47
2.6 Treatment of the total source term 52
2.6.1 Fixed source computations 52

2.6.2 Eigenvalue computations 53
2.7 Boundary conditions 56
2.8 Solution procedure 59

CHAPTER III DESCRIPTION OF CUT-CELL METHODS 68

3.1 Mesh generation 68
3.2 Numerical cut-cell discretizations 73
3.2.1 Homogenization 74
3.2.2 Continuous flux cut-cell method (CFCM) 77
3.2.3 Discontinuous flux cut-cell method (DFCM) 81
3.2.4 Summary 85
3.3 Interface treatments 86
3.3.1 Mesh refinement (no-treatment=NT) strategy 89
3.3.2 Cell-wise mass preservation (CWMP) treatment 90
3.3.3 Region-wise mass preservation (RWMP) treatment 92
3.3.4 Summary 93

CHAPTER IV TEST PROBLEMS 95

4.1 Test problem techniques and evaluation metrics 95
4.1.1 The method of manufactured solutions 95
4.1.2 Scalar flux norms 96
4.1.3 Computational efficiency 98
4.2 Linear flux manufactured solution 101
4.3 Spatial kink manufactured solution 106
4.4 Fixed source test problem 115
4.5 Eigenvalue test problem 126

CHAPTER V DOCUMENT SUMMARY 142

CHAPTER VI CONCLUSIONS AND FUTURE WORK 146

5.1 Conclusions 146
5.2 Future work 148

REFERENCES 152

APPENDIX A CUT-CELL TYPE DESCRIPTIONS 154

LIST OF FIGURES

Page

Figure 1. Diagram of RZ coordinate system 16

Figure 2. Depiction of a level set angular quadrature on unit half sphere 20

Figure 3. Depiction of a Gauss-Chebyshev triangular quadrature set on the unit half sphere 33

Figure 4. Graphical depiction of S_8 Gauss-Chebyshev angular quadrature 36

Figure 5. Coordinate transformation for triangular element integration 38

Figure 6. Coordinate system transformation for quadrilateral element integration 42

Figure 7. Decomposition of pentagonal elements into three triangular elements 46

Figure 8. Depiction of line integral over cell face 47

Figure 9. Flow diagram for numerical solution of RZ transport equation for fixed source problem 60

Figure 10. Graphical representation of a RZ spatial mesh 64

Figure 11. Sphere encased in RZ cylinder 69

Figure 12. Linear interface meshing procedure 71

Figure 13. Sketch of representative mixed-cell 73

Figure 14. Homogenization sweeping procedure 76

Figure 15. CFCM sweeping procedure 79

Figure 16. DFCM sweeping procedure 82

Figure 17. Mesh interface residual 87

Figure 18. Behavior of meshing characteristics for linear flux test problem 104

Figure 19. Numerical conditioning of cut-cell methods applied to linear flux test problem....106

Figure 20. Graphical representation of angular flux profile in spatial kink manufactured solution test problem....108

Figure 21. Graphical representation of material distribution and meshing for spatial kink manufactured solution test problem....111

Figure 22. Method performance for scalar flux convergence in spatial kink manufactured solution test problem....112

Figure 23. Computational efficiency for spatial kink test problem....114

Figure 24. Graphical depiction of problem setup for fixed source test problem....116

Figure 25. Scalar flux convergence with S_8 for fixed source test problem....118

Figure 26. Scalar flux convergence with S_{16} for fixed source test problem....119

Figure 27. Scalar flux convergence with S_{32} for fixed source test problem....120

Figure 28. Scalar flux convergence with S_{48} for fixed source test problem....122

Figure 29. Computational efficiency for scalar flux with S_{48} for fixed source test problem....125

Figure 30. Graphical representation of setup for eigenvalue test problem....128

Figure 31. Scalar flux convergence with S_4 for eigenvalue test problem....129

Figure 32. Scalar flux convergence with S_8 for eigenvalue test problem....130

Figure 33. Scalar flux convergence with S_{16} for eigenvalue test problem....131

Figure 34. Scalar flux convergence with S_{32} for eigenvalue test problem....133

Figure 35. Computational efficiency for scalar flux convergence with S_{32} for eigenvalue test problem....134

Figure 36. Eigenvalue convergence with S_4 for eigenvalue test problem....136

Figure 37. Eigenvalue convergence with S_8 for eigenvalue test problem....137

Figure 38. Eigenvalue convergence with S_{16} for eigenvalue test problem 138

Figure 39. Eigenvalue convergence with S_{32} for eigenvalue test problem 139

Figure 40. Computational efficiency for eigenvalue computation with S_{48} for eigenvalue test problem 140

Figure 41. Visual depiction of quadrilateral/pentagon type cut-cells 156

Figure 42. Visual depiction of quadrilateral/quadrilateral type cut-cells.....157

LIST OF TABLES

Page

Table 1. Gauss-Chebyshev S_8 angular quadrature 35

Table 2. Triangular spatial quadrature set 41

Table 3. Product Gaussian quadrature set for spatial integration on quadrilaterals 45

Table 4. Two-point Gaussian quadrature on bi-unit line segment 51

CHAPTER I

INTRODUCTION

1.1 Use of the transport equation

Hastened significantly by the discovery of the neutron by Chadwick in 1932, significant engineering feats have been accomplished during the last century through the harnessing of nuclear physics and radiation transport [1]. Some of the most significant applications involve the design and operation of nuclear reactors, the harnessing of radiation for medical applications, the application of radiation physics for the purposes of national defense, and a host of other industrial applications [2]. In each case the ability to model and predict the behavior of systems governed by radiation physics is of primary concern to the application.

The fundamental equation used to describe the distribution and behavior of nuclear particles interacting with a system of background atoms or nuclei is known as the linearized Boltzmann transport equation (LBTE), although many nonlinear forms of the equation exist [3]. This equation is used as the starting point for modeling neutron, photon, and charged particle interactions. In this work we will consider a linear version of the neutron transport equation though nonlinear versions of the Boltzmann equation occur in the descriptions of other systems and most of the underlying results are relevant to these forms as well. The neutron transport equation is a physical statement of particle

conservation that forms the basis of essentially all neutron modeling. A compact form of the neutron transport equation may be written as

$$\frac{1}{v(E)}\frac{\partial \psi(\underline{r},E,\underline{\Omega},t)}{\partial t}+\underline{\Omega}\cdot\underline{\nabla}\psi(\underline{r},E,\underline{\Omega},t) \\ +\sigma(\underline{r},E,t)\psi(\underline{r},E,\underline{\Omega},t)=Q(\underline{r},E,\underline{\Omega},t), \tag{1.1}$$

with associated boundary conditions that provide the incoming angular flux along the boundaries of the problem domain

$$\psi(\underline{r}_{boundary},\mathrm{E},\underline{\Omega},\mathrm{t})=\psi_{boundary}(\underline{r}_{boundary},\mathrm{E},\underline{\Omega},\mathrm{t}),\ \underline{\Omega}\cdot\underline{n}_{boundary}<0, \tag{1.2}$$

as well as an appropriate initial boundary condition

$$\psi(\underline{r},\mathrm{E},\underline{\Omega},\mathrm{t}_0)=F(\underline{r},\mathrm{E},\underline{\Omega}). \tag{1.3}$$

A derivation and thorough description of this equation may be found in [4]. The physical interpretation of this equation is facilitated by considering a differential phase space cell represented as the product of independent differentials $dV\ dE\ d\Omega$. If we consider multiplication of Eq. (1.1) by this quantity then we may easily associate each term of the equation with a physical interpretation. The first term in the equation denotes the time rate of change of the particle population existing in the phase space cell. The second term denotes the net rate at which particles are removed from the phase space cell due to spatial streaming (outflow of particles from dV minus inflow of particles into dV). The third term represents the rate at which particles incur collisions, thus being removed from the phase space cell. The right hand side is written as a general, problem-dependent

source term comprised of mechanisms giving rise to new particles in the phase space such as in-scatter, fixed source terms, fission, delayed neutron precursor ingrowth, etc. The fundamental unknown in the neutron transport equation is the angular flux of particles denoted as ψ. This quantity is simply the product of the particle density and speed denoted as v in the equation. The total macroscopic cross section is indicated by σ and is a material property denoting the probability that a particle incurs an interaction per unit path length traversed. The argument $\underline{r}$ represents a point in three-dimensional space, $\underline{\Omega}$ is a unit vector describing particle direction of travel, E is used to denote particle energy, and t represents time.

The source term appearing on the right side of Eq. (1.1) takes various forms defining different problem types. In a fixed source computation with scattering the source term takes the form

$$\begin{aligned} Q_{fixed-source}\left(\underline{r},E,\underline{\Omega},t\right) &= S\left(\underline{r},E,\underline{\Omega},t\right) \\ +\int dE' \int d\underline{\Omega}'\, \sigma_s\left(\underline{r},E'\to E,\underline{\Omega}'\to\underline{\Omega}\right)\psi\left(\underline{r},E',\underline{\Omega}',t\right). \end{aligned} \tag{1.4}$$

Here the doubly differential scattering cross section is given as σ_s, and its product with the angular flux integrated over all incident angles and energies yields the rate at which particles emerge into the phase space cell $dV\,dE\,d\Omega$ due to scattering [4]. The term S denotes an inhomogeneous source term associated with the underlying media. This term is used to model sources that are material-dependent and do not depend on the neutron

flux itself. One example includes spontaneous fission sources such as ^{252}Cf. This form of the equation is used to model non-multiplying systems. For problem types modeling dynamic systems such as pulsed neutron sources the full time-dependent form is used. For many problem types the underlying system behavior is steady state and a time-independent form is used.

In k-eigenvalue computations, a time-independent form of the transport equation is solved where the source term on the right side of Eq. (1.1) takes the form

$$\begin{aligned} Q_{eigenvalue}\left(\underline{r},E,\underline{\Omega}\right) &= \frac{\chi(E)}{k}\int dE'\,\nu\sigma_f\left(\underline{r},E'\right)\psi\left(\underline{r},E',\underline{\Omega}'\right) \\ &+\int dE'\int d\underline{\Omega}'\,\sigma_s\left(\underline{r},E'\to E,\underline{\Omega}'\to\underline{\Omega}\right)\psi\left(\underline{r},E',\underline{\Omega}'\right). \end{aligned} \tag{1.5}$$

Here the quantity k is known as the multiplication factor, the quantity χ denotes the fraction of fission neutrons born with energy E, and the product $\nu\sigma_f$ denotes the neutron production cross section [4]. The quantity k is the single most important parameter governing the behavior of multiplying systems and may be interpreted physically as the mean number of neutrons born from fission following the insertion of a single neutron into the system of interest. The quantity ν represents the mean number of neutrons born per fission, and the fission cross section σ_f denotes the probability of fission per unit path length for a neutron traversing the system. This form of the neutron transport equation is used to model multiplying systems, most notably a nuclear reactor.

1.2 Solution of the neutron transport equation

In general the transport equation is an equation in seven independent unknowns: three in space, two in angle, one in energy, and one in time. Since the source term present on the right hand side of Eq. (1.1) involves an integral of the angular flux and the left hand side involves partial derivatives of the angular flux, the equation is known as a partial integro-differential equation. Save for very simplified cases, there exist no analytical solutions to this equation, and the solution requires the use of various numerical techniques. Though several assumptions are made in deriving Eq. (1.1), there exist many regimes in which the equation provides highly accurate descriptions of particle behavior [4].

Probably the most widely used deterministic method for solving the transport equation is known as the discrete ordinates, or S_N method [5]. The method is known as the discrete ordinates method because it relies on solving the transport equation for particles travelling along a collection of discrete directions. For complicated fundamental physics equations such as Eq. (1.1) the standard solution technique involves the discretization of the analytical equations into a set of linear algebraic equations that may be solved on a computer. The S_N method is a technique by which the angular dependence of the transport equation is discretized by requiring that the transport equation itself hold for a discrete collection of angles. With respect to the spatial variable, discretization may be thought of as slicing the problem domain into a collection of sub-domains known as

spatial cells. This process is known as spatial discretization, and a common method employed is that known as the finite element method (FEM) [6]. The FEM converts the problem domain into a collection of simple spatial elements. The fundamental unknown is then represented by an expansion in space of a particular polynomial form on each individual element. The underlying governing equations are then used to obtain a set of equations that relate the unknowns in one cell to the expansion in other cells as well as known boundary conditions. Elements are linked to one another by spatial proximity and an underlying connectivity that is prescribed again based on the underlying physics. This process is widely used in all of engineering because of its relative simplicity and the fact that it reduces very complex systems to a large system of linear algebraic equations that may be readily solved on a modern computer. A description of the FEM applied to various forms of the neutron transport equation may be found in [7], [8], or [9].

By far the most popular spatial elements chosen for the FEM are those formed from an orthogonal meshing of the problem domain. Thus in 1D a collection of connected slab elements is most common, in 2D a collection of juxtaposed rectangles is most common, and in 3D a collection of brick elements is most common. In each case the underlying numerical discretization proves to be exceedingly simple for orthogonal meshes relative to more general polyhedral meshes. In addition to greatly simplifying method development and implementation, orthogonal meshes have significant advantages with respect to parallelization on a computer system. Essentially the orthogonal nature of the

spatial mesh defines very separable work that may be easily distributed across processors with little need for communication overhead [10]. For this reason orthogonal FEM meshing remains very common in the radiation transport community despite the advent of general polygonal mesh generation and the associated spatial discretizations.

Once a set of equations are prescribed on each cell within the spatial mesh the discrete ordinates equation is solved via what is known as sweeping. Sweeping is a term used to denote the characteristic stepping procedure employed by which the transport equation is solved in succession on adjacent elements. For a particular angle in the angular mesh the flux of particles entering a particular cell is known from a boundary condition on the domain. The underlying spatial discretization is used in conjunction with these unknowns to determine the angular flux within that cell and thus the angular flux that leaves the cell and enters an adjacent cell is known. Having now prescribed the angular flux entering the adjacent cell the spatial discretization equations may be used to find the angular flux within that element. This procedure continues until the angular flux for the current angle is known throughout the entire problem domain. A single occurrence of this procedure for a particular angle is known as a spatial transport sweep. We note that if one formulates the spatial discretization equations for all elements into a single matrix equation the underlying matrix is block lower triangular. This means that the solution may be decomposed into the sweeping procedure described above with subsequent savings of computer memory and computational complexity.

1.3 Limitations of orthogonal meshes

Despite significant advantages relative to general polyhedral meshes, orthogonal meshes have one fundamental shortfall. Very few real-world geometries are themselves orthogonal in nature. In general, curved surfaces are much more common and indeed represent most items of interest in the world we live in. Examples of objects of interest for the transport equation include: a set of cylindrical annuli comprising a fuel pin found in a nuclear reactor or the surface of a biological organ that is to receive radiation therapy. For these applications the underlying material distribution cannot be exactly described by a collection of orthogonal elements. Considering such an object an orthogonal meshing will result in cells that exhibit a material discontinuity along a curved interface internal to the cell.

The standard approach for dealing with the meshing of curvilinear domains with orthogonal meshes involves a collection of mesh refinement and homogenization. Homogenization is the term used to refer to the procedure whereby a heterogeneous spatial cell is replaced with a homogeneous cell having material properties that correspond to some form of averaging of the original constituents [11]. The fundamental problem with homogenization concerns its degradation of numerical method accuracy. Truncation error is a term used to describe numerical error in the limit as the mesh size approaches zero. In this limit the error made by a numerical method using a step size h may typically be shown to be bounded by a constant multiplied by some power of h.

The analysis of methods in this limit is known as asymptotic truncation error analysis, and the results are typically written in "Big-O" notation as

$$\tau \le ch^n \Rightarrow \tau = O\left(h^n\right). \tag{1.6}$$

An in-depth discussion of truncation error analysis is developed in [12]. Referring to Eq. (1.6) we note that the truncation error is given by τ, the step size is given by h, the order of the method is represented by the parameter n, and the error bound is given for some constant c (independent of mesh size). The truncation error analysis details the behavior of the numerical method as the mesh size is reduced. Thus in the asymptotic limit reducing the mesh size by a factor of two results in the reduction of discretization error by a factor of two for a first order method, a factor of four for a second order method, and so on.

Homogenization is itself a first-order method locally. This means that in the asymptotic limit a reduction in meshing error of a factor of two requires a factor of two increased computational work in 1-D, a factor of four in 2-D, and a factor of eight in 3-D. This unfortunate circumstance has driven the development of high order spatial discretizations that are more computationally efficient. The linear discontinuous (LD) spatial discretization is one such method that is second order convergent in the asymptotic limit. Thus in the asymptotic limit a reduction in meshing error of a factor of two requires a factor of $\sqrt{2}$ more computational work in 1-D, a factor of 2 in 2-D, and a

factor of $2\sqrt{2}$ in 3-D. This additional computational efficiency can translate to reduced computer runtime, or more typically better solutions (lower error) in the same run time as compared to a first order method. However, when applied to problems exhibiting material interfaces treated with homogenization, the resulting method is degraded to be locally first-order convergent. In summary, the use of homogenization degrades spatial discretization order of convergence and serves as a major impediment to computational efficiency in problems with material interfaces that do not coincide with mesh boundaries. It is upon a solution of this central problem that this dissertation shall focus.

1.4 Summary of previous work

To date very little direct work in the radiation transport community has been done to confront the underlying dilemmas associated with homogenization. Instead, the community has employed a host of treatments that reduce the ill effects of homogenization but do not solve the underlying problem. The one exception involves the use of general polyhedral unstructured meshing. However, for the reasons discussed in section 1.2 it is still desirable to find an alternative to homogenization for orthogonal meshes. Nonetheless, we make particular note of this exception because our new method shall rely upon a very limited implementation of the same principle.

In recent years much work has been done to employ a technique known as adaptive mesh refinement (AMR) in which the underlying mesh is non-uniformly refined in resolution in such a way to specifically target features with preferential mesh resolution

[13]. For spatial meshing this typically involves the relative refinement of material interfaces and optically thick cells. This technique when used in conjunction with homogenization can significantly improve computational efficiency. Still, all spatial discretization schemes are locally first-order convergent in homogenized cells, and the fundamental problem has not been solved. We note that AMR is in general highly advantageous, and it is desirable that any interface reconstruction scheme be capable of seamlessly integrating into spatial AMR techniques.

Though little work has been done in the field of neutron transport, the subject of interface reconstruction has seen much work in the field of fluid mechanics beginning with the Navier-Stokes equation, which is itself derived from the nonlinear Boltzmann transport equation. In contrast to the neutron transport equation, this equation is used to describe a distribution of particles that interact with themselves. In fluid mechanics Cartesian grids have proven useful relative to boundary-conforming meshes due to simplicity in automated mesh generation and refinement flexibility. In these applications so called cut-cell techniques have been created in which local modifications to the mesh and discretizations are made in the presence of cells containing material discontinuities. The cells in question are converted into cut-cells where the “cut” representing the material interface forms distinctive spatial domains in the original cell. The resulting sub-cells are referred to as cut-cells. This technique can be traced back to a paper by Purvis published in 1979 [14]. A more modern overview of the approach as applied in fluid mechanics is given by Ingram [15]. The cut-cell method has been widely applied in

the field of fluid mechanics to deal with the material interfaces occurring between rigid bodies and the fluid flow around them [16-18]. In particular, Bouchon and his collaborators developed a second-order-convergent method for the solution of the Navier-Stokes equations that relies on cut-cell meshes [19]. The work done in fluid mechanics serves as the primary inspiration for the work described in this thesis.

As part of this research some initial work was done exploring non-cut-cell based alternatives to homogenization. In an attempt to improve the accuracy of the LANL-developed S_N code PARTISN, we developed, implemented, and tested a first-order interface reconstruction technique based upon the diamond difference spatial discretization [20]. This method is equivalent to standard diamond difference on orthogonal grids but with appropriate modifications in the underlying material properties. The angular flux is assumed linear within a cell, while cross sections are assumed piece-wise constant with a discontinuity coinciding with the material interface. The underlying method was shown to be equivalent to flux-weighting the underlying cross sections. The method was implemented in one-, two-, and three-dimensional geometries and was shown to be significantly more accurate than standard volume-weighted homogenization. For completeness, an analog of this approach is implemented in this work for comparison purposes.

1.5 Summary of new cut-cell discretization

In this dissertation we introduce a new cut-cell discretization for the S_N equations in RZ geometry. This cut-cell method begins with an orthogonal mesh but modifies the mesh by creating cut-cells in the geometry to account for the presence of curvilinear material interfaces. The standard linear-discontinuous finite element method is applied in all orthogonal cells, and modification is made to account for the presence of cells comprised of two distinct material regions. This method is shown to be second-order convergent for the scalar flux and also significantly more computationally efficient than the standard approach of homogenization.

In Section 2 we develop a numerical method for solution of the one-group, steady state transport equation in RZ geometry for orthogonal finite elements. After straightforward modification, the equations generated are applicable to general polygons in 2-D. In Section 3 we introduce two new cut-cell discretizations for the RZ transport equation and show how the discretizations developed in Section 2 may be modified to account for the presence of cut-cells. We also detail the homogenization method which is the historical standard technique for dealing with heterogeneous mesh cells. Next we detail three methods for accounting for mass discrepancies that arise from linearizing a material interface. In Section 4 we present the results of test problems including two manufactured solutions, a fixed source test problem, and an eigenvalue test problem. In

sections 5 and 6 we summarize the results, make some conclusions, and enumerate some areas for future work.

CHAPTER II

DEVELOPMENT OF A 2D RZ TRANSPORT CODE ON ORTHOGONAL MESHES

In this section we introduce the one-group RZ transport equation with isotropic scattering with the goal of developing the equations and methods needed to create a computational code for solving the transport equation on homogeneous orthogonal grids. In section 3.2 we will develop modifications to these methods that account for cells with material interfaces that do not coincide with orthogonal mesh boundaries.

2.1 One-group transport equation in RZ geometry with isotropic scattering

In order to focus this research on the development of a cut-cell discretization we will employ the steady state, mono-energetic (one-group), transport equation with isotropic scattering. While these simplifications are not strictly necessary, it will greatly simplify the work required to develop and implement our new cut-cell discretization. We begin development of our basic equations with a picture of the cylindrical coordinate system we wish to employ as shown in Figure 1.

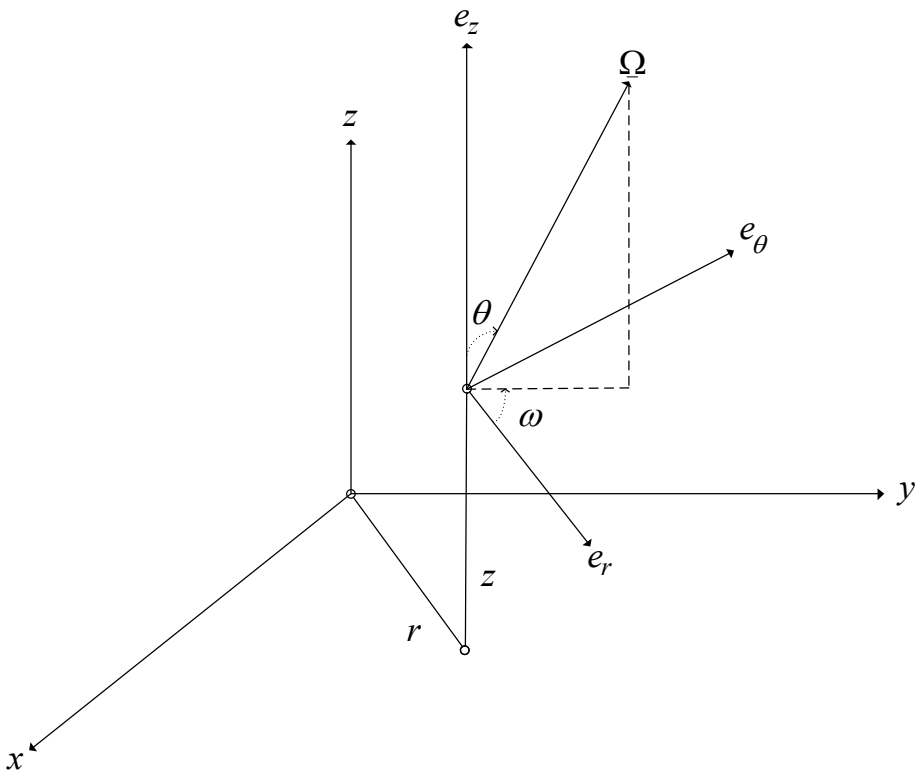

Figure 1. Diagram of RZ coordinate system

With this coordinate system we write the one-group, steady state, direction and space dependent RZ transport equation in conservation form assuming isotropic fixed and scattering sources as:

$$\frac{\mu}{r}\frac{\partial}{\partial r}r\psi\left(r,z,\vec{\Omega}\right)-\frac{1}{r}\frac{\partial}{\partial\omega}\eta\psi\left(r,z,\vec{\Omega}\right)+\xi\frac{\partial}{\partial z}\psi\left(r,z,\vec{\Omega}\right)+\sigma\left(r,z\right)\psi\left(r,z,\vec{\Omega}\right) = Q\left(r,z\right) \tag{2.1}$$

Here the unknown angular flux is given by ψ , the total source is Q , the total cross section is σ , and the angular variable ω is as shown in the picture above.

The direction of particle travel is denoted by the unit vector $\underline{\Omega}$, where its directional cosines are

$$\begin{aligned} \mu &\equiv \underline{\Omega} \cdot \underline{e}_r = \sin\theta\cos\omega, \\ \eta &\equiv \underline{\Omega} \cdot \underline{e}_\theta = \sin\theta\sin\omega, \\ \xi &\equiv \underline{\Omega} \cdot \underline{e}_z = \cos\theta. \end{aligned} \tag{2.2}$$

The total source term for a fixed source problem may be expressed as a sum of a scattering source and an inhomogeneous source term by simplifying Eq. (1.4) to obtain

$$Q(r,z) = \sigma_s \frac{\phi(r,z)}{4\pi} + \frac{S(r,z)}{4\pi}. \tag{2.3}$$

Likewise, the total source term for an eigenvalue problem is found via simplification of Eq. (1.5) to obtain:

$$Q(r,z) = \sigma_s \frac{\phi(r,z)}{4\pi} + \frac{1}{k}\nu\sigma_f \frac{\phi(r,z)}{4\pi}. \tag{2.4}$$

Here the term σ_s denotes the macroscopic scattering cross section, S denotes the inhomogeneous source term, ν gives the mean number of neutrons per fission, σ_f is the fission cross section, and k is the multiplication factor of the system . The variable ϕ denotes the scalar flux and is related to the angular flux via

$$\phi = \int_{4\pi} d\Omega \psi(r,z,\underline{\Omega}). \tag{2.5}$$

We note that the left side of Eq. (2.1) consists of a set of independent first-order partial differential equations (PDE's) for each direction that are easily inverted. All coupling

between directions occurs in the source term. Thus the equation is iteratively solved by inverting the left side while lagging the source terms. This is commonly called source iteration. This procedure requires a starting angular flux iterate which is typically taken as zero.

The discrete ordinates method (S_N) is characterized by collocating the transport equation at a set of quadrature directions on the unit sphere and use of an associated quadrature formula to compute the scalar flux [5].

In order to solve Eq. (2.1) we will discretize the equation in both space and angle. In section 2.2 we discretize in angle and in section 2.3 we discretize in space. In general the energy and time variables must be discretized as well. In an effort to focus on the cut-cell spatial discretization we have restricted this work to steady state problems with a single energy group.

2.2 Weighted diamond angular discretization

The existence of the angular derivatives appearing in Eq. (2.1) requires some form of angular discretization. We shall proceed to develop an angular discretization by presuming a level set angular quadrature on the unit half sphere. In this case the term level set indicates that angular ordinates are comprised of a set of distinct rows of

constant ξ directional cosine, each with various values of μ directional cosine. An image depicting an angular quadrature set of this form is given in Figure 2 below.

In Figure 2 we have shown a total of N distinct ξ levels with an equal amount of levels for both negative and positive directions. We restrict ourselves to symmetric level sets in which the parameter N is even. On each ξ level we have a set of angles formed by allowing the μ directional cosines to take on various values symmetric about zero. We number the ξ levels from negative to positive starting with ξ level one and proceeding to ξ level N. For a given ξ level n the various values of μ are labeled from least to greatest with the index m_f^n being the first angular direction and the index m_l^n being the last angular direction. We have introduced the concept of angular cell center indices and angular edge indices by enumerating each with integer indices and half-integer indices respectively.

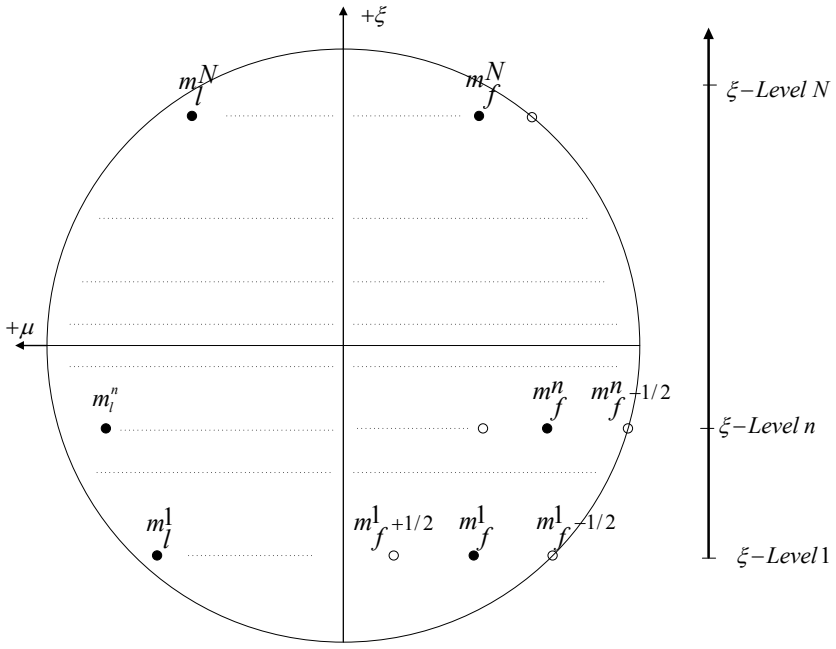

Figure 2. Depiction of a level set angular quadrature on unit half sphere

The standard weighted diamond angular discretization for the RZ transport equation given by (2.1) may now be written as

$$\frac{\mu_m}{r}\frac{\partial}{\partial r} r\psi_m - \frac{\alpha_{m+1/2}\psi_{m+1/2} - \alpha_{m-1/2}\psi_{m-1/2}}{r\, w_m} + \xi_m \frac{\partial}{\partial z}\psi_m + \sigma\psi_m = Q_m, \tag{2.6}$$

where m is the angular index [4]. Equation (2.6) denotes a balance over the angular cell associated with index m. Integral values of m are associated with the cell averages and half-integer values are associated with cell edges. The α coefficients are constructed in such a way as to preserve the constant solution. They are defined as

$$\begin{aligned} \alpha_{m_f^n-1/2} &= 0 \\ \alpha_{m+1/2} &= \alpha_{m-1/2} + \mu_m w_m \qquad . \\ n = 1 \to N \qquad & m = m_f^n \to m_l^n \end{aligned} \tag{2.7}$$

Here we have used m_f^n to represent the first weighted direction on the n'th ξ level and m_l^n is the index of the last weighted direction on the n'th ξ level. The parameter w_m represents the quadrature weight associated with direction m. As written in Eq. (2.7) we have assumed that the angular quadrature set has a total of N discrete ξ cosines. Likewise, for an arbitrary ξ cosine direction indexed as n we have assumed a total of $m_f^n - m_l^n + 1$ discrete μ cosines.

We relate the angular cell-edge fluxes appearing in Eq. (2.6) to the angular cell-center fluxes using the weighted diamond relation:

$$\begin{aligned} \psi_{m+1/2} &= \tau_m^{-1}\psi_m - \tau_m^{-1}\left(1-\tau_m\right)\psi_{m-1/2} \\ \psi_{m_f^n-1/2} &= \psi_{s,n} \qquad . \\ n = 1 \to N \qquad & m = m_f^n \to m_l^n \end{aligned} \tag{2.8}$$

The τ weighting factors used in Eq. (2.8) are defined with the relation set

$$\begin{aligned} \tau_m &= \left(\mu_m - \mu_{m-1/2}\right)/\left(\mu_{m+1/2} - \mu_{m-1/2}\right) \\ \mu_{m+1/2} &= \sin\left(\theta_n\right)\cos\left(\omega_{m+1/2}\right) \\ \omega_{m+1/2} &= \omega_{m-1/2} - \pi C_n w_m \\ n = 1 \to N \quad m = m_f^n \to m_l^n \quad & \omega_{m_f^n-1/2} = \pi. \end{aligned} \tag{2.9}$$

The term θ_n is the angle appearing in Figure 1 for the set of quadrature directions defined by ξ level n. The numerical value may be computed from

$$\theta_n = \cos^{-1}\left(\xi_n\right). \tag{2.10}$$

The factor C_n is a normalization constant chosen such that the sum of the quadrature weights multiplied by that constant on level n is unity according to

$$C_n = \frac{1}{\sum_{i=m_l^n}^{m_f^n} w_i}. \tag{2.11}$$

The angles denoted $\omega_{m-1/2}$ and $\omega_{m+1/2}$ are the angles associated with the angular cell edges for direction m as defined in Figure 1. The starting value for the angle ω on each ξ level is $\omega_{m_f^n-1/2} = \pi$ and is associated with the starting direction flux. Given this angle the starting value of μ on each level, $\mu_{m_f^n-1/2}$ may be computed from

$$\mu_{m_f^n-1/2} = -\sqrt{1-\xi_n^2}. \tag{2.12}$$

In Eq. (2.8) we have used $\psi_{s,n}$ to indicate the so-called starting direction flux. The starting direction flux which is the value of $\psi_{m_f^n-1/2}$ on each ξ level and is the angular flux associated with the direction $\omega = \pi$. When we evaluate Eq. (2.1) at $\omega = \pi$ we get the following equation for the starting direction flux on each ξ level:

$$\mu_n \frac{\partial}{\partial r}\psi_{s,n} + \xi_n \frac{\partial}{\partial z}\psi_{s,n} + \sigma(r,z)\psi_{s,n} = Q_{s,n}. \tag{2.13}$$

It is important to note that the starting direction flux equation is a Cartesian equation. In particular, this equation is identical to that used for neutron transport in XY geometry. When employing an angular quadrature to update the scalar flux, the starting direction carries no weight and thus does not appear on the right side of (2.6). Its presence is necessary solely to initialize the angular differencing procedure introduced in Eq. (2.6) to discretize the angular variable. In order to solve for the weighted directions on each xi-level, we must first solve the XY transport equation for the starting direction. The starting direction equation given by Eq. (2.13) has no angular derivative and thus requires no further angular discretization. The spatial discretization required to solve this equation will be developed in section 2.3.2.

If we make use of the weighted diamond in angle extrapolation given by Eq. (2.8) in order to eliminate the angular outflow edge value of the angular flux within Eq. (2.6) we obtain:

$$\frac{\mu_m}{r}\frac{\partial}{\partial r} r\psi_m - \frac{\frac{1}{\tau_m}\alpha_{m+1/2}\psi_m - \left(\frac{1-\tau_m}{\tau_m}\alpha_{m+1/2} + \alpha_{m-1/2}\right)\psi_{m-1/2}}{r\, w_m} \tag{2.14}$$
$$+\xi_m \frac{\partial}{\partial z}\psi_m + \sigma\psi_m = Q_m.$$

The recursion formula given by Eq. (2.8) defines a stepping procedure whereby angular outflows are determined and used as angular inflows for the next angle starting with the

starting direction fluxes given by Eq. (2.13). Since the angular inflow edge fluxes are known we rearrange Eq. (2.14) by moving them to the right hand side to obtain

$$\begin{aligned}&\frac{\mu_m}{r}\frac{\partial}{\partial r}r\psi_m - \frac{\alpha_{m+1/2}}{r\,\tau_m w_m}\psi_m + \xi_m\frac{\partial}{\partial z}\psi_m + \sigma\psi_m \\ &\quad = Q_m - \frac{\left(\frac{1-\tau_m}{\tau_m}\alpha_{m+1/2} + \alpha_{m-1/2}\right)}{r\,w_m}\psi_{m-1/2}.\end{aligned} \tag{2.15}$$

To simplify things we now define some coefficients that simplify the angular differencing terms above as follows.

$$\gamma_m \equiv \left(1-\tau_m\right)\tau_m^{-1}w_m^{-1}\alpha_{m+1/2} + w_m^{-1}\alpha_{m-1/2} \tag{2.16}$$

$$\beta_m \equiv \alpha_{m+1/2}\tau_m^{-1}w_m^{-1} \tag{2.17}$$

These definitions allow us to write Eq. (2.15) in the more compact form

$$\frac{\mu_m}{r}\frac{\partial}{\partial r}r\psi_m - \frac{1}{r}\beta_m\psi_m + \xi_m\frac{\partial}{\partial z}\psi_m + \sigma\psi_m = Q_m - \frac{1}{r}\gamma_m\psi_{m-1/2}. \tag{2.18}$$

This equation and the underlying auxiliary angular differencing procedure are the angularly discretized RZ transport equation. This equation shall be the starting point for the spatial discretization described in section 2.3.1.

In this section we have developed an angular discretization for the RZ transport equation that depends on an angular quadrature set with angles arranged in levels of constant directional ξ cosines, with each individual level comprised of a set of angles of differing μ cosines. The solution algorithm takes the form of stepping procedure

whereby angular fluxes are computed a single ξ level at a time. A form of the XY geometry transport equation is solved for the starting direction flux to initiate the procedure. The angular fluxes are evaluated for the angular cell centered directions via Eq. (2.18) and then the weighted diamond relation given by Eq. (2.8) is used to determine the angular edge outflow fluxes. These fluxes are the angular edge inflow fluxes for the next direction on the current ξ level and thus a marching procedure for evaluating angular fluxes for a given ξ level is defined. The angular fluxes between successive ξ levels are coupled only through the scalar flux and thus the above procedure is repeated for all ξ levels present in the angular quadrature set.

2.3 Linear discontinuous Galerkin finite element spatial discretization

Eq. (2.13) and Eq. (2.18) are angularly discretized, spatially analytic representations of particle balance for RZ geometry. In this section we apply a linear discontinuous Galerkin finite element method (LDGFEM) discretization in space to each equation.

2.3.1 Spatial discretization for weighted directions

In order to spatially discretize Eq. (2.18) we multiply through by an arbitrary weight function b_i, and then integrate over the volume of a spatial cell. In RZ geometry this is equivalent to multiplying through by $2\pi r$, and then integrating over the area of the cell. We omit the factor of 2π since it appears in all terms. Doing this we obtain

$$\int_{A_{cell}} dA\, r\, b_i \left[\frac{\mu_m}{r} \frac{\partial}{\partial r} r\psi_m - \frac{1}{r} \beta_m \psi_m + \xi_m \frac{\partial}{\partial z} \psi_m + \sigma \psi_m = Q_m - \frac{1}{r} \gamma_m \psi_{m-1/2} \right]. \quad (2.19)$$

We now use the Divergence Theorem on the spatial leakage terms. If we consider the spatial streaming terms in Eq. (2.19) separately for a moment we note that we may write

$$\int_{A_{cell}} dA\, r\, b_i \left[\frac{\mu_m}{r} \frac{\partial}{\partial r} r\psi_m + \xi_m \frac{\partial}{\partial z} \psi_m \right], \quad (2.20)$$

as

$$\int_{A_{cell}} dA\, b_i \left[\mu_m \frac{\partial}{\partial r} r\psi_m + \xi_m r \frac{\partial}{\partial z} \psi_m \right]. \quad (2.21)$$

Using the divergence operator this may be written as

$$\int_{A_{cell}} dA\, b_i \left[\underline{\Omega}_m \cdot \underline{\nabla} r\psi_m \right]. \quad (2.22)$$

Now we consider the product rule applied to the divergence operator. We note that

$$\underline{\Omega}_m \cdot \underline{\nabla} \left(b_i\, r\, \psi_m \right) = b_i\, \underline{\Omega}_m \cdot \underline{\nabla} \left(r\, \psi_m \right) + r\, \psi_m\, \underline{\Omega}_m \cdot \underline{\nabla} \left(b_i \right). \quad (2.23)$$

Using this expansion we may rewrite the spatial streaming term as

$$\int_{A_{cell}} dA\, b_i \left[\underline{\Omega}_m \cdot \underline{\nabla} r\psi_m \right] = \int_{A_{cell}} dA\, \underline{\Omega}_m \cdot \underline{\nabla} b_i\, r\, \psi - \int_{A_{cell}} dA\, r\psi\, \underline{\Omega}_m \cdot \underline{\nabla} b_i. \quad (2.24)$$

Using this for the streaming term we may rewrite Eq. (2.19) as

$$\begin{aligned} \int_{A_{cell}} dA\, \underline{\Omega}_m \cdot \underline{\nabla} b_i\, r\, \psi - \int_{A_{cell}} dA\, r\psi_m\, \underline{\Omega}_m \cdot \underline{\nabla} b_i + \int_{A_{cell}} dA\, b_i\, r\sigma\psi_m - \int_{A_{cell}} dA\, b_i\, \beta_m \psi_m \\ = \int_{A_{cell}} r\, b_i\, Q_m dA - \int_{A_{cell}} dA\, \gamma_m\, b_i\, \psi_{m-1/2}. \end{aligned} \quad (2.25)$$

If we use the divergence theorem we may rewrite this as

$$\int_{\partial s} ds\, b_i\, \underline{n}_s \cdot \underline{\Omega}_m r\psi_m - \int_{A_{cell}} dA\, r\psi_m\, \underline{\Omega}_m \cdot \underline{\nabla} b_i + \int_{A_{cell}} dA\, b_i\, r\sigma\psi_m - \int_{A_{cell}} dA\, b_i\, \beta_m \psi_m = \int_{A_{cell}} r\, b_i\, Q_m dA - \int_{A_{cell}} dA\, \gamma_m\, b_i\, \psi_{m-1/2}. \tag{2.26}$$

Here the term $\underline{n}_s$ denotes the outward unit normal to surface S and ∂S represents the surface of our element.

We will be considering polygonal elements of S-sides. This means we can write the surface integral as a sum of integrals over each face of our spatial elements. We also note that we may split up the contributions between the surface terms involving the unknowns and the surface terms known from the upstream condition, which we will explain shortly. If we do this and split the interface terms we have

$$\begin{aligned} &\sum_{\substack{s=1, \\ \underline{\Omega}_m \cdot \underline{n}_s > 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{\Omega}_m \cdot \underline{n}_s \right| r\psi - \int_{A_{cell}} dA\, r\psi\, \vec{\Omega}_m \cdot \vec{\nabla} b_i \\ &\qquad + \int_{A_{cell}} dA\, b_i\, r\sigma\psi_m - \int_{A_{cell}} dA\, b_i\, \beta_m \psi_m \\ &= \sum_{\substack{s=1, \\ \underline{\Omega}_m \cdot \underline{n}_s < 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{\Omega}_m \cdot \underline{n}_s \right| r\psi + \int_{A_{cell}} r\, b_i\, Q_m dA - \int_{A_{cell}} dA\, \gamma_m\, b_i\, \psi_{m-1/2}. \end{aligned} \tag{2.27}$$

Here L_s is used to indicate a particular face of a polygonal element and dL_s is a differential element of length on that face. The surface terms are split up according to the upstream condition, which is determined by the sign of the quantity $\underline{\Omega} \cdot \underline{n}_s$. If this quantity is negative then the current quadrature angle $\underline{\Omega}_m$ is inflow for the face in consideration and its value is known from a boundary condition on the element in

question. In this case this term is known and has been moved to the right-side of Eq. (2.27). If the quantity $\underline{\Omega} \cdot \underline{n}_s$ is positive then $\underline{\Omega}_m$ is outflow for the face in question and this term is a loss term and thus is placed on the left-side of Eq. (2.27).

The next step in our spatial discretization involves the specification of a trial space for the spatial dependence of the angular flux on each element. We choose a trial space that is linear in the variables r and z. We seek a linear basis expansion for the angular flux of the form

$$\psi_m(r,z) = b_0 \psi_m^0 + b_r \psi_m^r + b_z \psi_m^z. \tag{2.28}$$

Here the linear angular flux representation has been written in terms of the three unknowns ψ_m^0, ψ_m^r, and ψ_m^z using arbitrary basis functions b_0, b_r, and b_z. This expansion is known as the linear discontinuous (LD) expansion because no continuity between elements is enforced. The choice of basis functions is somewhat arbitrary but they must span the linear trial space. For the weighted-direction equations we choose to use

$$\begin{aligned} b_0 &= 1 \\ b_r(r) &= \frac{r - r_{ctrd}}{\tilde{r}} \\ b_z(z) &= \frac{z - z_{ctrd}}{\tilde{z}}. \end{aligned} \tag{2.29}$$

Here the variables r_{ctrd} and z_{ctrd} denote coordinates of the centroid of the finite element under consideration. The parameters $\tilde{r}$ and $\tilde{z}$ are characteristics sizes in the r and z

dimensions for the finite element under consideration. These characteristic sizes are used to scale the basis functions, which reduces issues with machine round-off error. We note that this choice of basis functions simplifies evaluation of the element-averaged angular fluxes since the mean value of a linear function over a domain is equal to its value at the centroid coordinates of that domain. In this manner the basis functions chosen above dictate that the mean value of the angular flux over any element for angle m is simply ψ_m^0.

If we insert our linear flux expansion from Eq. (2.28) into Eq. (2.27) we obtain an equation with three unknowns for an arbitrary weight function b_i. We note that both the spatial inflow fluxes and angular inflow fluxes are presumed known at this point. In order to solve this system we require three separate equations which we shall obtain from substituting various forms of the weighting function b_i into Eq. (2.27). By choosing weighting functions equal to the basis functions used in the trial space we obtain what is known as a Galerkin finite element method (GFEM). This defines a system of three equations in three unknowns in conjunction with inflow spatial fluxes and inflow angular edge fluxes whose solution provides a linear angular flux profile on a particular element.

The complete weighted-direction equations for our LDGFEM are given below.

$$\left\{\begin{array}{c} \sum\limits_{\substack{s=1,\\ \underline{\Omega}_m\cdot\underline{n}_s>0}}^{S} \int_{L_s} dL_s\, b_i \left|\underline{\Omega}_m\cdot\underline{n}_s\right| r\psi \;-\int_{A_{cell}} dA\, r\psi\, \bar{\Omega}_m\cdot\bar{\nabla} b_i \\ +\int_{A_{cell}} dA\, b_i\, r\sigma\psi_m - \int_{A_{cell}} dA\, b_i\, \beta_m\psi_m \\ = \sum\limits_{\substack{s=1,\\ \underline{\Omega}_m\cdot\underline{n}_s<0}}^{S} \int_{L_s} dL_s\, b_i \left|\underline{\Omega}_m\cdot\underline{n}_s\right| r\psi \;+\int_{A_{cell}} r\, b_i\, Q_m dA - \int_{A_{cell}} dA\, \gamma_m\, b_i\, \psi_{m-1/2} \end{array}\right\}$$

$$\psi_m\left(r,z\right) = b_0\psi_m^0 + b_r\psi_m^r + b_z\psi_m^z \tag{2.30}$$

$$b_i = \left\{b_0, b_r, b_z\right\} = \left\{1, \frac{r-r_{ctrd}}{\tilde{r}}, \frac{z-z_{ctrd}}{\tilde{\mathrm{z}}}\right\}$$

2.3.2 Spatial discretization for non-weighted directions

The LDGFEM equations in RZ geometry given by Eq. (2.27) require the starting direction fluxes appearing in Eq. (2.8). The starting direction fluxes obey the XY transport equation given in Eq. (2.13). In this section we discretize the starting direction flux equation using a LDGFEM.

We begin by following the same procedure outlined in section 2.3.1 but applied to the starting direction equation. If we multiply this equation by an arbitrary weight function b_i and integrate over the Cartesian element area we obtain

$$\int_{A_{cell}} dA\; b_i\left[\mu_m\frac{\partial}{\partial r}\psi_m + \xi_m\frac{\partial}{\partial z}\psi_m + \sigma\psi_m = Q_m\right]. \tag{2.31}$$

For simplicity we have denoted the starting direction fluxes for a particular direction with an m prefix rather than the integer half-multiple indices used in section 2.3.1. The divergence theorem may be applied to rewrite this as

$$\int_{\partial s} ds\, b_i\, \underline{n}_s \cdot \underline{\Omega}_m \psi - \int_{A_{cell}} dA \psi\, \underline{\Omega}_m \cdot \underline{\nabla} b_i + \int_{A_{cell}} dA\, b_i\, \sigma \psi_m = \int_{A_{cell}} b_i\, Q_m dA. \tag{2.32}$$

As before the surface integral may be split up into contributions from each polygonal face, with inflow and outflow terms moved to separate sides of the equation. When this is done we obtain the general FEM form of the starting direction flux equation.

$$\begin{aligned} &\sum_{\substack{s=1,\\ \underline{n}_s \cdot \underline{\Omega}_m > 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{n}_s \cdot \underline{\Omega}_m \right| \psi - \int_{A_{cell}} dA \psi\, \underline{\Omega}_m \cdot \underline{\nabla} b_i + \int_{A_{cell}} dA\, b_i\, \sigma \psi_m \\ &\qquad = \sum_{\substack{s=1,\\ \underline{n}_s \cdot \underline{\Omega}_m < 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{n}_s \cdot \underline{\Omega}_m \right| \psi + \int_{A_{cell}} b_i\, Q_m dA \end{aligned} \tag{2.33}$$

The final step in spatial discretization of the starting direction equations is expansion of the angular flux in a linear trial space and choice of weighting functions. We choose the same linear flux that was specified for the weighted-direction equation as provided by Eq. (2.28) and Eq. (2.29). The full system may be written as below.

$$\begin{aligned} &\left\{ \begin{aligned} &\sum_{\substack{s=1,\\ \underline{n}_s \cdot \underline{\Omega}_m > 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{n}_s \cdot \underline{\Omega}_m \right| \psi - \int_{A_{cell}} dA \psi\, \underline{\Omega}_m \cdot \underline{\nabla} b_i + \int_{A_{cell}} dA\, b_i\, \sigma \psi_m \\ &\qquad = \sum_{\substack{s=1,\\ \underline{n}_s \cdot \underline{\Omega}_m < 0}}^{S} \int_{L_s} dL_s\, b_i \left| \underline{n}_s \cdot \underline{\Omega}_m \right| \psi + \int_{A_{cell}} b_i\, Q_m dA \end{aligned} \right\} \\ &\qquad \psi_m(r,z) = b_0 \psi_m^0 + b_r \psi_m^r + b_z \psi_m^z \\ &\qquad b_i = \{b_0, b_r, b_z\} = \left\{ 1, \frac{r - r_{ctrd}}{\tilde{r}}, \frac{z - z_{ctrd}}{\tilde{z}} \right\} \end{aligned} \tag{2.34}$$

We note that in Cartesian geometry our choice of basis functions is not as convenient since the centroid coordinates are those of the curvilinear grid and not the Cartesian. This means the Cartesian cell-average fluxes are not simply ψ_m^0 as before. One could choose a different basis for the starting direction equations than that for the curvilinear grid in order to achieve the same simplicity. Though this makes cell-averaged flux evaluation simple in the Cartesian grid, the convenience is essentially washed away because the starting direction fluxes must be mapped to the curvilinear grid prior to starting the first weighted direction.

2.4 Angular quadrature

The discrete ordinates method for solving the transport equation depends on the ability to integrate the angular flux to obtain the scalar flux and thus compute the scattering source. The transport equation is solved for the angular flux at discrete angles and then a quadrature formula is used to compute the scalar flux. Thus the scalar flux is approximated with a quadrature formula of the form

$$\int_{4\pi} d\Omega \psi\left(r, E, \underline{\Omega}, \mathrm{t}\right) \approx \sum_{m=1}^{ndir} \psi_m\left(r, E, \mathrm{t}\right) w_m. \tag{2.35}$$

Here ψ_m represents the angular flux at the discrete ordinate m, while w_m is the weight assigned to quadrature point m. A total of $ndir$ discrete directions are included in the quadrature set.

For this work we make use of the standard Gauss-Chebyshev triangular quadrature sets described in [21]. Essentially the quadrature uses specific ordinates on the unit sphere and associated weights to perform the integration given by Eq. (2.35). A graphical depiction of such a quadrature set is shown in Figure 3.

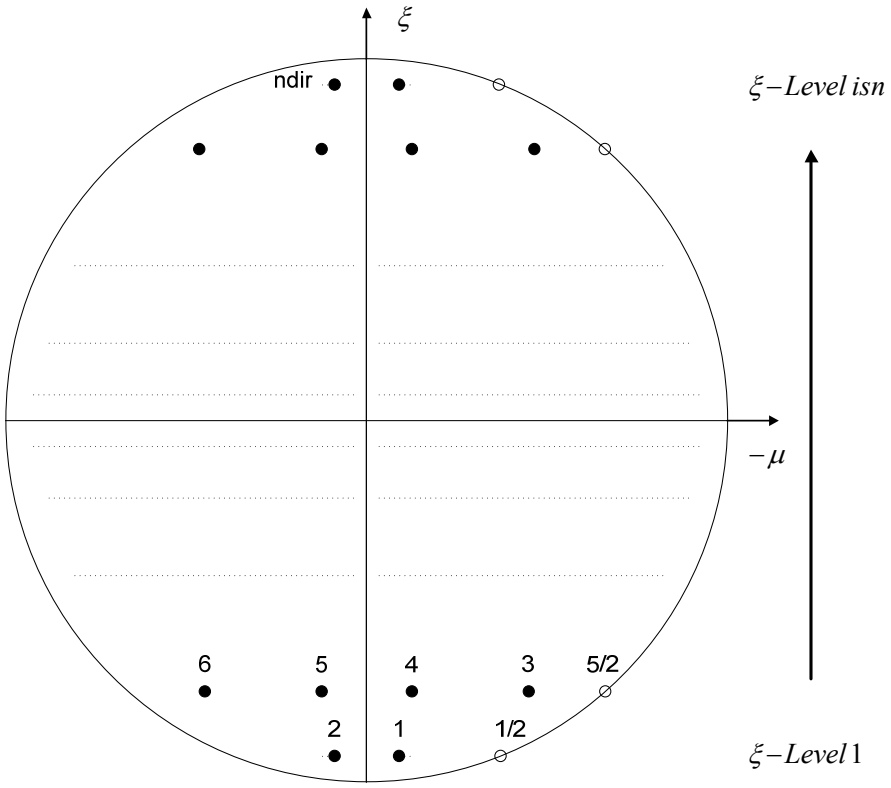

Figure 3. Depiction of a Gauss-Chebyshev triangular quadrature set on the unit half sphere

This quadrature set consists of a total of *isn* discrete ξ levels where *isn* is chosen as an even number such that the quadrature symmetrically accounts for both negative and positive ξ directional cosines. The distinct ξ level values are determined from the 1-D Gaussian quadrature of order *isn* for the interval $\xi \in [-1,1]$. Considering the negative valued ξ levels the values of μ are found from the formula

$$\mu_i = \sqrt{1-\xi_n^2}\cos(\omega_i), \tag{2.36}$$

where the azimuthal angle ω is as shown in Figure 1. The I values of ω_i for ξ level n are found from the formula for the Chebyshev nodes distributed on the interval $\omega \in [\pi, 2\pi]$. These may be computed from the formula

$$\omega_i = \pi + \frac{\pi}{2I}(2i-1). \tag{2.37}$$

The associated quadrature weights are determined by the weights of the 1-D Gaussian quadrature defining the ξ cosines. The quadrature weights for all values of μ on a particular ξ level are the same. The sum of all quadrature weights for all directions in the set is normalized to 4π.

This defines the ordinates and weights for $\xi < 0$. The top quarter sphere is symmetrical and may be obtained by reflecting those nodes about $\xi = 0$. The total directions *ndir*,

contained in a triangular Gauss-Chebyshev quadrature set having isn ξ levels may be computed from

$$ndir = \frac{isn\left(isn+2\right)}{2}. \tag{2.38}$$

For this work we have used various orders of the Gauss-Chebyshev quadrature set defined above. For reference the S_8 triangular quadrature set is given in Table 1 below. A graphical depiction of this quadrature set is shown in Figure 4.

Table 1. Gauss-Chebyshev S_8 angular quadrature

Ordinate	mu	xi	wt	Ordinate	mu	xi	wt
1	-0.1973	-0.9603	0.3180	21	-0.9641	0.1834	0.2849
2	0.1973	-0.9603	0.3180	22	-0.8174	0.1834	0.2849
3	-0.5584	-0.7967	0.3493	23	-0.5461	0.1834	0.2849
4	-0.2313	-0.7967	0.3493	24	-0.1918	0.1834	0.2849
5	0.2313	-0.7967	0.3493	25	0.1918	0.1834	0.2849
6	0.5584	-0.7967	0.3493	26	0.5461	0.1834	0.2849
7	-0.8218	-0.5255	0.3285	27	0.8174	0.1834	0.2849
8	-0.6016	-0.5255	0.3285	28	0.9641	0.1834	0.2849
9	-0.2202	-0.5255	0.3285	29	-0.8218	0.5255	0.3285
10	0.2202	-0.5255	0.3285	30	-0.6016	0.5255	0.3285
11	0.6016	-0.5255	0.3285	31	-0.2202	0.5255	0.3285
12	0.8218	-0.5255	0.3285	32	0.2202	0.5255	0.3285
13	-0.9641	-0.1834	0.2849	33	0.6016	0.5255	0.3285
14	-0.8174	-0.1834	0.2849	34	0.8218	0.5255	0.3285
15	-0.5461	-0.1834	0.2849	35	-0.5584	0.7967	0.3493
16	-0.1918	-0.1834	0.2849	36	-0.2313	0.7967	0.3493
17	0.1918	-0.1834	0.2849	37	0.2313	0.7967	0.3493
18	0.5461	-0.1834	0.2849	38	0.5584	0.7967	0.3493
19	0.8174	-0.1834	0.2849	39	-0.1973	0.9603	0.3180
20	0.9641	-0.1834	0.2849	40	0.1973	0.9603	0.3180

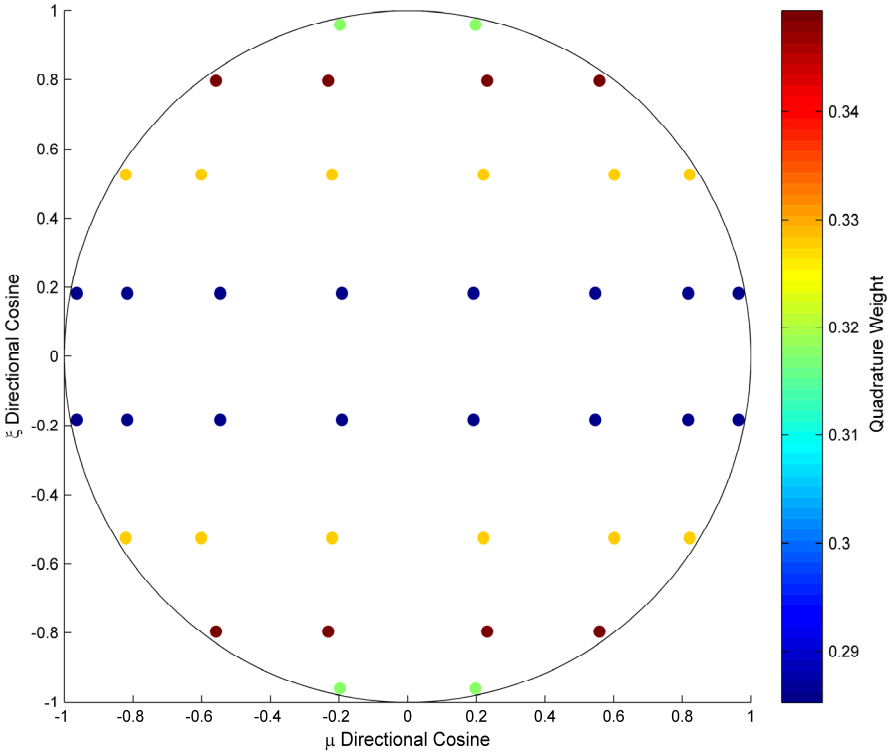

Figure 4. Graphical depiction of S_8 Gauss-Chebyshev angular quadrature

2.5 Spatial quadrature

The inner products appearing in our LDGFEM equations given in Eq. (2.30) and Eq. (2.34) require the integration of basis functions over our polygonal elements. In each case we will need to integrate polynomials over general elements of these forms. Here we describe quadrature sets that make this possible for arbitrary pentagons, quadrilaterals, and triangles.

2.5.1 Integrations on triangular elements

In order to perform integrations on triangular elements we shall make use of a coordinate transformation that facilitates easy numerical evaluation. We wish to map all triangular elements in our mesh to the right unit-isosceles triangle. Figure 5 depicts two triangular elements that will be used to describe the process. On the left is a triangular domain Ω^e which depicts an arbitrary triangular element within our RZ domain. Each of the vertices are labeled counter clockwise as shown. On the right side of Figure 5 is a right unit-isosceles triangle that encompasses the domain Ω^p. In this coordinate system the vertices are also labeled counter clockwise with the vertex associated with the right-angle labeled vertex 1. A local coordinate system is setup for this parent element in which the coordinates s and t vary between 0 and 1. We now wish to define a transformation that maps functions defined on the so-called element space Ω^e to the so-called parent space Ω^p which has a very simple form for which area integrals are readily computed.

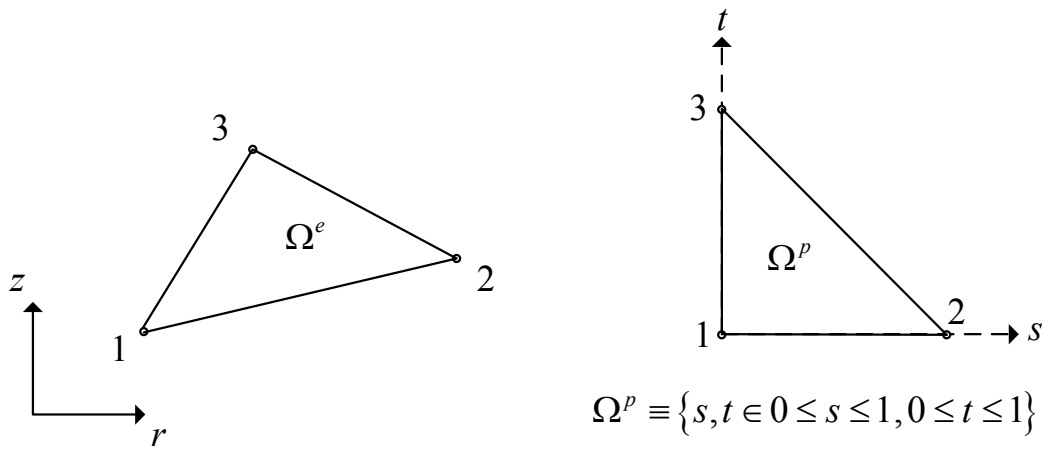

Figure 5. Coordinate transformation for triangular element integration

In order to map the element space to the parent space we require interpolation functions $\varphi_i(s,t)$ for each of the 3 parent nodes that have particular properties. The interpolation functions must be linearly independent and span the geometric space. In addition the interpolation functions must each uniquely take on a value of unity at one node and zero at the other nodes. In terms of the Kronecker delta this simply means

$$\begin{aligned} &\varphi_i(s_j,t_j) = \delta_{ij} \\ &i = 1 \to 3 \\ &j = 1 \to 3 \end{aligned} \quad , \tag{2.39}$$

where there is an interpolation function for each node. We choose to use the following well known set of interpolation functions which satisfy all the above requirements.

$$\begin{aligned} \varphi_1 &= 1 - s - t \\ \varphi_2 &= s \\ \varphi_3 &= t \end{aligned} \tag{2.40}$$

With these interpolation functions we may now define a 1:1 transformation that maps any point in the triangular element Ω^e, to a point on the parent unit right isosceles triangle Ω^p.This mapping is

$$\begin{aligned} r &= \sum_{i=1}^{3} r_i \varphi_i(s,t) \\ z &= \sum_{i=1}^{3} z_i \varphi_i(s,t). \end{aligned} \tag{2.41}$$

The relationship between a differential area in the element space Ω^e and the parent space Ω^p is given by the determinant of the Jacobian matrix of the transformation. The Jacobian matrix for a 2-D change of variables from $(r,z) \to (s,t)$ is

$$\underline{\underline{J}} = \begin{bmatrix} \dfrac{\partial r}{\partial s} & \dfrac{\partial r}{\partial t} \\ \dfrac{\partial z}{\partial s} & \dfrac{\partial z}{\partial t} \end{bmatrix}. \tag{2.42}$$

The determinant of this Jacobian matrix is:

$$J = \frac{\partial r}{\partial s}\frac{\partial z}{\partial t} - \frac{\partial r}{\partial t}\frac{\partial z}{\partial s} \tag{2.43}$$

The differential elements in each space are then related by:

$$dr\,dz = |J|\,ds\,dt \tag{2.44}$$

If we evaluate the Jacobian for this transformation we find that it is independent of s and t, and is related to the area of the element triangle A by:

$$|J| = 2A = \left|(r_1 - r_3)(z_2 - z_3) - (r_2 - r_3)(z_1 - z_3)\right| \tag{2.45}$$

We will perform our element integrations of the arbitrary function $F(r,z)$ in the parent space Ω^p by transforming both the integrand and the domain according to

$$\int_{\Omega^e} F(r,z)\,dA = \int_{\Omega^p} F\left(r(s,t),z(s,t)\right)|J|\,ds\,dt. \tag{2.46}$$

With the integrations transformed to the parent triangle we may use one of several available quadrature formulas. These quadrature formulas are normally written in terms of area coordinates. The geometric interpolation functions we have used are completely equivalent. The resulting integration formula is written as

$$\int_{\Omega^e} F(r,z)\,dr\,dz \approx \frac{1}{2}\sum_{i=1}^{N} F\left(r(s_i,t_i),z(s_i,t_i)\right) J\,w_i, \tag{2.47}$$

where the N point quadrature formula approximates the integral using ordinates (s_i,t_i) and associated weights w_i.

In determining which quadrature set to use for our triangular element integrations we note that we would like to exactly integrate all polynomials appearing in our discretized equations. The inner product $\langle r\,b_r\,b_r\rangle$ results in the highest order polynomial in s and t. We seek a quadrature that is exact for up to and including order three polynomials. We shall make use of a known four-point quadrature set that is exact for polynomials of up to order three. Table 2 contains the four point quadrature set used for this work [22].

Table 2. Triangular spatial quadrature set

φ_1	φ_2	φ_3	w
1/3	1/3	1/3	–27/96
3/5	1/5	1/5	25/96
1/5	3/5	1/5	25/96
1/5	1/5	3/5	25/96

2.5.2 Integration on quadrilateral elements

In order to perform integrations on quadrilateral elements we shall again make use of a coordinate transformation that facilitates easy numerical evaluation. We wish to map all quadrilateral elements in our mesh to the bi-unit square. Figure 6 depicts two quadrilateral elements that will be used to describe the process. On the left is a quadrilateral domain Ω^e which depicts an arbitrary quadrilateral element within our RZ domain. Each of the vertices are labeled counter clockwise as shown. On the right side of Figure 6 is a bi-unit square that encompasses the domain Ω^p. In this coordinate system the vertices are also labeled counter clockwise with the vertex associated with the lower left corner labeled vertex 1. A local coordinate system is setup for this parent element in which the coordinates s and t vary between -1 and 1. The transformation proceeds by finding interpolation functions mapping the element space Ω^e to the parent space Ω^p, which has a very simple form for which area integrals are readily computed.

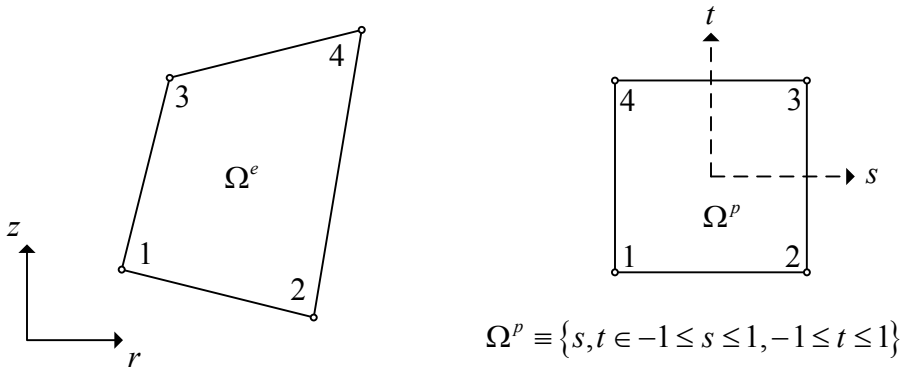

Figure 6. Coordinate system transformation for quadrilateral element integration

In order to map the element space to the parent space we require interpolation functions $\varphi_i(s,t)$ for each of the 4 parent nodes that have particular properties. As before, the interpolation functions must be linearly impendent and span the geometric space. In addition the interpolation function must each take on a value of unity at one node and zero at the other nodes. In terms of the Kronecker delta this simply means

$$\begin{aligned} &\varphi_i(s_j,t_j) = \delta_{ij} \\ &i = 1 \to 4 \\ &j = 1 \to 4 \end{aligned} \quad , \tag{2.48}$$

where there is an interpolation function for each node. We choose to use the following well known set of interpolation functions which satisfy all the above requirements.

$$\begin{aligned}\varphi_1 &= (1-s)(1-t)/4\\ \varphi_2 &= (1+s)(1-t)/4\\ \varphi_3 &= (1+s)(1+t)/4\\ \varphi_4 &= (1-s)(1+t)/4\end{aligned} \tag{2.49}$$

With these interpolation functions we may now define a 1:1 transformation that maps any point in the rectangular element Ω^e, to a point on the bi-unit square Ω^p.

The mapping is:

$$\begin{aligned}r &= \sum_{i=1}^{4} r_i \varphi_i(s,t)\\ z &= \sum_{i=1}^{4} z_i \varphi_i(s,t)\end{aligned} \tag{2.50}$$

The relationship between a differential area in the element space Ω^e and the parent space Ω^p is again given by the determinant of the Jacobian matrix of the transformation.

If we evaluate the Jacobian for this transformation using Eq. (2.43) we obtain Eq. (2.51).

$$\begin{aligned}J(s,t) = &\left[-\frac{r_1}{4}(1-t)+\frac{r_2}{4}(1-t)+\frac{r_3}{4}(1+t)-\frac{r_4}{4}(1+t)\right]\\ &*\left[-\frac{z_1}{4}(1-s)-\frac{z_2}{4}(1+s)+\frac{z_3}{4}(1+s)+\frac{r_4}{4}(1-s)\right]\\ &-\left[-\frac{r_1}{4}(1-s)-\frac{r_2}{4}(1+s)+\frac{r_3}{4}(1+s)+\frac{r_4}{4}(1-s)\right]\\ &*\left[-\frac{z_1}{4}(1-t)+\frac{z_2}{4}(1-t)+\frac{z_3}{4}(1+t)-\frac{z_4}{4}(1+t)\right]\end{aligned} \tag{2.51}$$

We may now perform our element integrations of the arbitrary function $F(r,z)$ in the parent space Ω^p using the formula given by Eq. (2.46). With the integrations transformed to the bi-unit square we may use one of several available quadrature formulas. The general quadrature integration formula is written as

$$\int_{\Omega^e} F(r,z)\,dr\,dz \approx \sum_{i=1}^{N} F\left(r(s_i,t_i), z(s_i,t_i)\right) J\, w_i, \tag{2.52}$$

where the N point quadrature formula approximates the integral using ordinates (s_i,t_i) and associated weights w_i.

In determining which quadrature set to use for our quadrilateral element integrations we again note that we would like to exactly integrate all polynomials appearing in our discretization. The inner product $\langle r\, b_r\, b_r \rangle$ results in the highest order polynomial in s and t. We seek a quadrature that is exact for up to and including order three polynomials. A four-point product Gaussian quadrature set achieves this. Table 3 contains the four point quadrature set used predominantly for this work [21].

Table 3. Product Gaussian quadrature set for spatial integration on quadrilaterals

s_i	t_i	w_i
$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{4}$
$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{4}$
$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{4}$
$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{4}$

2.5.3 Integration on pentagonal elements

Numerical integration on pentagons is performed via triangular decomposition. A graphical depiction of the decomposition procedure is shown in Figure 7. On the left hand side of the figure is shown an arbitrary pentagonal element encompassing the domain Ω^e. Vertices are numbered using the counter clockwise convention starting in the lower left hand corner. Three distinct triangles are formed by connecting the sets of vertices $(1,2,3)$, $(1,3,4)$, and $(1,4,5)$. The new sub elements are denoted Ω_1^e, Ω_2^e, and Ω_3^e. Numerical integration on the pentagonal elements is then computed using the triangular quadrature described in section 2.5.1 via accruing the contributions from each triangular sub-element. We note that careful choice of the triangular construction procedure can make some amount of difference in numerical conditioning. In particular

it is desirable to avoid elements that are near degenerate – with one side that is very small relative to the other sides.

To summarize we transform the pentagonal integration via triangular decomposition using

$$\int_{\Omega^e} F(r,z)\,dA = \int_{\Omega_1^e} F(r,z)\,dA + \int_{\Omega_2^e} F(r,z)\,dA + \int_{\Omega_3^e} F(r,z)\,dA. \tag{2.53}$$

Each sub element is a triangle and may be integrated using the quadrature formula given by Eq. (2.47) and the quadrature set given in Table 2.

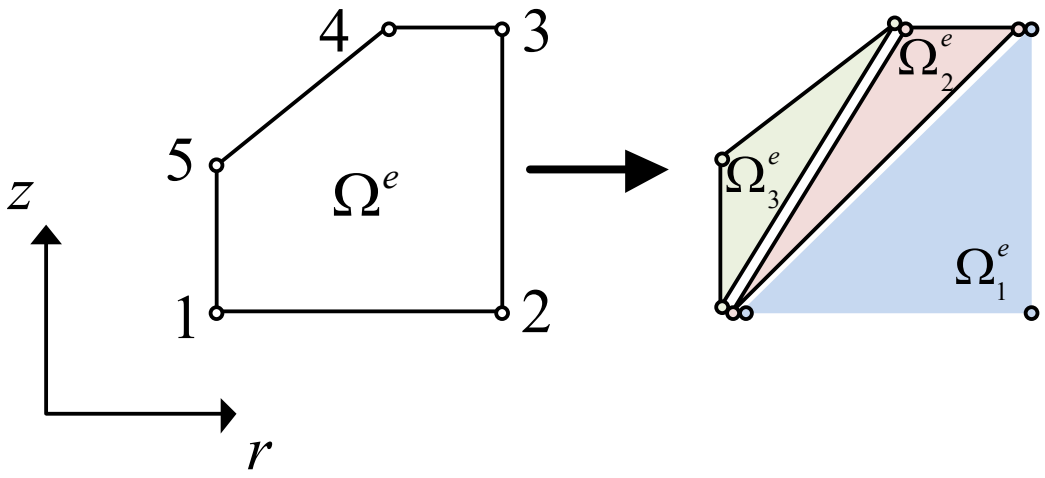

Figure 7. Decomposition of pentagonal elements into three triangular elements

2.5.4 Line integrals on polygonal faces

The advection terms present in the LDGFEM equations require computation of line integrals of basis functions over the faces of our mesh elements. We will perform these integrations with coordinate transformations and standard quadrature formula just as we did the area integrals. In order to describe this procedure we will make use of the sketch shown in Figure 8.

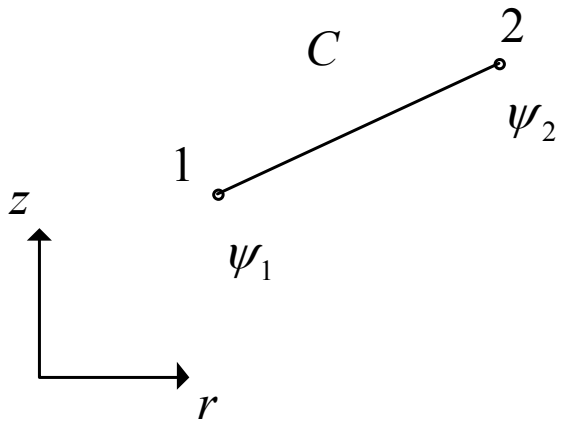

Figure 8. Depiction of line integral over cell face

Here we have labeled the two arbitrary vertices 1 and 2 using the lower left-hand corner convention. We have denoted the curve representing this face as C and we have also defined the value of the angular flux at each vertex as ψ_1 and ψ_2. We shall also denote the RZ coordinates at each vertex as (r_1, z_1) and (r_2, z_2).

We now consider the line integral of some arbitrary function $F(r,z)$ over the curve C

$$\oint_C ds\, F(r,z) \tag{2.54}$$

We define the slope of the line segment comprising C as:

$$m_{face} \equiv \frac{z_2 - z_1}{r_2 - r_1} \tag{2.55}$$

We may write a differential unit of arc length along C in two ways.

$$ds = \sqrt{dr^2 + dz^2} = \begin{cases} \sqrt{1 + m_{face}^2}\, dr \\ \sqrt{1 + m_{face}^{-2}}\, dz \end{cases} \tag{2.56}$$

We choose to integrate in the variable z for vertical faces and integrate in the variable r otherwise. Using this convention we may write the form of an arbitrary line integral on the polygonal face as follows.

$$\oint_C ds\, F(r,z) = \begin{cases} \int_{z_1}^{z_2} dz\, F(r,z)\, dz & , \quad r_1 = r_2 \\ \int_{r_1}^{r_2} dr\, F(r,z) \sqrt{1 + m_{face}^2}\, dr & , \quad otherwise \end{cases} \tag{2.57}$$

In practice it is best to choose the variable of integration in a way that maximizes its deviation across the line segment in order to reduce numerical round-off issues.

In order to perform the surface integrations we first perform a coordinate transformation. In order to setup the coordinate transformation we need geometrical interpolation functions that map a line segment. The nodes are the endpoints of the segment. As

before we require interpolation functions $\varphi_i(s,t)$ for each of the nodes span the 1-D space. We choose to use the following interpolation functions which satisfy our requirements.

$$\begin{aligned} \varphi_1 &= \frac{1-s}{2} \\ \varphi_2 &= \frac{1+s}{2} \\ s &\in [-1,1] \end{aligned} \tag{2.58}$$

The transformation maps an arbitrary face to a bi-unit line segment defined in one one-dimension. With these interpolation functions we may now define a 1:1 transformation that maps any polygonal face to a unit length line segment. The coordinate mapping is

$$\begin{aligned} r &= \sum_{i=1}^{2} r_i \varphi_i(s) \\ z &= \sum_{i=1}^{2} z_i \varphi_i(s). \end{aligned} \tag{2.59}$$

Using Eq. (2.43) we may write down a formula for the Jacobian of this transformation.

$$|J| = \begin{cases} z_2 - z_1 & , \quad r_1 = r_2 \\ \dfrac{\sqrt{(r_2 - r_1)^2 + (z_2 - z_1)^2}}{\sqrt{1 + m_{face}^2}} & , \quad otherwise \end{cases} \tag{2.60}$$

We note that the Jacobian is just the Euclidean length of the line segment analogous to how the Jacobian for our triangular transformation was twice the triangle area.

We now choose to approximate the line integral with a N point quadrature to obtain the formula

$$\oint_C ds\, F(r,z) \approx \begin{cases} \sum_{i=1}^{N} F\left(r(s_i), z(s_i)\right) |J| w_i & , \quad r_1 = r_2 \\ \sum_{i=1}^{N} F\left(r(s_i), z(s_i)\right) \sqrt{1+m_{face}^2}\, |J| w_i & , \quad otherwise. \end{cases} \tag{2.61}$$

The function $F(r,z)$ will be a product of basis functions when corresponding to outflow faces in a cell. For inflow faces the integrand is a product of basis functions and the linear representation of the angular flux on that face. Assuming that code data structures store the value of the angular flux at vertices the inflow terms require that we construct the linear representation of the angular flux along the face. This may be done one of two ways depending on which variable we have chosen to integrate against.

$$\psi_{face}(r,z) = \begin{cases} \psi_1 + \dfrac{\psi_2 - \psi_1}{r_2 - r_1}(r - r_1) \\ \psi_1 + \dfrac{\psi_2 - \psi_1}{z_2 - z_1}(z - z_1) \end{cases} \tag{2.62}$$

Here the linear representation of the angular flux ψ_{face} is constructed from the two known values at each vertex of the face.

For our purposes all integrals may be written as polynomials in the variable s. The most complex integrand encountered is a cubic. By using a 2-point Gaussian quadrature set

we may ensure that all integrals are evaluated exactly by our quadrature rule given in Eq. (2.61). The well-known two-point Gaussian quadrature formula is given in Table 4 [21].

Table 4. Two-point Gaussian quadrature on bi-unit line segment

s_i	w_i
$-\frac{1}{\sqrt{3}}$	$\frac{1}{2}$
$+\frac{1}{\sqrt{3}}$	$\frac{1}{2}$

2.6 Treatment of the total source term

To this point we have not discussed the right hand side of the transport equation given by (2.1). In this section we detail various considerations for treatment of the total source term for two particular scenarios of interest in this work. The simplest most common form of the source term is that taken for so called fixed source problems while another particularly relevant form is known as a k-eigenvalue computation. Below we discuss the relevant treatments in each scenario.

2.6.1 Fixed source computations

So called fixed source computations are those characterized by a total source that includes contributions from in-scatter and an inhomogeneous source. Fixed source computations are the name given to computations that seek the neutron distribution in a medium given a known neutron source. An example of such a problem might include the modeling of domain containing a spontaneous fission source located in specific location in the domain. In this scenario we may write the monoenergetic transport equation with isotropic scattering source term as

$$Q_{fixed-source}\left(\underline{r},\underline{\Omega}\right)=\frac{\sigma_s\left(\underline{r}\right)\phi\left(\underline{r}\right)}{4\pi}+\frac{S\left(\underline{r}\right)}{4\pi}. \tag{2.63}$$

Here the total scattering cross section is σ_s, the inhomogeneous source is S, and the scalar flux is still ϕ. The presence of the 4π parameter is used to uniformly distribute in angle the particle production. When solving the discretized form of the transport

equation the inhomogeneous source is specified by problem input and the scattering source is computed using the last iterate of the scalar flux associated with the source iteration procedure.

The discretized forms of the transport equation given by Eq. (2.30) and Eq. (2.34) require that we compute moments of the total source term on each finite element. In order to perform this computation we simply use the full linear representation of the source term and perform inner products on finite elements using the spatial quadrature schemes developed in section 2.5.

2.6.2 Eigenvalue computations

K-eigenvalue computations are the name given to a class of transport problems involving multiplying media with no fixed source which are the most prevalent application of the neutron transport equation. These problems model the neutron population in systems exhibiting some form of multiplication whereby the underlying materials have neutron generation cross sections whereby neutrons having collisions in the media may give rise to more neutrons. Such chain reactions form the basis of the field of nuclear engineering and are fundamental to the operation of nuclear reactors as well as national defense applications.

For eigenvalue computations we may write the monoenergetic transport equation with isotropic scattering source term as:

$$Q_{k-eigenvalue}\left(\underline{r},\underline{\Omega}\right)=\frac{1}{k}\frac{\nu\sigma_f\left(\underline{r}\right)\phi\left(\underline{r}\right)}{4\pi}+\frac{\sigma_s\left(\underline{r}\right)\phi\left(\underline{r}\right)}{4\pi}. \tag{2.64}$$

The parameter k is known as the k-eigenvalue or multiplication factor of the system and characterizes the system's ability to maintain a chain reaction. The terms ν and σ_f are the mean numbers of neutrons produced per fission and the macroscopic cross section for fission respectively. These terms are typically grouped and the product of each may be interpreted as the probability of neutron production from the fission reaction per unit path length traveled by a neutron. The term σ_s represents the macroscopic scattering cross section and ϕ represents the scalar flux as before.

If we examine the transport equation for an eigenvalue problem we note that we may write the analytic expression in operator form as

$$H\psi = S\psi + \frac{1}{k}F\psi, \tag{2.65}$$

where H represents the streaming plus absorption operator, S is the scattering operator, and F is the fission source operator. We recognize (2.65) as an eigenvalue problem and note that in general solutions will take the form of a set of eigenvalue and eigenvector pairs. For our problems it may be shown that only a single eigenvector is positive throughout the problem domain. This eigenvector is known as the fundamental mode

and represents the neutron distribution in a multiplying system. The eigenvalue is known as the multiplication factor of the system.

The traditional procedure for solving k-eigenvalue problems is known as power iteration and is the technique employed in this work. The basic iteration formula may be written in operator notation as

$$\psi^{(l+1)} = \frac{1}{k^{(l)}}\left(H - S\right)^{-1} F\psi^{(l)}, \tag{2.66}$$

where a standard choice for the eigenvalue iteration is

$$k^{(l+1)} = k^{(l)} \frac{\int_V dV\, F\psi^{(l+1)}}{\int_V dV\, F\psi^{(l)}}. \tag{2.67}$$

The integrals appearing in (2.67) are performed over the entire volume of the problem domain. In practice the angular flux is typically normalized prior to each iteration.

To understand how power iterations works it is instructive to note that the multiplication factor may be physically interpreted as the mean number of neutrons generated by a given neutron during its traversal from inception to ultimate disappearance (capture, leakage, etc.). Following one complete iteration of source iteration, successive iterates of the scalar flux may be used to generate an estimate of the multiplication factor via the relationship given by Eq. (2.67). As the power iteration procedure converges the spatial

distribution of the scalar flux attains that of the fundamental mode and then Eq. (2.67) yields the multiplication factor for the system.

We note that k-eigenvalue problems do not themselves specify the neutron population in a system but rather its distribution and its multiplication characteristics. The magnitude of the scalar flux is mathematically arbitrary in such systems. In particular the neutron population can only be at steady state when the multiplication factor of the system is unity. Even in this case the amplitude of the neutron population is still arbitrary. For this reason as well as numerical considerations the fluxes are normalized at the beginning of each iteration.

The implementation of k-eigenvalue problems proceeds by introducing another iteration loop on the outside of the scattering source iteration procedure. The discretized forms of the transport equation given by Eq. (2.30) and Eq. (2.34) still require that we compute moments of the total source term on each finite element. These inner products on finite elements are computed as before using the spatial quadrature schemes developed in section 2.5.

2.7 Boundary conditions

Both the weighted and non-weighted discretized LDGFEM equations require the specification of boundary conditions whereby the angular flux entering the problem

domain is provided. If we refer to the cylinder caps with a "top" and "bottom" designation, and the cylinder lateral face with the "face" designation, then we must have for a cylinder of extent $\{r,z\} \in \left[(0,R) \mathrm{x} (0,Z) \right]$ the following:

$$\begin{aligned} &\psi\left(r, z=0, \underline{\Omega}_m \cdot \underline{\mathrm{n}}_s < 0\right) = \psi_{bottom}\left(r, \underline{\Omega}_m\right) \quad , \\ &\psi\left(r, z=Z, \underline{\Omega}_m \cdot \underline{\mathrm{n}}_s < 0\right) = \psi_{top}\left(r, \underline{\Omega}_m\right) \quad , \\ &\psi\left(r=R, z, \underline{\Omega}_m \cdot \underline{\mathrm{n}}_s < 0\right) = \psi_{face}\left(z, \underline{\Omega}_m\right). \end{aligned} \tag{2.68}$$

Here we have stated that the angular flux for directions incoming to the problem domain must be prescribed by the problem statement as a boundary condition. In general the functional form of the angular flux along a boundary must be used to determine the moments of the angular flux incoming to a cell boundary. For this work boundary conditions are set by storing fluxes at each node of the cell face coinciding with the cylinder boundary. The actual fluxes stored are found by enforcing that the spatial moments of the boundary flux match the moments of the mapped linear representation. For linear boundary fluxes this simplifies to storing value of the boundary flux at each vertex of the boundary face.

Both the spatial and angular dependence of the angular flux must be specified. In theory it is possible to discretely specify the angular flux for all incoming directions in the angular quadrature. For this work we have restricted ourselves to typical scenarios involving either isotropic or normal incident partial currents on the problem boundaries.

For an isotropic angular flux the user specifies the incoming partial current and thus the angular flux is found via

$$\psi^{boundary}_{\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}<0} = \frac{J^{boundary}_{incident}}{\sum_{m;\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}<0} \left|\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}\right| w_m}. \tag{2.69}$$

Here $J^{boundary}_{incident}$ is the specified incoming partial current on an arbitrary boundary, $\underline{\Omega}_m$ is a particular ordinate in the angular quadrature, w_m is the weight associated with that quadrature angle, and $\underline{\mathrm{n}}_{boundary}$ denotes the outward unit normal associated with the boundary. Vacuum boundary conditions may be obtained by simply setting the incoming partial current to zero. For normal incidence we simply associate all angular fluxes with the quadrature angle that is closest to normally incident on the boundary face. Thus the boundary fluxes are obtained via

$$\psi^{boundary}_{\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}<0} = \begin{cases} \dfrac{J^{boundary}_{incident}}{\max\left(\left|\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}\right|\right) w_m} & , \quad m \,|\, \max\left(\left|\underline{\Omega}_m \cdot \underline{\mathrm{n}}_{boundary}\right|\right) \\ 0 & otherwise. \end{cases} \tag{2.70}$$

For RZ geometry is it necessary to specifically implement a cylinder centerline boundary condition. To obtain the appropriate expression for the cylinder boundary condition we note that due to the definition of the angular variables the only way a particle may stream into the cylinder centerline is if it has direction $\omega = \pi$. These are particles transported in the starting direction flux sweep. Thus the appropriate centerline boundary condition may be verbally stated as: the angular flux at the cylinder origin, for all

weighted directions, must be equal to the starting direction flux at the origin for that particular ξ-level. In equation form this says simply:

$$\psi_m\left(r=0,z\right)=\psi_{m\_sd}\left(r=0,z\right) \tag{2.71}$$

Here the linear representation of the angular flux for angle m, for an arbitrary cell face along the cylinder centerline is denoted ψ_m. The associated starting direction flux for angle m is denoted ψ_{m_sd}. We note that the full linear spatial dependence must be mapped on each cell face aligned with the cylinder origin. This is equivalent to mapping the weighted-direction fluxes at each cell face vertex. Thus this boundary condition provides a total of two equations for each polygonal element having a face coinciding with the cylinder origin. Since a total of three equations are needed in each cell the two equations denoted by Eq. (2.71) take the place of the r and z moment equations given by Eq. (2.30). We summarize by stating that for elements along the cylinder centerline the LDGFEM equation reduce to conservation and the matching of the linear angular flux profile along the cylinder centerline as specified by the starting direction flux.

2.8 Solution procedure

In this section we have developed many equations which may be collectively used to numerically solve the neutron transport equation on orthogonal grids in RZ geometry. We now discuss in some detail the procedure by which these equations are employed. We begin with a simple flow diagram for the process given in Figure 9.

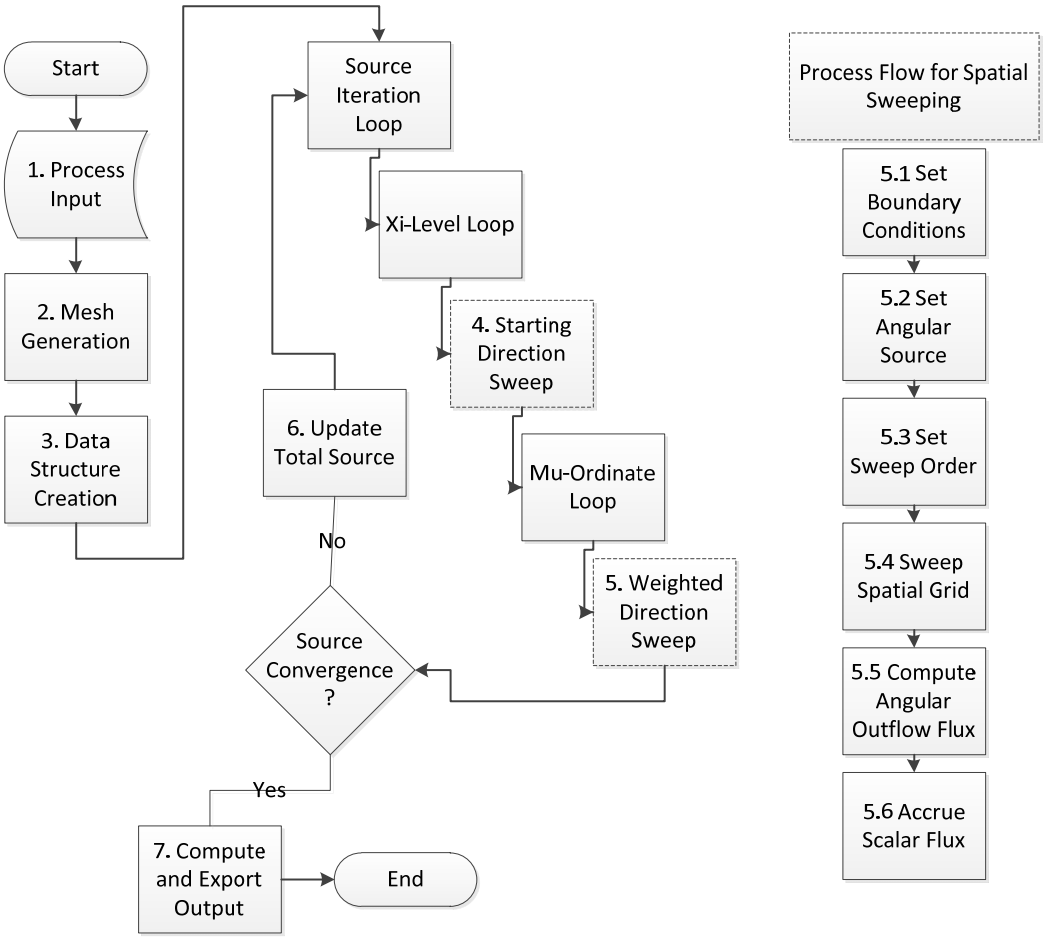

Figure 9. Flow diagram for numerical solution of RZ transport equation for fixed source problem

Referring to this figure we note that processes are given numbers. Using this numbering scheme as a reference we shall describe the solution procedure. We note that the diagram

shown is for fixed source computations. Problems involving k-eigenvalue computations are modified with the power iteration procedure on top of the scattering source iteration.

The solution procedure begins with problem input where the problem geometry, material properties, and boundary conditions are specified. Also specified are numerical discretization parameters such as mesh size, angular quadrature order, etc. Next the problem is encased in a space/angle mesh as indicated by step 2. The meshing procedure used is discussed in some detail in section 3.1 of this document. In step 3 of the flow diagram data structures are set up to perform the computation.

The actual computation begins with the outermost loop referred to as the source iteration loop. Since the angular flux appears on both sides of Eq. (2.1), an iterative procedure is used in which an iterate of the angular flux is used to compute the total source term, the discretized equations are solved to yield a new angular flux iterate, and this new angular flux is used to compute a new source term. This procedure continues until convergence is reached whereby solving the discretized equations results in a flux that is arbitrarily close to the last iterate. In practice the scalar flux itself is used to test convergence. The standard form of this iteration is a form of Richardson iteration and may be written for a fixed source problem as

$$\psi^{(l+1)} = L^{-1}P\psi^{(l)} + L^{-1}\,\mathrm{q}, \tag{2.72}$$

where $\psi^{(l+1)}$ denotes the new iterate of the angular flux to be solved for, $\psi^{(l)}$ is the old angular flux iterate used to compute the total source term, the terms L and P represent the discretized loss and production operators describing Eq. (2.30), and the term q represents the fixed source term. As the space-angle grid is swept a new iterate of the scalar flux is accrued using Eq. (2.35).

The processes located within the source iteration loop are typically referred to as the space-angle sweeps and are indicated by the xi-level loop, the mu loop, and the spatial sweeps indicated by steps 4 and 5 within Figure 9. The xi and mu loops represent the angular discretization and are arranged such that each xi-level is independent, while the ordinates represented by each value of mu on a particular xi-level are linked via the angular differencing procedure given by Eq. (2.8). Referring to Figure 3 the angular loops may be visualized as marching from bottom to top and from right to left in the quadrature set. Thus for a particular xi-level the solution procedure is as follows. First the starting direction flux is swept through the spatial grid. Next the weighted diamond in angle approximation given by equations (2.8) and (2.9) is used to compute the angular inflow fluxes associated with the first weighted direction present on the current xi-level. Then for all weighted directions on the current xi-level, the weighted-direction discretization is used to sweep the spatial grid. Subsequent spatial sweeps are performed for each distinct value of mu using the weighted diamond in angle approximation to provide angular edge inflow fluxes until all ordinates on a given xi-level have new

iterates for the angular flux. These new angular flux iterates give rise to the new iterate of the scalar flux, thus completing the source iteration procedure. At this point a test for convergence is performed upon the scalar fluxes by computing a norm of the difference of two successive iterates. If convergence has been achieved then the sweeping procedure ends and control proceeds to program output as indicated in step 7. If convergence has not been achieved then the new scalar flux is used to compute a new total source term and the source iteration procedure continues.

To this point we have not detailed the spatial sweeping procedure indicated by steps 4 and 5 and referred to above in the description of the angular sweep. For both the weighted and un-weighted directions the spatial sweeping procedure is essentially identical save that the set of equations applied is different. The right side of Figure 9 details the spatial sweeping procedure and will be used to describe the process. A spatial sweep begins in step 5.1 where the problem boundary conditions are used to establish the incoming angular flux along the problem boundaries. The spatial sweeping procedure may be thought of physically as tracking this known flux through the problem domain, across the underlying media, and out of the problem domain as an outgoing boundary flux. With the boundary conditions set the total angular source term is computed for each spatial element using the procedure discussed in section 2.6. For a particular angular ordinate being swept the orientation of the various cell faces dictate a particular order in which the spatial cells must be swept. In particular the angular fluxes incident upon a cell face must either come from previously determined outflow angular

fluxes on the adjoining cell, or from a boundary condition. For orthogonal grids this process is simple and may viewed as sweeping the cylinder from top to bottom for negative xi, from bottom to top for positive xi, from outside to inside for negative mu, and from inside to outside for positive mu. For example when sweeping a particular angular ordinate having negative mu and positive xi the sweep order proceeds first from the cylinder lateral surface to the cylinder centerline, and then from the bottom to top of the cylinder. In order to detail the spatial sweeping procedure itself we start with a picture of a typical spatial grid as shown in Figure 10 below.

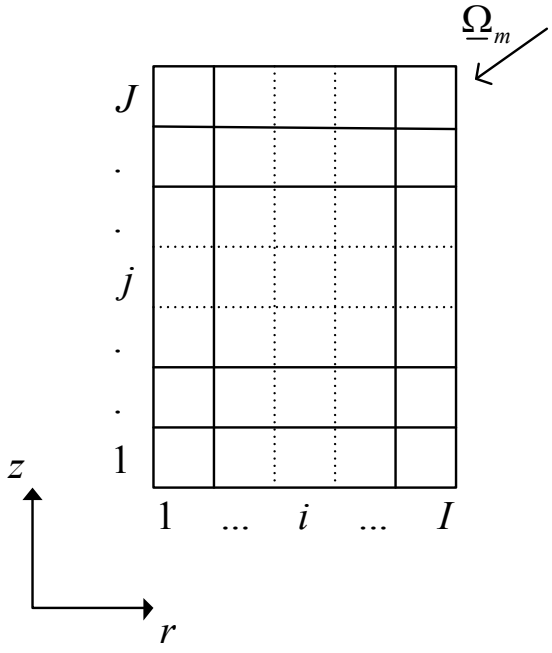

Figure 10. Graphical representation of a RZ spatial mesh

The RZ coordinate system is oriented as shown and the spatial grid is comprised of a total of I cell columns, each comprised of J rows. This gives a total of $I*J$ orthogonal cells. Cell numbering begins in the lower left corner and marches outward as shown. An arbitrary cell index is denoted by the location indices (i, j). We wish to consider the sweeping procedure for an arbitrary angle denoted as $\underline{\Omega}_m$ as shown in the figure. For the particular angle shown the direction cosines are $\mu_m < 0$ and $\xi_m < 0$. Thus the first cell to be swept lies at the location indices (I, J). The solution process involves solving for the three unknowns used to expand the linear angular flux in the cell. For weighted directions this is given by Eq. (2.30) and for un-weighted directions this is given by Eq. (2.34). The solution of these equations require: the incoming angular flux profiles on the inflow faces of the cell, the angular edge inflow fluxes for the current ordinate, and the angular source term.

With regard to the picture shown in Figure 10 the incoming angular fluxes for the cell with location indices (I, J) lie along the top and right side faces and are known from boundary conditions. The incoming angular edge fluxes are required only for the weighted direction equations and are in general determined from either the starting direction flux sweep on the current xi-level, or the previously swept weighted angular ordinate. The actual expansion of the angular source term is dependent on problem type

and is detailed in section 2.6 of this document. By filling in all the known parameters one obtains a 3x3 matrix equation whose solution provides the angular flux expansion in the element located at indices (I, J). With the angular flux profile in this cell determined it is now possible to compute the outgoing angular flux profile occurring along the left face of this cell. This flux profile is in turn the previously unknown incident flux profile for the right face of the cell lying at index $(I-1, J)$. Since the other incoming flux profile along the top of this cell is known from a boundary condition, the equations may now be solved to find the flux profile in this cell. In this manner the entire mesh row defined by the indices $(i, J), i = 1 \rightarrow I$ is swept with the resulting linear angular flux profile stored for each cell.

When the sweeping procedure reaches the cell lying at indices $(1, J)$ the underlying equations must be modified for weighted directions to account for the RZ centerline boundary condition. The underlying equations still represent a 3x3 matrix equation for the angular flux within the cell. The first equation is the conservation statement provided by the zero'th moment of Eq. (2.30). The second and third equations are found by specifying the angular flux profile along the cylinder centerline to match that of the starting direction flux along the centerline for the current xi-level. This detail was discussed in section 2.7 of this document. With this modification the underlying system may be solved to find the angular flux within the cell lying at indices $(1, J)$.

Returning to the cylinder lateral face we note that the cell lying at indices $(I, J-1)$ now has defined angular flux profiles on the incident faces occurring at the top and right side of the cell. Therefore the spatial sweeping procedure may be executed for this row of cells. This procedure continues until the entire spatial mesh has been swept and the angular flux profiles are known for the current angular ordinate within each spatial cell. When this process is completed the contributions to the new scalar flux iterate are accrued, angular outflow fluxes are computed, and the procedure continues to the next angular ordinate.

CHAPTER III

DESCRIPTION OF CUT-CELL METHODS

3.1 Mesh generation

In order to test our new cut-cell methods we will need the ability to create meshes exhibiting curvilinear material interfaces. In general, multi-dimensional mesh generation is a complex task. The purpose of this work is the development of the underlying spatial discretization and not mesh generation itself, thus we seek a simple solution for the creation of meshes that exhibit curvilinear material interfaces. A simple yet adequately sophisticated approach involves the modeling of 1-D spherical geometries within a 2-D RZ mesh. In this section we describe this meshing technique.

We start by considering a single homogeneous sphere with radius a_0. This sphere is encased symmetrically in a right circular cylinder. For simplicity we require that the centroid of the sphere coincide with the centroid of the cylinder. For a cylinder of radius R and height Z we require that the sphere be totally encased such that

$$R > a_0, \tag{3.1}$$

and

$$Z > 2a_0. \tag{3.2}$$

A sketch of such a sphere embedded in a cylinder is shown on the left side of Figure 11.

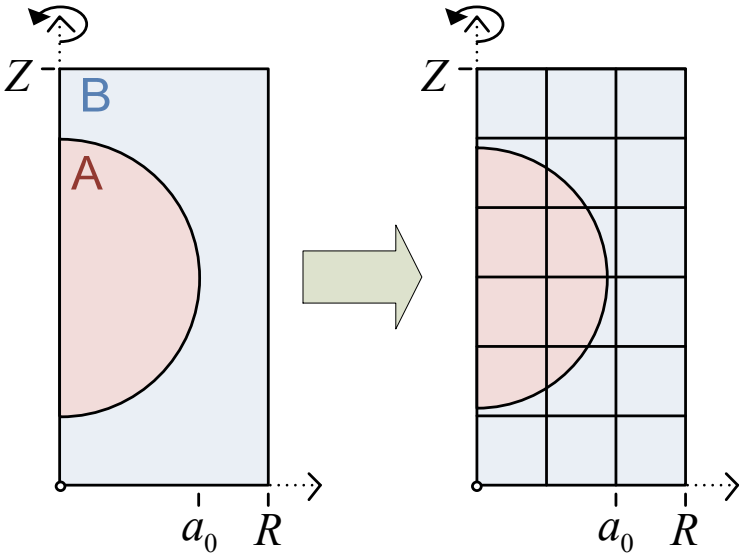

Figure 11. Sphere encased in RZ cylinder

We note that this procedure forms two distinct material regions with a curvilinear interface as desired. We will denote the material inside of the sphere as material A and the material outside of the sphere as material B as indicated in the figure. If material B is taken as a void and vacuum boundary conditions are applied to the cylinder, then we are modeling a 1-D bare sphere. If instead reflective boundary conditions are employed on the cylinder lateral face and caps, then we have an infinite lattice of spheres comprised of material A arranged in a matrix of material B.

Geometries such as this are those for which we create meshes in this work. We shall also consider geometries constructed from a series of $N_{spheres}$ spherical shells with radii

$a_0, a_1, \ldots a_{N_{spheres}}$, where $a_0 < a_1, \ldots < a_{N_{spheres}}$. The inner sphere has radius a_0 while each subsequent ring creates a spherical shell around the inner sphere. Thus a mesh created from an embedded sphere and a single shell with reflective boundary conditions might model an infinite lattice of fuel spheres each having a metal coating.

Our meshing procedure proceeds by encasing the RZ cylinder in a rectangular base mesh with no attention paid to the presence of material interfaces. The right side of Figure 11 depicts this meshing for a uniform 3x6 mesh. This orthogonal mesh is then given substructure by reconstructing the material interfaces with a line segment in every cell that contains multiple materials. Such cells will heretofore be referred to as mixed-cells while the underlying non-orthogonal sub-components shall be referred to as cut-cells. An illustration depicting this procedure is shown in Figure 12.

A linear representation of the material interface provides two degrees of freedom. The line segment is chosen to coincide with the true interface position on each of the mixed-cells. The resulting linear reconstructed interface is continuous throughout the entire domain. In particular this choice of fit imposes continuity of the line segments across adjacent mixed-cells sharing an interface. One could choose the segment to be a least squares approximation of the arc in each mixed-cell, but this would result in the reconstructed interface being discontinuous throughout the mesh. This discontinuity creates complications since the spatial dependence of the flux must be remapped to

convert the outflows of one cell to the inflows of its adjacent neighbor when the fluxes are not uniquely defined. By choosing the line segment to interpolate the true interface along the boundaries of the mixed-cell, this situation is avoided.

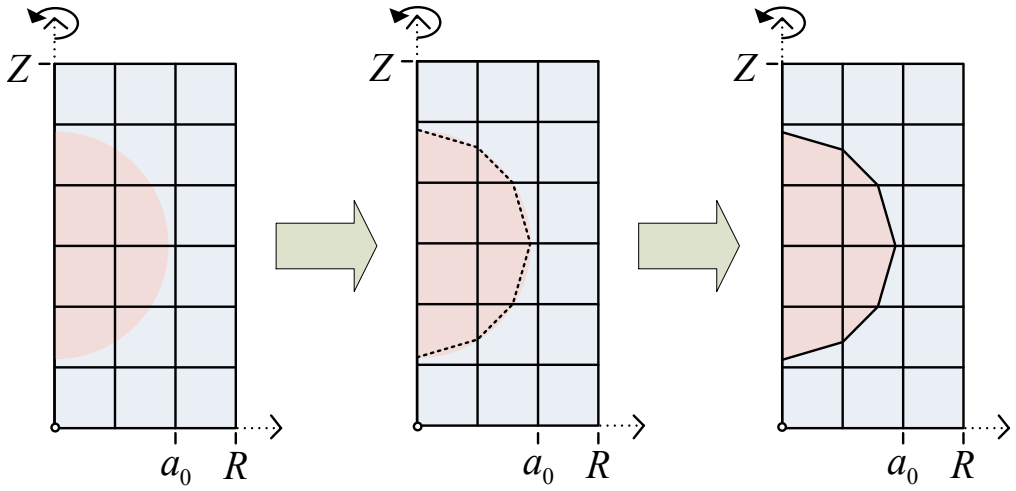

Figure 12. Linear interface meshing procedure

Referring to Figure 12 we note that our linear reconstruction of the curvilinear interface leaves a region in each mixed-cell where the linear reconstruction does not match the original curvilinear interface. This meshing error changes the mass of each material present in a mixed-cell. Although this effect is ignored in the graphic shown in Figure 12, we shall establish mechanisms for accounting for this issue in section 3.3.

For this work we restrict ourselves to base meshes that result in at most a single material interface in any mixed-cell. If the base orthogonal grid is insufficiently fine to achieve this result, it is globally refined to satisfy the criterion. Such a restriction is not absolutely necessary but it suffices to demonstrate the efficacy of our approach. In practice limiting the breadth of substructure has certain advantages in terms of implementation, resource efficiency, and, most importantly, amenability to parallelization.

The resulting mesh consists of an orthogonal base mesh comprised of a mixture of homogeneous rectangles and heterogeneous mixed-cells containing two non-orthogonal polygonal elements. All mixed-cells are comprised of a pair of juxtaposed polygons that share a face at the reconstructed interface. These polygonal sub-elements of the mixed-cell are referred to as cut-cells. For the two-material case all cut-cells are triangles, quadrilaterals, or pentagons.

It is theoretically possible for a material interface to exactly coincide with the corner of a mixed-cell. Such an occurrence introduces additional complexity in the implementation but does not change the underlying discretization in any fundamental way. For this work we have restricted ourselves to meshes that do not exhibit such an effect. During mesh generation, if such an occurrence is encountered, the base orthogonal mesh is tweaked, and mesh generation restarts using this new base mesh. With the restrictions denoted

previously there are a total of ten distinct mixed-cell types arising from the meshing procedure described above. A summary of these cut-cell types may be found in Appendix A.

3.2 Numerical cut-cell discretizations

In order to motivate the discussion of the various numerical methods it is instructive to begin with a sketch of a mixed-cell. A representative sketch of a mixed-cell and associated cut-cells is shown in Figure 13. Using a lower left-hand numbering scheme, the volumes and an arbitrary cross section have been labeled for cut-cell 1 and cut-cell 2.

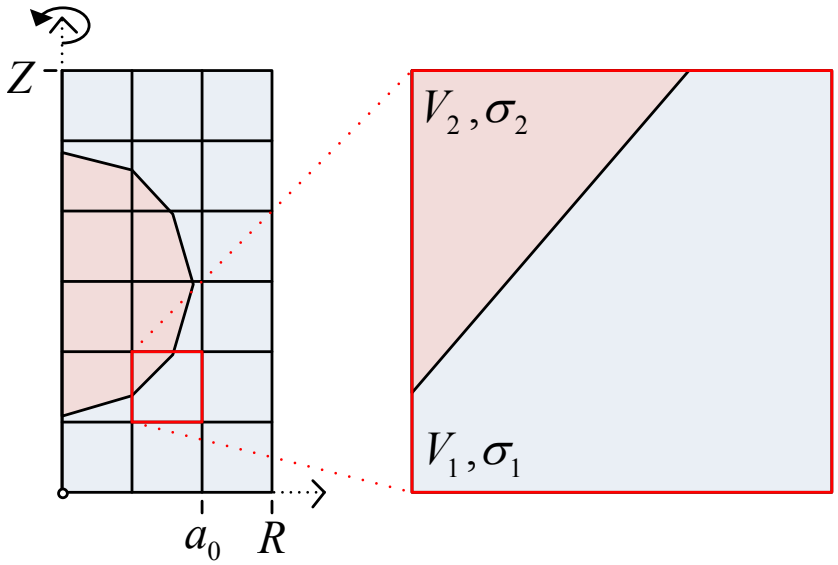

Figure 13. Sketch of representative mixed-cell

Using this sketch, a total of three different numerical methods shall be introduced in this section.

3.2.1 Homogenization

The standard treatment applied in the presence of material interfaces within a mixed-cell is the well-known technique of homogenization. Homogenization is so named because it approximates a multi-material region by replacing it with a homogenous region having material properties that are some average of those of the original constituents. More specifically the simplest homogenization scheme volume-weights the cross-sections of each material to produce an average:

$$\sigma_{\text{hom}} = \frac{\sum_i \sigma_i V_i}{\sum_i V_i} \tag{3.3}$$

Here σ_{hom} is the resulting homogenized microscopic cross section for some arbitrary reaction, σ_i is the cross section for material i, and V_i represents the volume occupied by material i. With respect to Figure 13 a homogenization treatment of the interface replaces the mixed-cell with a singular rectangular cell with the average cross section given by:

$$\sigma_{\text{hom}} = \frac{\sigma_1 V_1 + \sigma_2 V_2}{V_1 + V_2} \tag{3.4}$$

In this manner all mixed-cells in the problem domain are converted to homogenous cells. Homogenization combined with successive mesh refinement is almost universally used

to treat multi-material cells resulting from the imposition of a rectangular grid on a problem domain containing interfaces that do not coincide with the mesh boundaries.

In order to describe these steps and the associated underlying sweeping procedure in the context of our cut-cell meshes, we shall use the sketch shown in Figure 14. On the far left of this figure we have our 3x6 rectangular meshing resulting in mixed-cells. Shown is a particular angle in the quadrature set that is to be swept through the spatial grid. In the middle is shown the mixed-cell lying at rectangular grid cell indices of (2,2) in our mesh. This mixed-cell is comprised of a juxtaposition of a triangular and pentagonal element. Our numbering scheme has been used to label the volume elements and associated material properties with a "1" and "2" subscript as shown. We note that the cross-section shown is arbitrary and represents any material property including a fixed source term or fission cross-section. On the far right of the figure is shown the single rectangular element created from the homogenization procedure described above and by Eq. (3.4). A single volume element V has been formed from the union of the triangular and pentagonal elements. Homogenized cross sections are computed via Eq. (3.4) for each material property. The resulting homogeneous rectangular element has had each of its four faces labeled f_1, f_2, f_3, f_4 in a counter clockwise, negative-to-positive directional cosine convention.

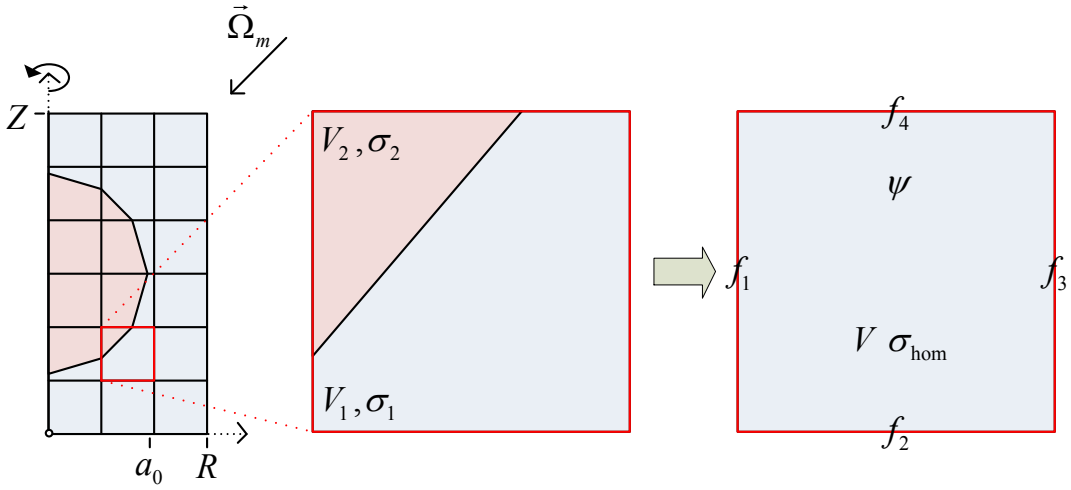

Figure 14. Homogenization sweeping procedure

The sweeping procedure for the homogenization method degenerates to the same procedure used on a regular orthogonal mesh after the homogenization process. In particular, considering the incoming angle $\vec{\Omega}_m$ shown in the figure element, faces f_3 and f_4 are inflow faces determined from outflows computed from mesh cells lying at indices (3,2) and (2,3), respectively. Note that these upwind faces provide a single linear angular flux profile on each of these faces since the homogenization procedure is applied to those cells as well. For the quadrature angle in consideration, the faces f_1 and f_2 are outflow faces. Thus the solution to the angular flux within the cell may now be computed using the inflow flux profiles and the DGFEM equations given by Eq. (2.30).

Although simple and easy to implement, homogenization is locally first-order accurate. This means that any higher order spatial discretization scheme for the transport equation will be degraded by the homogenization process. The effective order of convergence exhibited for such a scheme will vary depending upon the problem type, the quantity of interest, and primarily on the number and importance of the homogenized cells. For this work we have implemented a homogenization treatment to be used as the baseline comparison for the new cut-cell method.

3.2.2 Continuous flux cut-cell method (CFCM)

A very intuitive improvement on the homogenization technique may be theorized by considering for a moment the collision integral present in the RZ LDGFEM equations:

$$\int_{A_{cell}} dA b_i \, r \sigma \psi_m \tag{3.5}$$

If one assumes that the exact geometry is known at the time of meshing, the piecewise-constant spatial dependence of the cross-section appearing in Eq. (3.5) may be preserved in the fundamental discretization while retaining a single linear representation of the angular flux over the mixed-cell. In particular, with respect to Figure 13 the collision integral is written as:

$$\int_{V_{cell}} dA b_i \, r \sigma \psi_m = \int_{V_1} dV_1 \, b_i \, r \sigma_1 \psi_m + \int_{V_2} dV_2 \, b_i \, r \sigma_2 \psi_m \tag{3.6}$$

If one treats the total cross-section, scattering cross-section, fission cross-section, and inhomogeneous source terms in this manner, the resulting LDGFEM equations give a

scheme we shall refer to as the continuous flux cut-cell method (CFCM). We denote this method the "continuous flux" cut-cell method because although the discrete material regions in the mixed-cell are retained, the finite element trial space for the angular flux is a single linear representation across the mixed-cell. That is the angular flux is forced to be continuous across the material interface. This is physically incorrect as the true flux remains continuous but experiences a discontinuity in the derivative.

In order to describe these steps and the associated underlying sweeping procedure we shall again make use of a sketch, which is shown in Figure 15. The left and middle graphics in this figure show a base mesh, resulting mixed-cells, and a blown-up view of a particular mixed-cell identical to that described in Section 3.2.1. The graphic on the far right in Figure 15 depicts the mixed-cell with terminology appropriate for the CFCM. In particular this mixed-cell retains distinct material properties for the triangular and pentagonal cut-cells. Using the face numbering scheme described in Section 3.2.1, the faces of the mixed-cell have been numbered. We note that despite the distinct cut-cell structure to this mixed-cell the CFCM method need only consider the four faces as numbered in the figure and described below. The single linear representation of the angular flux within the mixed-cell has been denoted as ψ.

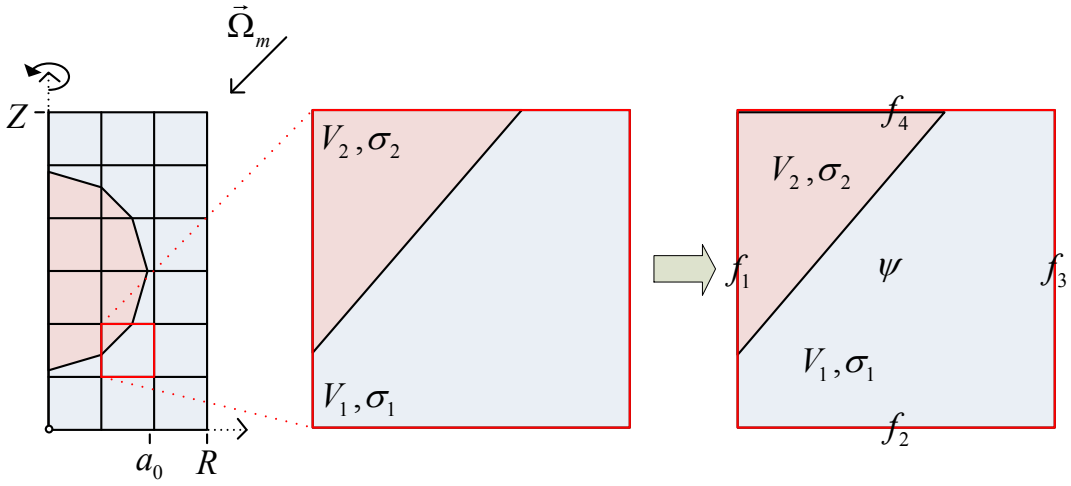

Figure 15. CFCM sweeping procedure

The sweeping procedure for the CFCM retains the characteristics of sweeping on a homogenous orthogonal grid with slightly increased complexity when a cut-cell is encountered. For the incoming angle $\vec{\Omega}_m$ shown in the figure, faces f_3 and f_4 are inflow faces determined from outflows computed from previously swept mesh cells lying at indices (3,2) and (2,3), respectively. Note that these upwind faces provide a single linear angular flux profile on each of these faces since the CFCM procedure generates a single linear angular flux profile in each mixed-cell. For the quadrature angle in consideration, the faces f_1 and f_2 are outflow faces. Thus the cell-to-cell interfacing for the CFCM remains identical to homogenization, which is itself identical to that used on a homogenous orthogonal mesh. The difference between the CFCM and homogenization lies in how the within-cell collision and source terms are computed

within the DGFEM equations. The CFCM method replaces all area integrals appearing in the DGFEM equations with piecewise-defined integrals that operate on each cut-cell individually. These integrals are all of the form shown in Eq. (3.6). With this modification, the solution to the angular flux within the cell may now be computed using the inflow flux profiles and the DGFEM equations.

As mentioned in Section 1.4 we have previously tested the analog of the CFCM technique for the diamond difference spatial discretization. The diamond difference, or Crank Nicolson, spatial discretization is equivalent to a continuous finite element method (CFEM) with a linear trial space. The results of that work may be summarized by stating that for multi-dimensional k-eigenvalue problems the CFCM method used in conjunction with diamond difference produces better accuracy and global order of convergence than homogenization. This occurs despite still yielding degraded order of convergence due to the presence of mixed-cells. We were able to show for that implementation that the CFCM method was equivalent to a particular homogenization in which the homogenized cross-sections are obtained by flux-volume weighting. This fact makes the method advantageous because it is easy to implement in existing codes, has relatively low additional computational cost, and gives improved accuracy and convergence compared to standard homogenization.

In this work we have implemented the linear-discontinuous analog of the CFCM described above. This has been done in order to provide a comparison to the new cut-cell method introduced in Section 3.2.3.

3.2.3 Discontinuous flux cut-cell method (DFCM)

The obvious flaw of the CFCM is due to the inability of a single linear function on the mixed-cell to adequately represent the discontinuous nature of the angular flux derivative at a material interface. Our new cut-cell method accounts for this by introducing distinct linear flux profiles within each cut-cell. Near the region of a material interface the two distinct fluxes allow for a more adequate representation of the spatial kink in the true solution. In addition, the jump discontinuity in the LD solution provides another degree of freedom to cope with the spatial kink in the underlying solution. We shall refer to this new cut-cell method as the discontinuous flux cut-cell method (DFCM). We have used the name DFCM to refer to the fact that the numerical angular flux across the material interface in a mixed-cell is itself discontinuous. We note that as the DFCM converges the angular flux becomes continuous across the interface but its derivative does not. Thus the DFCM maintains the correct physical behavior at the interface while the CFCM does not.

In order to describe these steps and the associated underlying sweeping procedure we shall again make use of a sketch, which is shown in Figure 16. The left and middle

graphics in this figure show a base mesh, resulting mixed-cells, and a blown-up view of a particular mixed-cell identical to that described in Section 3.2.1. The graphic on the far right in Figure 16 depicts the mixed-cell with terminology appropriate for the DFCM. In particular this mixed-cell retains distinct material properties and linear flux profiles for both the triangular and pentagonal cut-cells. Using the face numbering scheme described in Section 3.2.1, the faces of each cut-cell have been numbered. For simplicity the face describing the material interface has been dual-numbered as $f_{5/7}$. The distinct linear representations of the angular flux within each cut-cell have been denoted as ψ_1 and ψ_2.

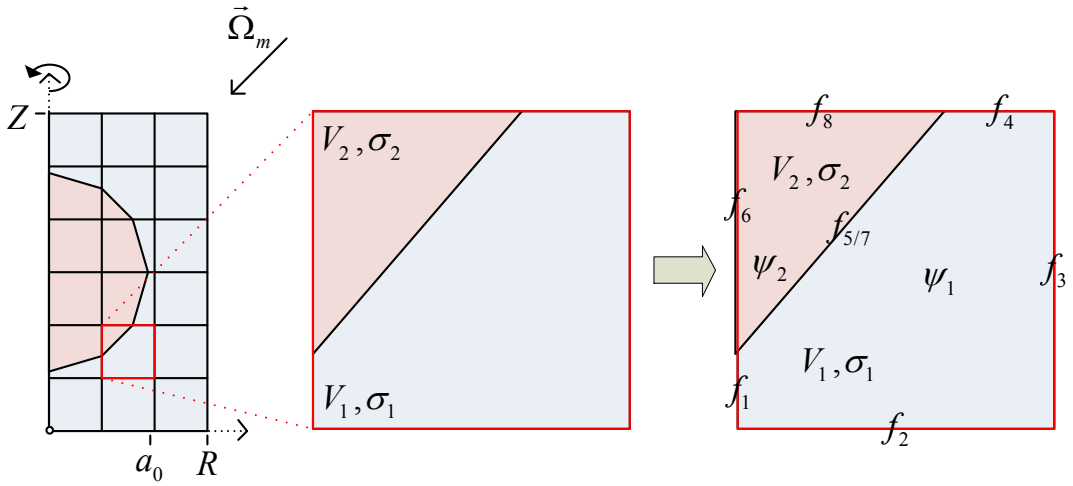

Figure 16. DFCM sweeping procedure

The sweeping procedure for the DFCM retains some of the characteristics of sweeping on a homogenous orthogonal grid but introduces cell-to-cell interaction complexities not encountered for homogenization or the CFCM. Referring to the incoming angle $\vec{\Omega}_m$ shown in Figure 16, we note that faces f_3, f_4, and f_8 are inflow faces. The face denoted $f_{5/7}$ represents the material interface and will be an outflow face for cut-cell 1 and an inflow face for cut-cell 2. Referring to the top face of the mixed-cell comprised of the faces f_4 and f_8 we note that the upwind cell located at mesh indices (2,3) will provide two distinct linear flux profiles – one for each face. The face f_3 will have its inflow flux profile provided by the upwind cell located at mesh indices (3,2). The faces denoted f_1 and f_6 will ultimately provide the two distinct inflow flux profiles required by the mesh cell lying at indices (1,2). Likewise, the outflow face denoted f_2 provides a single inflow flux profile to the mesh cell lying at indices (2,1).

In general the mixed-cell sweeping procedure for the DFCM relies on a unique flux profile on each inflow face for each cut-cell element. For the meshes defined in Section 3.1, this is ensured by continuity of the material interface throughout the mesh domain in conjunction with specified boundary conditions on the domain boundary. The procedure for determining sweep order on the cut-cells comprising the mixed-cell is the next step in the DFCM process. In general the sweep order is set by iteratively determining which cut-cells have defined flux profiles on each inflow face for the current quadrature angle.

This is done by computing the dot product of the outward face normal and the unit vector describing the direction of the current quadrature angle. For the meshes described in Section 3.1 there is at most one non-orthogonal face present in a mixed-cell and there are always two cut-cells in a mixed-cell; thus this process is quite simple. The quantity $\underline{\Omega}_m \bullet \underline{n}_{f_{int}}$, where $\underline{n}_{f_{int}}$ is the outward unit-normal associated with the shared material interface is evaluated for each of the two cut-cells, and the cut-cell for which this quantity is positive is swept first. The other cut-cell is swept afterward since the required inflow flux profile will then be known.

With the mixed-cell sweeping order determined as enumerated above, the angular flux profile in one cut-cell is computed using the LDGFEM equations given by Eq. (2.30) and the appropriate inflow flux profiles. Subsequently the outgoing flux profiles are stored, and the flux profile in the remaining cut-cell is computed using Eq. (2.30) and the newly available interface flux profile. The additional outgoing flux profiles in the mixed-cell are stored, and the sweeping procedure continues to the next mesh cell.

The DFCM accounts for both the material discontinuity as well as the known kink in the underlying flux within a mixed-cell. These are significant advantages relative to both homogenization and the CFCM and certainly provide the opportunity for increased computational efficiency and order of convergence. The cost of these improvements relative to the traditional methods involves increased memory usage and some increase

in the sweeping complexity. For all mixed-cells the memory storage requirements are approximately twice that of homogenization and about 50% more than the CFCM method.

Arguably the most significant advantage of the DFCM is that the basic sweeping procedure on orthogonal grids remains unchanged. The cut-cell treatment may be considered as a sub-sweeping step that occurs for a portion of the cells in the grid. By limiting the degree of substructure allowed, the DFCM is readily applicable to the block-orthogonal grids that exhibit significant parallelization advantages relative to general polyhedral grids.

Lastly, we make a note concerning the use of adaptive mesh refinement (AMR) relative to the use of DFCM on meshes exhibiting curvilinear material interfaces. AMR techniques typically implicitly provide increased mesh resolution in the vicinity of boundary layers such as those occurring at material interfaces. While this approach is quite powerful, we speculate that the DFCM technique used in combination with AMR will yield the most computationally efficient solutions to this class of problems.

3.2.4 Summary

In this section a total of three distinct spatial discretizations were introduced for solving the LDGFEM equations on cut-cell meshes. The homogenization method replaces the

mixed-cell with a single rectangular element with mean material properties obtained from volume-weighting the underlying constituents. Homogenization is the historical standard applied in essentially all codes today. The CFCM employs a single linear trial space for the angular flux over the entire mixed-cell but allows distinct material properties in each cut-cell. We have previously shown that a similar implementation for the Crank-Nicholson spatial discretization is more accurate than homogenization but is still first-order convergent. Our new cut-cell method referred to as the DFCM employs separate discontinuous linear trial spaces for the angular flux within each cut-cell comprising a mixed-cell as well as distinct material properties in each cut-cell. This new method requires sub-sweeping on mixed-cells but accounts for the known angular flux derivative discontinuity occurring at material interfaces.

3.3 Interface treatments

Figure 17 shows an exaggerated blow-up of a mixed-cell resulting from the mesh generation procedure described in Section 3.1. On the left side of the figure is our example 3x2 orthogonal mesh. On the right side of the figure is the mixed-cell located at mesh indices (2,2). Also shown is the original curvilinear arc passing through the mixed-cell, as well as the linear reconstruction. The sketch has been drawn in such a way to exaggerate the mesh error arising from representing the arc with a line segment. The resulting cut-cells and their respective volumes have been labeled in the figure. We shall

define δV as the volume lying between the interface arc and its linear representation. We shall refer to this volume as the mixed-cell interface residual.

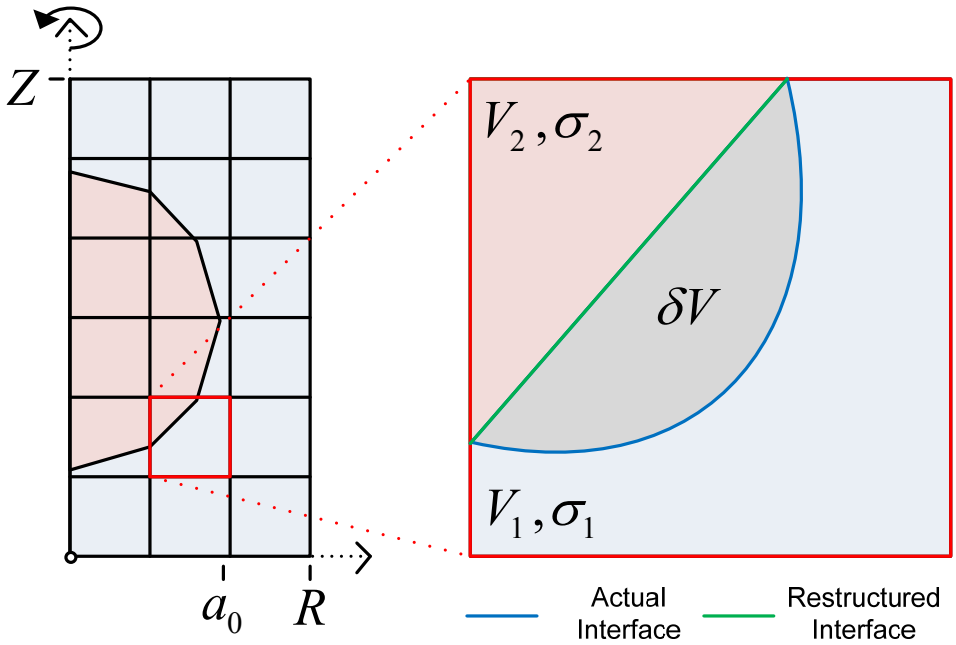

Figure 17. Mesh interface residual

The inability of the linear cut-cell mesh to exactly conform to curvilinear boundaries results in the presence of cut-cells whose underlying polygonal elements are not truly homogenous. More specifically, the cut-cells that are formed do not exactly replicate the distribution of materials in the original geometry. Typically one of the polygonal cut-

cells will be homogenous, and the other will not. This mixed-cell interface residual represents a meshing error that may affect the accuracy of our numerical discretizations.

If the cross-sections are not adjusted, the reconstructed homogeneous cut-cells will contain different amounts of material than their respective counterparts in the true geometry. In particular one cut-cell will gain mass and the other will lose mass, but the total mass present in the mixed-cell will not be preserved. There exist at least three reasonable approaches to treat this effect. We discuss each treatment in the sections that follow.

It is worthwhile to note that two of the three techniques introduced below rely on detailed geometry information that may not be available in real-world scenarios. In many instances, inputs to mesh generation routines may lack the fidelity needed to fully reconstruct the original geometry. An example might be a mesh generation package that takes a computer-aided design (CAD) input that has been rasterized, thus eroding the original geometric detail. Another notable issue that may arise concerns the desire to perform transport calculations on cut-cell meshes in multi-physics problems. In this scenario it is quite possible that the discretizations and numerical techniques used for other physics associated with the problem may place restrictions on what form of meshing may be used. In practice the best interface mass treatment will be dictated by the specific code implementation, desired accuracy, and the ability to control meshing

characteristics. In this work we have implemented three separate techniques for comparison.

3.3.1 Mesh refinement (no-treatment=NT) strategy

The simplest approach to handling the interface residual is to ignore its presence and accept the associated mass change. We shall refer to this approach as the no-treatment (NT) residual mass treatment. As the mesh is refined the total number of cut-cells increases while the total interface length converges to the line integral over the true curved geometry. Thus the mean interface residual associated with cut-cells decreases rapidly with mesh refinement. We note that this treatment is the only one presented that appropriately handles an interface reconstruction between two user-defined material regions that are in reality the same material. This means that the ability of our LDGFEM to preserve the linear solution in space for any mesh size is retained only when no interface residual treatment is used on cut-cell meshes. The other interface residual treatments do not retain this limit. Even though we point this result out we should note that in practice this result is purely academic as cut-cell meshes should not be generated for homogenous domains.

For the specific cut-cell meshes generated in this work, one may show that the interface residual volume represents the difference in areas between a circle and a special inscribing polygon. For real-world meshes the underlying behavior is not as simple, but

it is believed that such meshes could always be constructed such that the total interface residual throughout the domain is a smoothly decaying monotonic function of mesh refinement.

3.3.2 Cell-wise mass preservation (CWMP) treatment

The traditional approach to handling the interface residual that is used in fluid dynamics applications is based on the desire to preserve the total mass contained in a mixed-cell. For neutron transport this means modifying the macroscopic cross-sections in each sub-cell so the mass present in the polygonal representation of the material region post-meshing is the same as that in the original geometry. Since density is linearly proportional to the cross-section we may write this mass preservation as

$$\sum_i \sigma_i^{poly} V_i^{poly} = \sum_i \sigma_i^{true} V_i^{true}. \tag{3.7}$$

Here σ_i^{true} and σ_i^{poly} represent the true and polygonal reconstructed macroscopic cross-section, respectively, for material constituent i. The volumes within the mixed-cell occupied by material constituent i for the true geometry and the polygonal representation are given by V_i^{true} and V_i^{poly}, respectively. Referring to Eq. (3.7) we see that in order to preserve mass in a specific cut-cell given a polygonal representation that underestimates the true volume, the density of the material in that cut-cell should be increased. Likewise, when the polygonal representation overestimates the volume of a

particular cut-cell, the density is decreased to preserve the mass. If we define the volume differential associated with cut-cell i as

$$\delta V_i = V_i^{true} - V_i^{poly}, \tag{3.8}$$

then the appropriate cross section to use in a reconstructed polygonal cut-cell is simply:

$$\sigma_i^{poly} = \sigma_i^{true}\left(1 + \frac{\delta V_i}{V_i^{poly}}\right) \tag{3.9}$$

We note that δV_i may have either sign. We have defined the interface residual such that the quantity has a negative or positive sign depending on whether the underlying cut-cell has been under- or over-represented by its polygonal representation. If the cut-cell volume is underestimated by the polygonal representation, then the volume differential is given a positive sign. If the cut-cell volume is overestimated by the polygonal representation, then it has a negative sign. This interface residual treatment is applied to all mixed-cells present in the mesh. We will refer to this technique as the cell-wise mass preservation (CWMP) treatment.

We now mention two important notes concerning the implementation of the CWMP technique. This fix-up must be applied for all material properties. More specifically Eq. (3.9) is applied to the total cross-section, the scattering-cross section, the fission cross-section, and the fixed source terms. We also note that this procedure should be performed individually for the Cartesian and curvilinear meshes since the underlying mixed-cell interface residuals attain different values on these two meshes.

The CWMP treatment is fundamentally sound in that there are physical reasons for preserving the mass in a cell. In particular for a k-eigenvalue problem it is well known that preservation of the total fissionable material is important. There are some drawbacks to the CWMP approach, however. First, such a treatment requires some additional computational effort. Possibly more importantly this process creates many additional boundary layers in the modeled geometry. Finally, the CWMP treatment has the potential for causing significant statistical noise in the numerical solution because the amount and location of the moved mass will exhibit a non-smooth behavior as the mesh is refined.

3.3.3 Region-wise mass preservation (RWMP) treatment

The final interface residual treatment implemented in this work will be referred to as the region-wise mass preservation (RWMP) treatment. This technique is motivated by a desire to damp the statistically noisy behavior exhibited by the CWMP treatment as well as reduce the memory requirements for the computational solution. While the location and relative size of the interface residuals for various meshes are statistically varying properties, the total interface residual accrued over a single material interface in the problem exhibits smooth behavior as the mesh is refined. For the test geometries we have run this is a very smooth, monotonically decreasing function. This is because mesh refinement for our test geometries is equivalent to inscribing a circle with polygons of increasing dimension.

The RWMP treatment is the region-wide analog of the CWMP treatment. In this treatment cross-sections are specified on a region-wide basis for each distinct, contiguous material region in the problem domain. Starting with the cut-cell mesh the integral interface residual is tallied for all mixed-cells on the periphery of material constituent A. Then the cross-section for each specific problem region is set to a single value which preserves the mass between the true geometry of the region and its representation by the combination of homogenous grid cells and cut-cells alike.

The RWMP treatment is more difficult to implement than the CWMP treatment but requires less memory since in general each cut-cell does not have distinct cross sections as in the case of CWMP. We also note that RWMP is in general more resistant to statistical oscillations and mesh imprinting.

3.3.4 Summary

In Section 3.3 we introduced three different interface treatments for accounting for the mesh discrepancies that are created by reconstructing curvilinear material interfaces with line segments. If not specifically accounted for, the total mass of material present in each material domain as meshed does not match that in the original geometry.

The NT method is used to denote the treatment where no attempt to account for this effect is undertaken. Instead, refinement of the base orthogonal mesh encasing the problem is used to gradually eliminate this error essentially by representing each curvilinear interface with more and more line segments.

The CWMP method is used to denote the treatment where cross-sections in each mixed-cell are perturbed in such a way as to preserve material mass in each mixed-cell. The RWMP method is used to denote the treatment where a single cross-section for each distinct material region in the geometry is used for all underlying cells in such a way as to preserve the integral mass present in the true geometry and the meshed representation for that material region.

CHAPTER IV

TEST PROBLEMS

4.1 Test problem techniques and evaluation metrics

In this section we detail several metrics that shall be used in the evaluation of the performance of the cut-cell methods for the various test problems.

4.1.1 The method of manufactured solutions

The method of manufactured solutions is a very common technique used to evaluate the performance of numerical methods. When applied to the transport equation in order to evaluate the performance of a spatial discretization, the method proceeds as follows. The evaluation begins by choosing a problem domain, material properties, and an assumed analytical angular flux. The presumed angular flux is inserted into the right side of the transport equation forming a total source term. In operator notation we denote this simply as:

$$L\psi - P\psi = Q \tag{4.1}$$

Here L denotes a "loss" operator and includes the streaming, collision, and out-scatter terms. The P term denotes a "production" operator and includes the in-scatter source only. Though in theory one may choose to employ a scattering medium, it is sufficient and convenient to select a purely absorbing medium. With the total source term defined by the presumed angular flux, one then applies the numerical method to the problem. If one chooses a purely absorbing medium, then the manufactured solution test problem

reduces to inverting the loss operator. For our purposes this is equivalent to performing one space-angle sweep using the numerical method to be evaluated. Following this space-angle sweep, one obtains the solution for the angular flux for the discretized equations on the selected space-angle grid. The convenience of the manufactured solution is that the discretized solution may now be directly compared to the analytical solution.

For this work we have made extensive use of manufactured solutions in order to evaluate the performance of our numerical methods. The power of this technique is that methods may be specifically evaluated in certain limits and the behavior of the methods may be isolated to particular characteristics. For instance, a spatial discretization believed to be exact for linear solutions may be evaluated using a manufactured solution that is linear in space. If this hypothesis is true, then any linear angular flux profile chosen for the manufactured solution will generate the exact analytical solution on any space-angle grid.

4.1.2 Scalar flux norms

In order to quantitatively evaluate the performance of our numerical methods, it will be necessary to measure the error associated with solving the discretized equations associated with a numerical method on a particular space-angle grid. Since the parameter most commonly of interest to transport computations is the scalar flux, this parameter

will be chosen as the metric of interest. In general the Euclidean norm will be used to quantitatively evaluate the discrepancy between the solution of the numerical method ϕ_h and the analytical solution ϕ. The definition of the Euclidean norm is simply:

$$L_\phi^2 = \sqrt{\int_\Omega dV \left(\phi - \phi_h\right)^2}, \tag{4.2}$$

where the spatial integration is performed over the entire problem domain Ω. If we consider for a moment a general test problem, we note that while the analytical flux ϕ is an arbitrary function, the discretized solution ϕ_h for our methods is a piece-wise linear function defined on polygonal elements comprising the domain of an RZ cylinder.

For this work we have chosen to evaluate the scalar flux norm given in Eq. (4.2) by accruing contributions over each spatial cell using quadrature integration. Thus if we denote the accrual over every cell as $\sum_{cells}$, the quadrature formula in an arbitrary cell as $\sum_{qdpts}$, and an arbitrary quadrature weight as w, we have

$$L_2^\phi = \sqrt{\sum_{cells}\left(\sum_{qdpts} w\left(\phi - \phi_h\right)^2\right)}. \tag{4.3}$$

The quadrature formula used to evaluate Eq. (4.3) is chosen as the same used to evaluate the inner products appearing in the discretizations as derived in Section 2.5. Thus the quadrature formula is only exact when the underlying function is linear within a

polygonal element. We note that in general this is not the case, and in particular, test problems run using the method of manufactured solutions will be subject to this discrepancy. With respect to manufactured solutions this error is inversely proportional to mesh size and is duplicated when inner products of the analytical flux are computed in order to set the total source.

For manufactured solutions, a more rigorous and expensive integration depending on Monte Carlo sampling was initially investigated. When it was determined that this technique did not affect the results, it was abandoned for the simple quadrature formula that neglects the conformity of the manufactured solution to the cut-cell boundaries.

4.1.3 Computational efficiency

The order of convergence for a particular numerical method is a mathematical property that describes the relationship between numerical error and discretization step size. Though this parameter is very much related, it is not in general the fundamental metric by which modelers typically evaluate the performance of a code. To be more specific, it is generally the computational efficiency that will most impact the end users of the software implementing a particular method. More concisely, a better metric for method performance involves either the time to compute a solution to a given tolerance or the tolerance that may be achieved by a method within a given amount of time.

In order to introduce this evaluation metric, we will define computational efficiency as the inverse of the product of error and solution time for a given problem as follows:

$$\eta = \frac{1}{ET}. \tag{4.4}$$

Here η is the computational efficiency, E is a measure of error in a numerical solution, and T represents the time required to achieve that solution. Referring to Eq. (4.4) we see that the computational efficiency normalizes method performance using time. Thus the computational efficiency for two numerical methods is equivalent if one method produces half the error in twice the time relative to another method. For our new spatial discretizations in RZ geometry, this metric will allow for method comparison despite discrepancies in mesh generation time, underlying data structure complexity, and computational cost per grid cell.

For the test problems run in this work, computational time shall be defined as the cpu time elapsing between the time a code begins mesh creation to the time a final scalar flux solution is obtained. For this work no parallelism has been implemented, and thus computational time is roughly proportional to the time required to mesh, setup data structures, and sweep the space-angle grid to a given tolerance on the scattering source.

In order to understand the effects of computational efficiency when applied to the various methods, it is instructive to briefly consider the behavior of computational time as the characteristic mesh size is reduced.

In RZ geometry the total number of grid-cells will scale in a quadratic manner as the mesh size is refined linearly. Thus a base mesh constructed from a 10x20 meshing generates four times the grid cells as a base mesh constructed from a 5x10 meshing. Neglecting cut-cells and mesh generation time we note that the time required to sweep a particular space-angle grid will scale in a quadratic manner with respect to the inverse of mesh step size. This result means that neglecting the contribution from cut-cells, a second order method will asymptotically approach a constant computational efficiency while the computational efficiency for lower order methods will asymptotically decay with mesh refinement.

4.2 Linear flux manufactured solution

The motivation for this test problem concerns the desire to validate our coding of the three basic discretizations, investigate meshing behavior, and investigate the presence of any numerically ill conditioning effects. A simple test of our new method involves application to a problem having a spatially linear flux profile throughout a fixed domain. For this test problem all runs have been performed with the NT interface residual treatment. We begin by presuming an isotropic, spatially linear angular flux

$$\psi(r,z)=\frac{r+z}{2}, \tag{4.5}$$

defined on the cylinder described by:

$$\begin{aligned} r &\in [0,R]=[0,1], \\ z &\in [0,Z]=[0,2]. \end{aligned} \tag{4.6}$$

We define the domain to be a purely absorbing homogeneous medium and thus $\sigma_t=\sigma_a=1$. The spherical meshing procedure described in section 3.1 is used to artificially insert a spherical material interface at the location $x_{\text{interface}}=0.4999.$ For this problem the 1-D spatial variable x is related to the cylindrical coordinates via

$$x=\sqrt{r^2+(z-1)^2}. \tag{4.7}$$

Material properties on each side of the interface are maintained homogenous, but this artificial interface creates cut-cells within the mesh so that we may test our three basic discretizations.

In order to determine the appropriate source term for this manufactured solution, we simply insert the angular flux given by Eq. (4.5) into the left side of the transport equation given by Eq. (2.1). When this is done we obtain

$$Q = \mu \frac{\partial}{\partial r}\left[r^2 + \tilde{z}^2 \right] - \frac{\eta}{r}\frac{\partial}{\partial \omega}\left[r^2 + \tilde{z}^2 \right] + \xi \frac{\partial}{\partial z}\left[r^2 + \tilde{z}^2 \right] + \sigma_A \left[r^2 + \tilde{z}^2 \right], \tag{4.8}$$

which simplifies to

$$Q = \left[\mu + \xi + \sigma \left(r + z \right) \right] / 2. \tag{4.9}$$

The scalar flux corresponding to the angular flux given by Eq. (4.5) is simply

$$\phi\left(r, z \right) = 2\pi \left(r + z \right). \tag{4.10}$$

The boundary conditions implied by Eq. (4.5) are

$$\psi\left(r, z = 0, \underline{\Omega} \cdot \underline{n}_s < 0 \right) = \frac{r}{2}, \tag{4.11}$$

$$\psi\left(r, z = 2, \underline{\Omega} \cdot \underline{n}_s < 0 \right) = \frac{r + Z}{2}, \tag{4.12}$$

and

$$\psi\left(r = 1, z, \underline{\Omega} \cdot \underline{n}_s < 0 \right) = \frac{R + z}{2}. \tag{4.13}$$

Our manufactured solution test problem proceeds using this source and associated boundary conditions. For a given mesh the chosen discretization is applied to produce the scalar flux solution. This scalar flux is then compared to the assumed form given by Eq. (4.10). The resulting error in the scalar flux is then computed for that mesh.

Testing for this problem was performed using meshes of the form $I \, x \, 2I$, where I represents the number of cells along the r-axis, $2I$ is the number of cells along the z-axis, and $2I^2$ is the total number of grid cells in the mesh. The test problem was run for each of the three spatial discretizations introduced in Section 3.2 using all possible meshes starting from a $5\text{x}10$ mesh and proceeding to a $1000\text{x}2000$ mesh. All problems were run with an S_4 angular quadrature set.

Figure 18 depicts a summary of the meshing behavior for this test problem. The number and type of cut-cells resulting from the meshing procedure is shown as a function of mesh refinement. The number of quadrilateral-quadrilateral cut-cells, pentagonal-triangular cut-cells, and total cut-cells are each shown versus the total number of grid cells. We note that for coarse meshes the resulting number of cut-cells formed exhibits the noisy behavior expected from the stair-stepping procedure. As the mesh is further refined the fraction of grid cells forming cut-cells rapidly declines, and the number of mixed-cells becomes a smooth function of mesh refinement. As the cell size increases linearly through subsequent mesh refinement the total number of cells within the mesh increases in a quadratic manner in 2-D. The linear nature of the notional material interface dictates that the resulting number of mixed-cells scales linearly in the mesh size. This result is particularly significant because it means that the computational cost

of the new cut-cell method will be rapidly dominated by sweeping on the orthogonal grid.

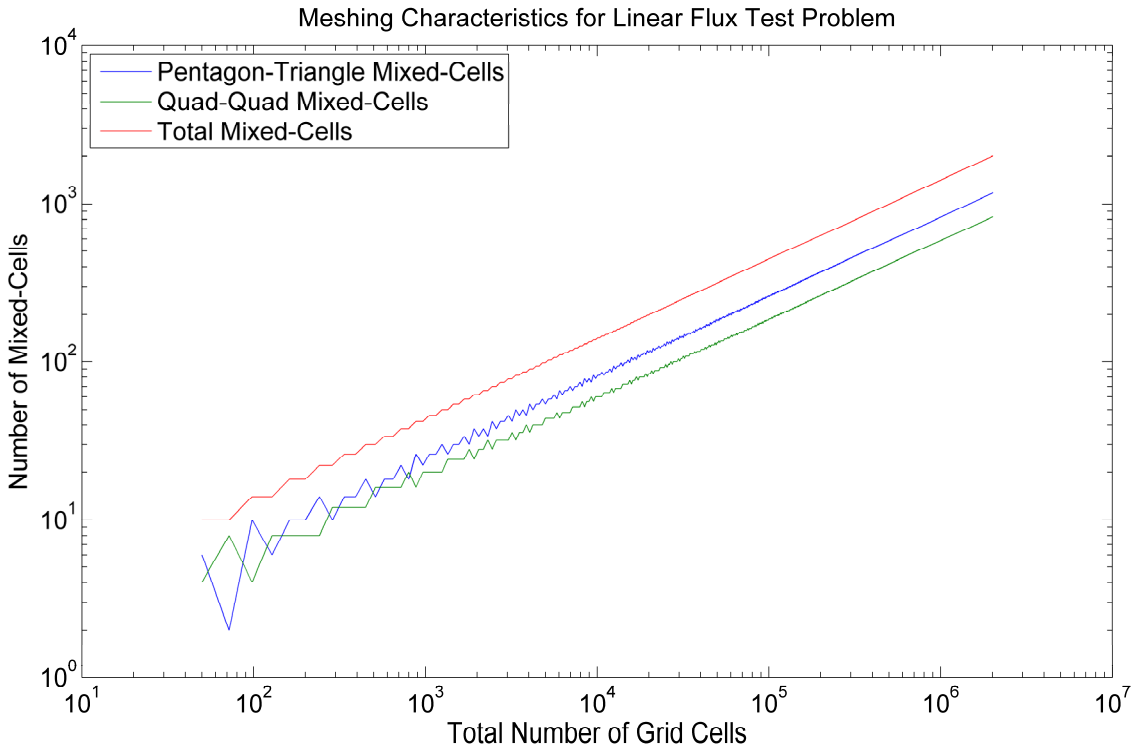

Figure 18. Behavior of meshing characteristics for linear flux test problem

Since the angular flux for this homogenous problem is isotropic and linearly varying in space, each of our three basic discretization schemes should generate the exact solution on any mesh. A plot depicting the resulting L_2 norm of the scalar flux for various mesh sizes is shown in Figure 19. Also shown is the code balance parameter computed from the resulting solution on each mesh. As expected the scalar flux norm is zero to the limits of machine double precision for all meshes. This result provides code validation for each of the methods. We note that both the DFCM and CFCM methods yield errors and balance parameters that are equivalent to that found with the homogenization method which has no mixed-cells. This result is significant in that it suggests no significant numerical difficulties in the implementation of the various methods. In each of the meshes there is no instance of cut-cell interfaces coinciding with grid cell corners and dramatically affecting the numerical methods. Additionally, the scaling parameters introduced into the basis functions appear to function as intended.

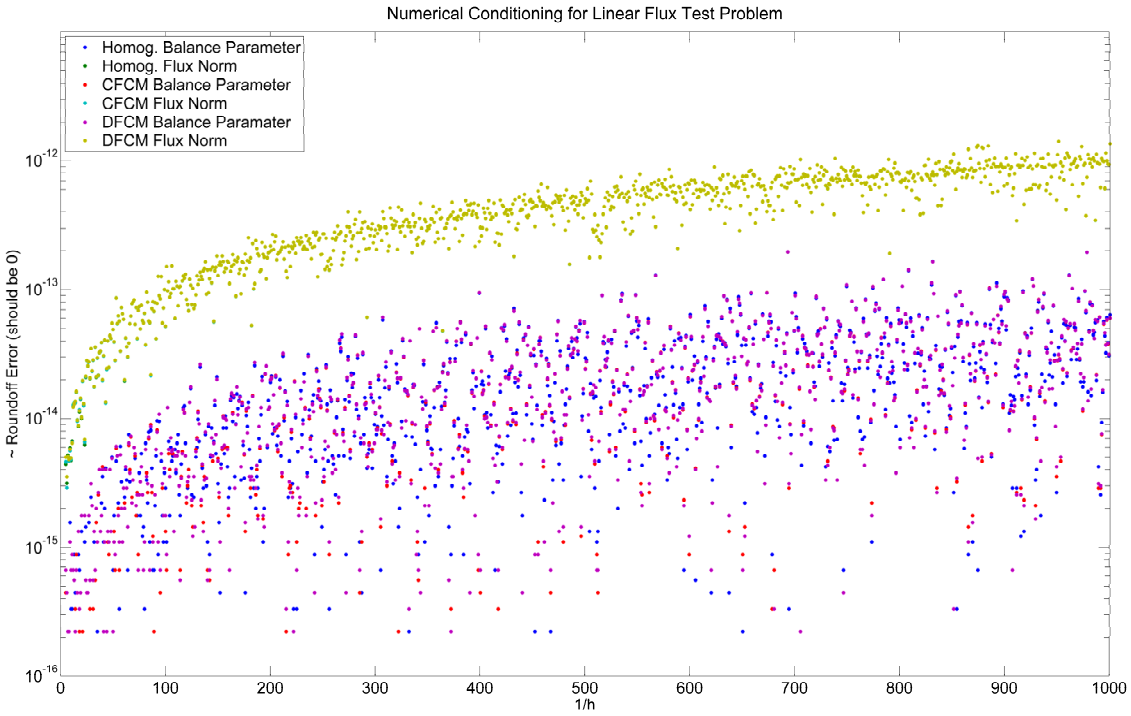

Figure 19. Numerical conditioning of cut-cell methods applied to linear flux test problem

4.3 Spatial kink manufactured solution

The motivation behind this problem concerns the desire to test method performance for problems in which the solution exhibits a discontinuity in its derivative that does not coincide with an orthogonal mesh boundary. The use of a manufactured solution with isotropic angular flux and a spatial kink allows us to analyze the behavior of the spatial discretization itself. We choose an angular flux profile that is the piecewise union of a linear, quadratic, and constant function as follows.

$$\psi = \begin{cases} \psi_A = x^2 & , \quad x \in [0, x_A] \\ \psi_B = \dfrac{x_A^2}{x_A - x_B}(x - x_B) & , \quad x \in [x_A, x_B] \\ \psi_C = 0 & , \quad \textit{otherwise} \end{cases} \tag{4.14}$$

The problem domain is the cylinder described by

$$\begin{aligned} r &\in [0, 1.4999], \\ z &\in [0, 2.9998]. \end{aligned} \tag{4.15}$$

The angular flux has been piecewise-defined as three distinct functions given by ψ_A, ψ_B, and ψ_C. Although the problem is encased in a 2-D mesh, the solution is circularly symmetric and is written as a function of the single spatial variable x. Material interfaces are present at spatial locations x_A and x_B. The underlying materials for each domain are purely absorbing and are also piecewise-defined in each region as follows.

$$\sigma_t = \sigma_a = \begin{cases} \sigma_A = 1 & , \quad x \in [0, x_A] \\ \sigma_B = 2 & , \quad x \in [x_A, x_B] \\ \sigma_C = 0 & , \quad \textit{otherwise} \end{cases} \tag{4.16}$$

For this problem the 1-D spatial variable x is related to the cylindrical coordinates via

$$x = \sqrt{r^2 + \left(z - \frac{2.9998}{2} \right)^2}. \tag{4.17}$$

For simplicity and subsequent use we shall define

$$\tilde{z} = z - \frac{2.9998}{2}, \tag{4.18}$$

which allows us to rewrite Eq. (4.17) in the more compact form

$$x = \sqrt{r^2 + \tilde{z}^2} \ . \tag{4.19}$$

The dimensions appearing in Eq. (4.15) and Eq. (4.17) are chosen to avoid any interference with the material interfaces x_A and x_B. The material interfaces are placed at $x_A = 0.5$ and $x_B = 1.0$. Figure 20 shows a graphical representation of the analytical angular flux profile in the spherical coordinate x. Note that the representation of ψ_C is zero and not shown.

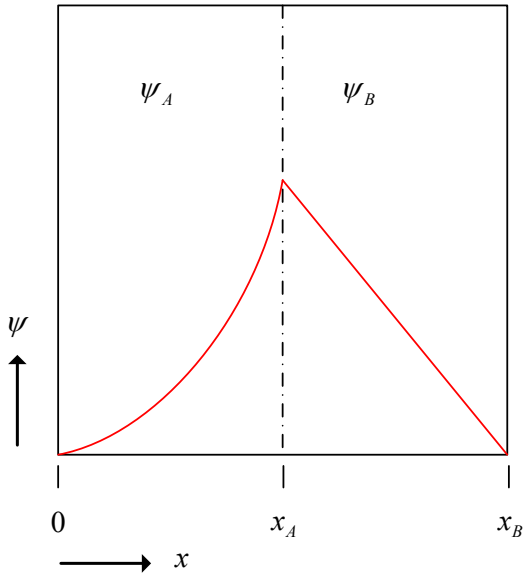

Figure 20. Graphical representation of angular flux profile in spatial kink manufactured solution test problem

In order to determine the appropriate source term for this manufactured solution we simply insert the angular flux given by Eq. (4.14) into the left side of the transport equation given by Eq. (2.1). For the first component we obtain

$$\mu\frac{\partial}{\partial r}\left[r^2+\tilde{z}^2\right]-\frac{\eta}{r}\frac{\partial}{\partial\omega}\left[r^2+\tilde{z}^2\right]+\xi\frac{\partial}{\partial z}\left[r^2+\tilde{z}^2\right]+\sigma_A\left[r^2+\tilde{z}^2\right], \tag{4.20}$$

which simplifies to

$$Q_A=2\mu r+2\xi\tilde{z}+\sigma_A\left(r^2+\tilde{z}^2\right). \tag{4.21}$$

For the second component we obtain

$$\begin{aligned}Q_B=\mu\frac{\partial}{\partial r}\left[\frac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right]-\frac{\eta}{r}\frac{\partial}{\partial\omega}\left[\frac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right]\\+\xi\frac{\partial}{\partial z}\left[\frac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right]+\sigma_B\left[\frac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right],\end{aligned} \tag{4.22}$$

which simplifies to

$$Q_B=\frac{x_A^2}{x_A-x_B}\frac{1}{\sqrt{r^2+\tilde{z}^2}}\left(\mu r+\xi\tilde{z}\right)+\sigma_t\left[\frac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right]. \tag{4.23}$$

For the last components we obtain

$$Q_C=0. \tag{4.24}$$

And so collectively we have

$$Q=\begin{cases} Q_A = 2\mu r + 2\xi\tilde{z} + \sigma_A\left(r^2+\tilde{z}^2\right) & , \quad x\in\left[0,x_A\right] \\ Q_B = \dfrac{x_A^2}{x_A-x_B}\dfrac{1}{\sqrt{r^2+\tilde{z}^2}}\left(\mu r+\xi\tilde{z}\right) & , \quad x\in\left[x_A,x_B\right] \\ \qquad +\sigma_t\left[\dfrac{x_A^2}{x_A-x_B}\left(\sqrt{r^2+\tilde{z}^2}-x_B\right)\right] & \\ Q_C = 0 & , \quad otherwise\ . \end{cases} \tag{4.25}$$

The appropriate boundary conditions are implied by Eq. (4.14) and are

$$\begin{aligned}\psi\left(r,z=0,\underline{\Omega}\right)\big|_{\underline{\Omega}\cdot\underline{n}>0} = \psi\left(r,z=2.9998,\underline{\Omega}\right)\big|_{\underline{\Omega}\cdot\underline{n}>0} = \psi\left(r=1.4999,z,\underline{\Omega}\right)\big|_{\underline{\Omega}\cdot\underline{n}>0} \\ = 0,\end{aligned} \tag{4.26}$$

where the cylinder outward unit normal at position $\left(r,z\right)$ is $\underline{n}$.

The scalar flux corresponding to the angular flux given by Eq. (4.14) is simply

$$\phi=\begin{cases} \phi_A = 4\pi x^2 & , \quad x\in\left[0,x_A\right] \\ \phi_B = \dfrac{4\pi x_A^2}{x_A-x_B}\left(x-x_B\right) & , \quad x\in\left[x_A,x_B\right] \\ \phi_C = 0 & , \quad otherwise \end{cases} \tag{4.27}$$

A graphic depicting a 5x10 meshing of this problem, the resulting material regions, and associated cut-cells is shown in Figure 21.

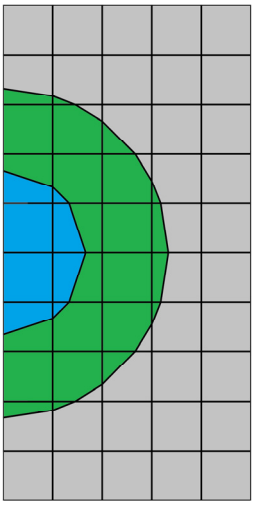

Figure 21. Graphical representation of material distribution and meshing for spatial kink manufactured solution test problem

Testing for this problem was performed using base meshes ranging from 5x10 up to 1280x2560. Each possible mesh size between these ranges was used in order to assess any noisiness that may be exhibited in the underlying convergence trend. All problems were run with an S_4 angular quadrature set. Using the known analytic solution, the L_2 norm of the scalar flux was computed for each mesh. This test problem was run using homogenization, the CFCM, and the DFCM. For each discretization the problem was run for the NT, CWMP, and RWMP interface residual treatments. Thus solutions were obtained for each of the nine possible permutations of these methods. The L_2 scalar flux convergence for all methods is shown in Figure 22.

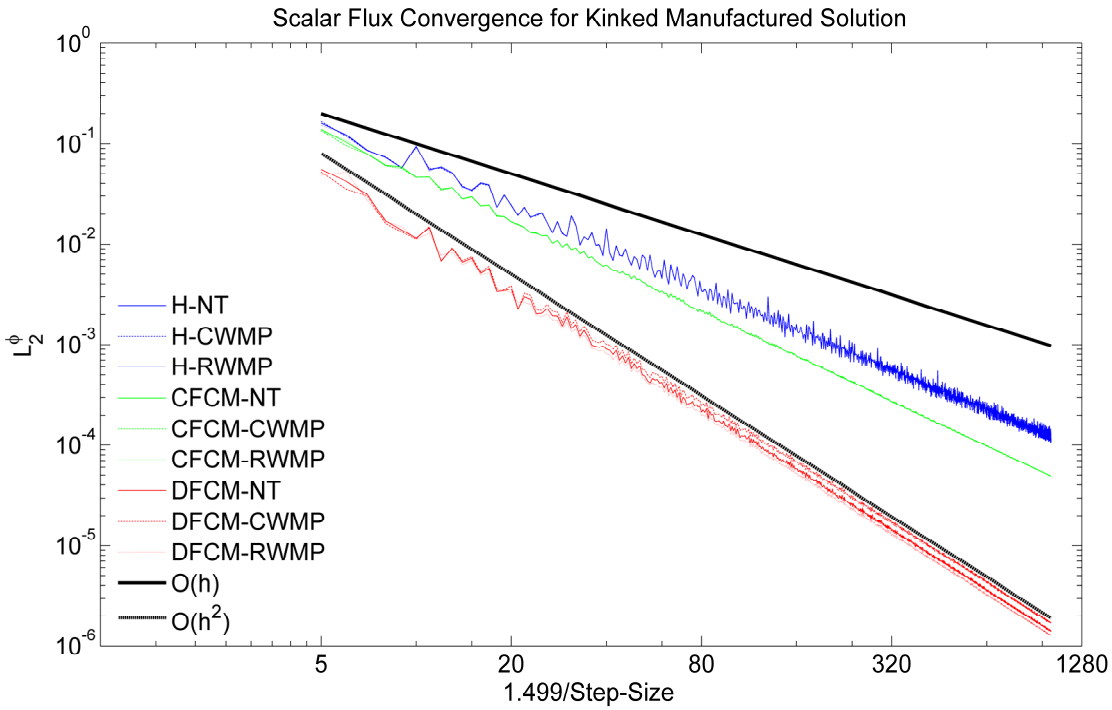

Figure 22. Method performance for scalar flux convergence in spatial kink manufactured solution test problem

Reference trend lines for first- and second-order convergence are shown in Figure 22. Referring to the convergence plot we may see that the DFCM is second-order convergent for the angular flux. The effective order of convergence for both Homogenization and CFCM is $O\left(h^{1.33}\right)$, although CFCM is the more accurate of the two methods. With respect to the interface mass treatments, we note no significant distinction in the methods. We surmise that comparison between the interface residual treatments has apparently been precluded because the angular source computation is

performed with the analytical source term rather than perturbed fixed sources in mixed-cells.

The results depicted in Figure 22 demonstrate increased accuracy for the CFCM and the DFCM relative to homogenization when applied to the same orthogonal mesh base meshes. However, the CFCM and the DFCM incur additional computational cost to achieve this. In an attempt to normalize for this effect, we have computed the computational efficiency as defined by Eq. (4.4) for the scalar flux norm computed on the various mesh sizes. Figure 23 shows the computational efficiency of each method for this test problem.

As the mesh is refined the DFCM obtains almost constant efficiency while the efficiency of the CFCM and homogenization methods peak and rapidly fall off as the mesh is refined. Given our definition of computational efficiency this is expected since the DFCM is second-order convergent and the CFCM and homogenization are first-order. This result is also dependent on the fraction of the mesh comprised of mixed-cells decreasing as the base mesh is refined. We showed in Figure 18 that the fraction of mixed-cells resulting from our meshing procedure is linear in the characteristic mesh size. Looking at the DFCM efficiency at fine mesh sizes, we do note a slight loss of efficiency with increasing mesh refinement. Most likely this slight degradation is caused by differences in CPU performance as problem sizes force computations from CPU

cache to RAM. There is little reason to expect any other factor since the method is shown to be second-order convergent and computational cost is quickly dominated by sweeping on the orthogonal grid cells. We conclude by noting that even for relatively coarse meshes the DFCM method is more efficient than homogenization by roughly an order of magnitude. This may be interpreted as an extra digit of accuracy for the same computational cost or the same accuracy for one-tenth of the computational time. As more accuracy is required, driving further mesh refinement, this discrepancy grows unbounded due to the discrepancy in order of convergence between the two methods.

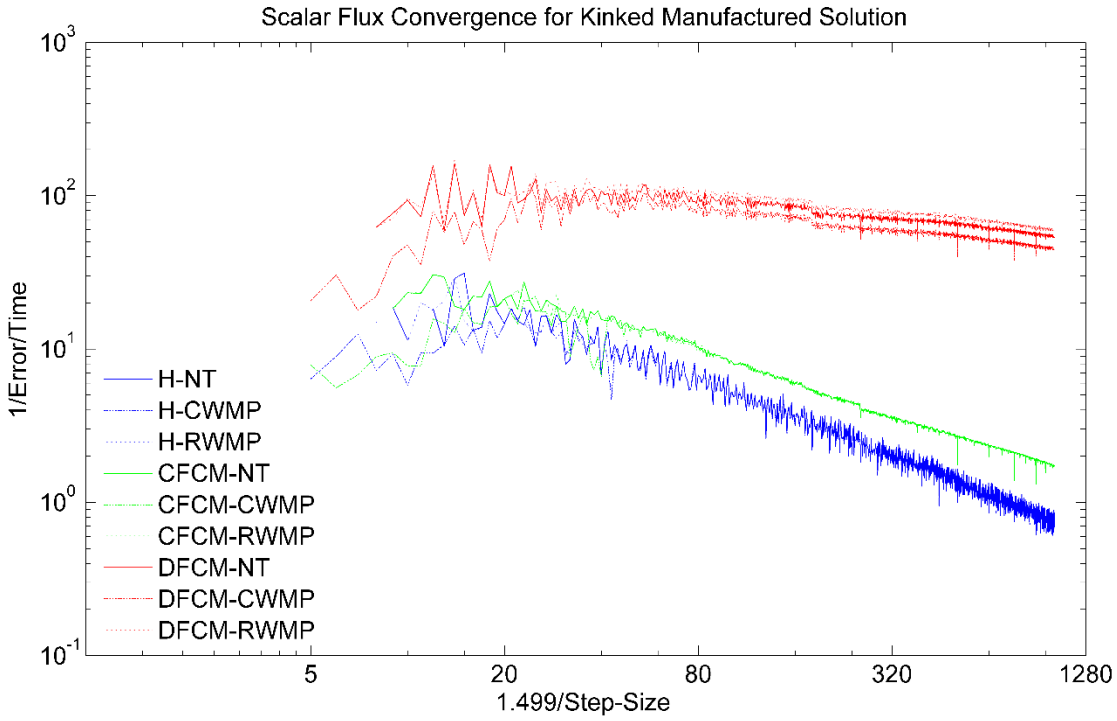

Figure 23. Computational efficiency for spatial kink test problem

This test problem has shown DFCM to be second-order convergent for the L_2 scalar flux norm while the CFCM and homogenization are essentially first-order convergent. With respect to the L_2 scalar flux norm the efficiency of the DFCM method was shown to be asymptotically constant until memory swapping issues are encountered. Starting at relatively coarse meshes, the DFCM method is significantly more accurate than either the CFCM or homogenization.

4.4 Fixed source test problem

A fixed source test problem was run to investigate the performance of the methods for real problems with material interfaces which exhibit ray-effects and lack smoothness.

The fixed source test problem is a 1-D spherical problem consisting of an inner source sphere surrounded by a moderating shell. Boundary conditions for this problem are vacuum on all sides. A graphical representation of this test problem is given in Figure 24 with associated cross-sections.

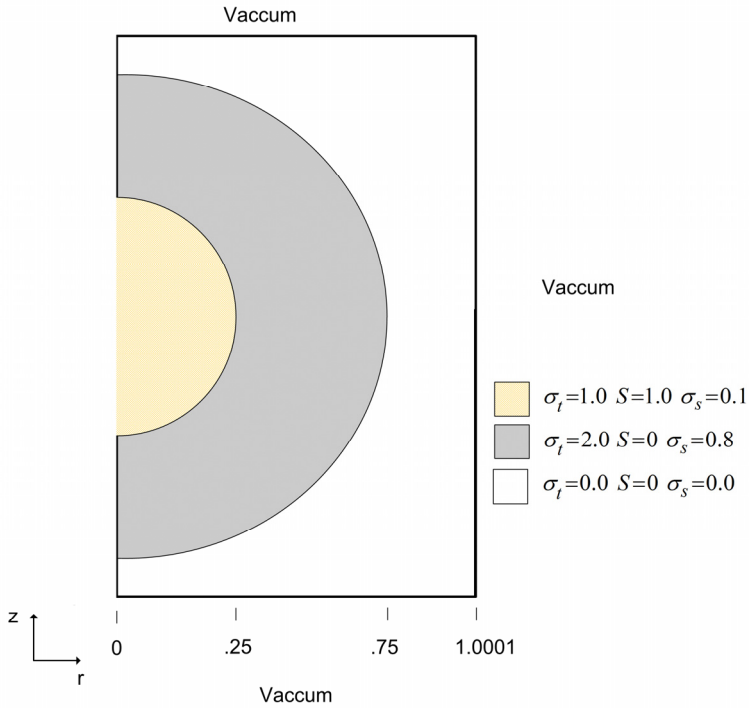

Figure 24. Graphical depiction of problem setup for fixed source test problem

The problem domain is the cylinder described by

$$\begin{aligned} r &\in [0, 1.0001], \\ z &\in [0, 2.0002]. \end{aligned} \tag{4.28}$$

The overall geometry is comprised of an inner fuel region with fixed source and a surrounding spherical shell scattering region as indicated in Figure 24. The radius of the fuel region is taken as 0.25, and the thickness of the moderating shell is set to 0.5. The remaining RZ cylinder is void. This problem has two distinct material interfaces occurring between the fuel and moderator and between the moderator and void regions.

Testing for this problem was performed using approximately six different spatial meshes ranging from base grids of 5x10 to 160x320. In order to investigate spatial convergence for this problem it is necessary to simultaneously refine in both space and angle. A series of test problems were run versus spatial mesh size for each S_8, S_{16}, S_{32}, and S_{48} angular quadrature sets. A scalar flux convergence tolerance of $\varepsilon = 10^{-13}$ was used for all test problems. To assist in error computations a fine-mesh solution was generated for this problem using a base spatial mesh of 640x1280 with S_{48} angular quadrature and the DFCM method with CWMP.

A plot depicting the scalar flux convergence for the fixed source test problem using S_8 angular quadrature is given in Figure 25. The immediate conclusion seen from this figure is that for mesh sizes finer than 40x80 all methods are essentially equivalent with no further scalar flux convergence occurring at finer mesh sizes. This is the expected result occurring due to the insufficient angular resolution provided by the S_8 quadrature set. As the spatial mesh is refined the error becomes dominated by the angular discretization and thus there is no discernible distinction between the spatial discretization schemes. For fine meshes we see that DFCM is dominantly the best performing cut-cell method with CWMP yielding the best results for each cut-cell method. Furthermore, we note that the DFCM/CWMP combination provides essentially the same scalar flux error for all spatial meshes. This indicates that angular discretization errors dominate on all spatial meshes for the DFCM/CWMP technique.

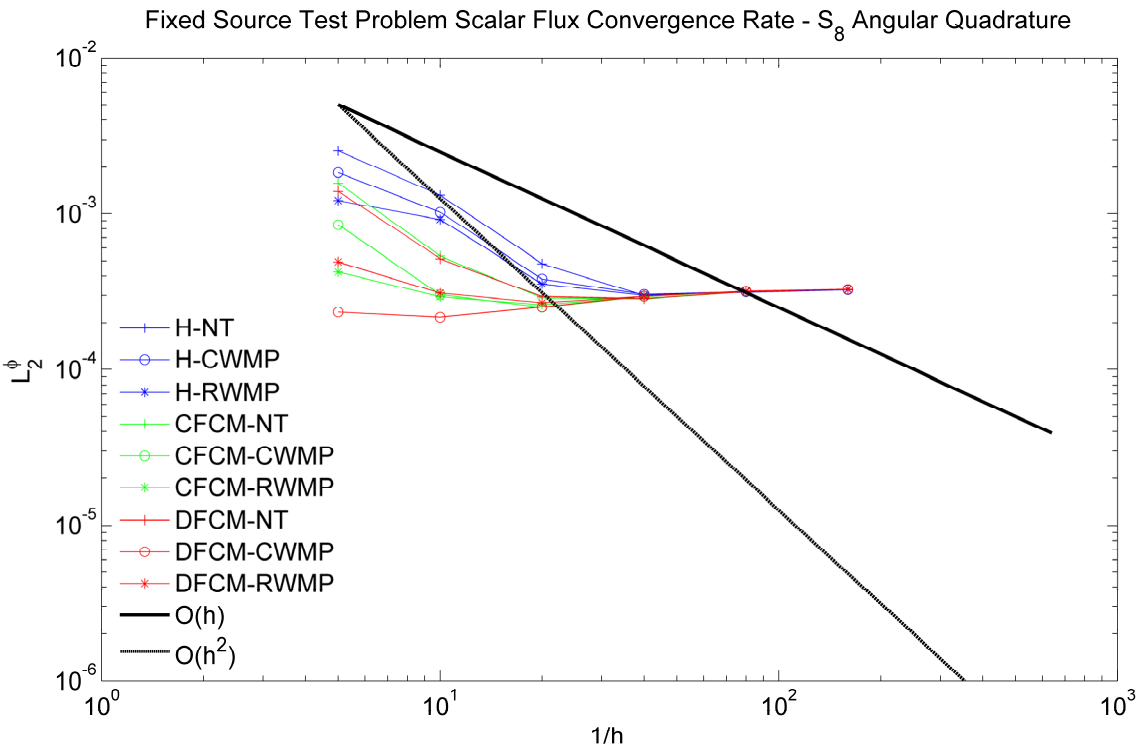

Figure 25. Scalar flux convergence with S_8 for fixed source test problem

A plot depicting the scalar flux convergence for the fixed source test problem using S_{16} angular quadrature is given in Figure 26. Here we note that all methods are essentially equivalent at spatial meshes finer than 80x160. As before, the DFCM cut-cell discretization is clearly the most accurate with CWMP being the most accurate residual technique. Both DFCM and CFCM are an order of magnitude more accurate at coarse mesh sizes. For this problem set it is still difficult to ascertain order of convergence

because all methods roll off so quickly. We again note that application of DFCM in conjunction with CWMP provides similar error across all spatial meshes.

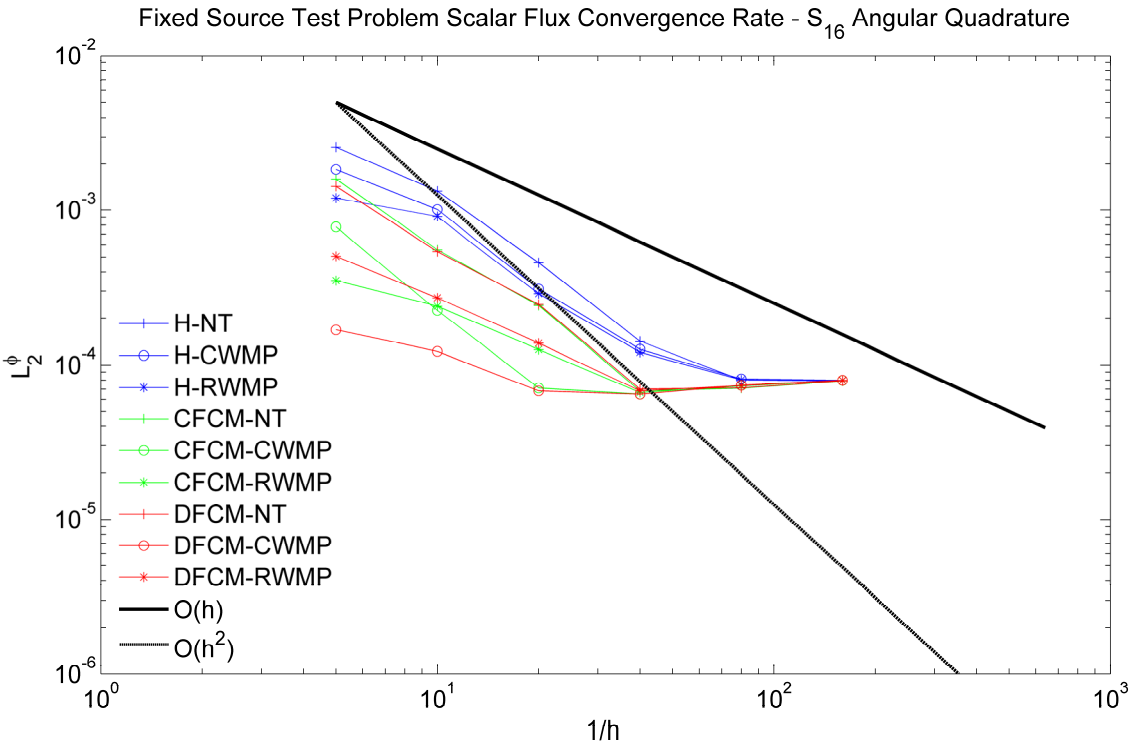

Figure 26. Scalar flux convergence with S_{16} for fixed source test problem

Continuing with angular refinement, the scalar flux convergence results for the fixed source test problem using S_{32} angular quadrature are given in Figure 27. Here we note that, unlike for coarser angular meshes, all methods show distinction throughout most of the spatial meshes, though some convergence occurs at the finest mesh. As before, the

DFCM cut-cell discretization is clearly the most accurate with CWMP being the most accurate residual technique. Both DFCM and CFCM are an order of magnitude more accurate than homogenization at coarse to moderate mesh sizes. Though there is still uncertainty in visually determining order of convergence, the moderate mesh sizes appear to exhibit near second-order convergence while homogenization is somewhat more degraded. This is the first angular mesh that is fine enough such that DFCM shows significant convergence with spatial refinement.

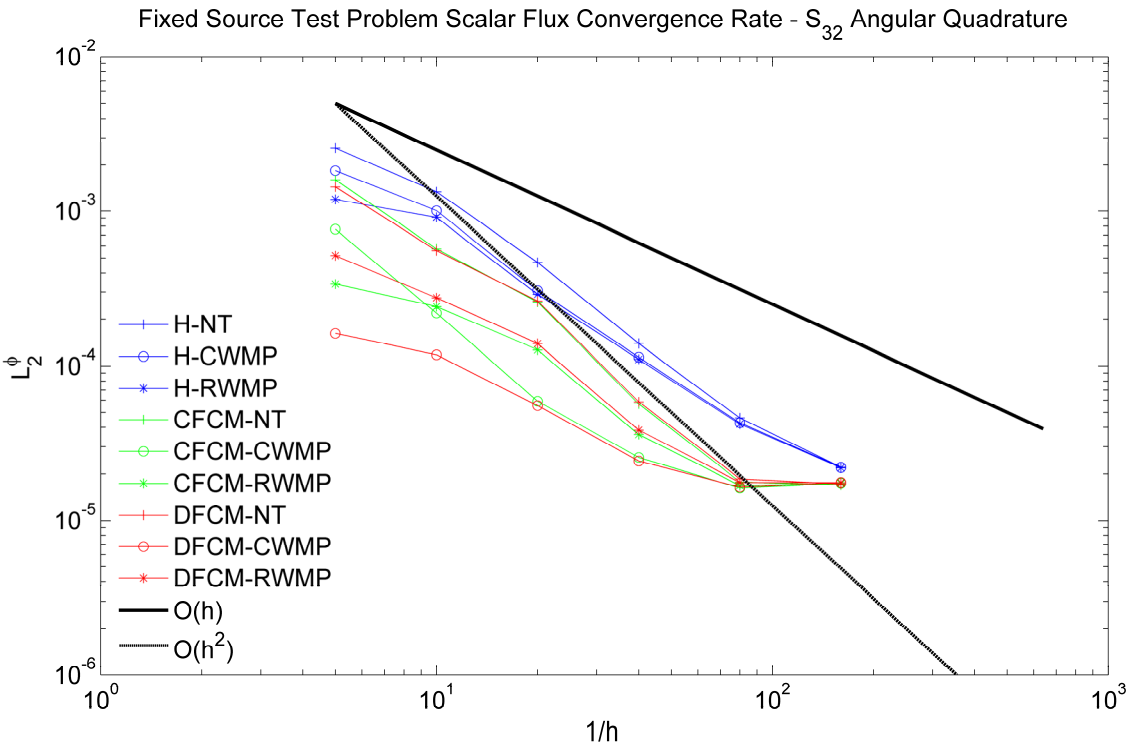

Figure 27. Scalar flux convergence with S_{32} for fixed source test problem

Advancing to the finest angular mesh, the scalar flux convergence results for the fixed source test problem using S_{48} angular quadrature are given in Figure 28. Here we note that for the first time the angular mesh appears sufficiently fine that spatial error dominates the majority of the results shown. We note that this is expected since the fine mesh solution used for comparison was computed on the same angular mesh. Each method shows distinction throughout all spatial meshes, though convergence begins to occur for CFCM and DFCM at the finest mesh. As before, the DFCM cut-cell discretization is clearly the most accurate with CWMP being the most accurate residual technique. Both DFCM and CFCM are significantly more accurate than homogenization with norms between 20 and 30 times smaller for the coarsest meshes shown. Though there is still some uncertainty in visually determining order of convergence, the moderate mesh sizes appear to exhibit near second-order convergence for DFCM and CFCM while homogenization is somewhat degraded. We note that the second order convergence trend is most notable for the NT and RWMP residual treatments, while the convergence of DFCM with CWMP is never as rapid in comparison. This suggests that despite using S_{48}, the DFCM/CWMP method is accurate enough that the angular discretization errors degrade the observed order of convergence. Finally, we note that both CFCM and DFCM seem to obtain equivalence as the mesh is refined. This suggests that the singularities in the reference solution quickly dominate the convergence

behavior. For this reason the interface residual treatments themselves also have little bearing on the convergence for all but the coarsest meshes for this test problem.

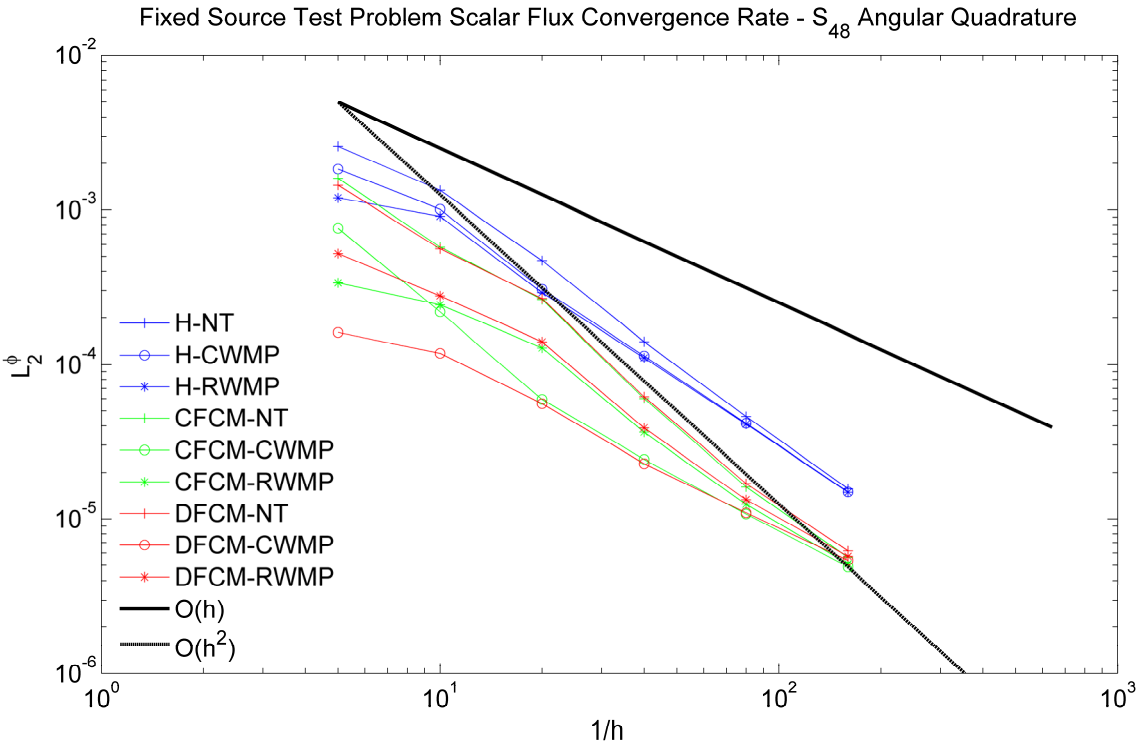

Figure 28. Scalar flux convergence with S_{48} for fixed source test problem

The interpretation of the scalar flux convergence results shown in Figure 25 through Figure 28 are complicated by the interplay in the spatial and angular discretizations as well as the use of a fine mesh reference solution. In order to properly summarize these results a few notes are in order. We begin by noting that the analytical solution to this

problem exhibits smoothness, meaning there are no discontinuities in the angular flux or its first derivative. However, the discrete ordinates angular flux solution for this problem exhibits singularities that degrade order of convergence. Two separate effects occur as the angular meshes are refined. First successive angular refinement diminishes the angular discretization contribution to error, thus allowing spatial discretization convergence to be exhibited in the convergence plots. Secondly successive angular refinement reduces the degree of singularity in the discrete ordinates solution. Noting these effects we may summarize the convergence of the spatial discretization methods for the fixed source test problem as follows. For coarse angular meshes the angular discretization error dominates and all convergence trends rapidly flat line with no subsequent convergence after a few spatial meshes. For moderate fidelity angular meshes the spatial discretization error dominates at coarse spatial meshes and some convergence in space is observed before angular discretization errors and singularities take over. At the finest angular mesh there is in essence no angular discretization error since angular quadrature matches that of the fine mesh reference solution. Here we still observe that the DFCM and CFCM methods show deteriorated convergence at the finest spatial meshes. For the S_{48} runs this degraded convergence is attributed purely to singularities in the underlying discrete ordinates solution. Further angular refinement continues to smooth out these singularities but eventually the spatial convergence for all methods is degraded by the underlying singularities present in the reference solution.

In summary, the scalar flux convergence trends for the fixed source test problem show significantly better convergence rate for CFCM and DFCM relative to homogenization. For moderate spatial mesh sizes both CFCM and DFCM when applied in conjunction with the NT residual treatment appear near second-order convergent briefly before experiencing degraded order of convergence due to angular discretization error. The CWMP residual treatment applied in conjunction with DFCM is so accurate that angular discretization error and singularities prohibit an asymptotic trend to the spatial convergence rate. In all cases the DFCM/CWMP appears most accurate and in the absence of angular effects this accuracy disparity is significant, with errors relative to homogenization of an order of magnitude lower typical.

The scalar flux convergence plots shown in Figure 25 through Figure 28 show that both the DFCM and the CFCM are significantly more accurate than homogenization. In addition, the CWMP appears to be almost universally the most accurate residual treatment. These results establish accuracy, but they do not take into account the increased computational cost associated with these methods relative to homogenization. In order to weigh these considerations, the associated run times were used in conjunction with the scalar flux norms in order to compute computational efficiency. The computational efficiency for the various methods was computed for the fixed source test problem set using the S_{48} angular quadrature. For the purposes of this calculation, run time is defined as the cpu time elapsing from the start of meshing to convergence of the

scalar flux solution. Thus runtime includes time to allocate data structures, time to mesh, as well as time to perform sweeps and obtain convergence of the scattering source. The results for computational efficiency are contained in Figure 29.

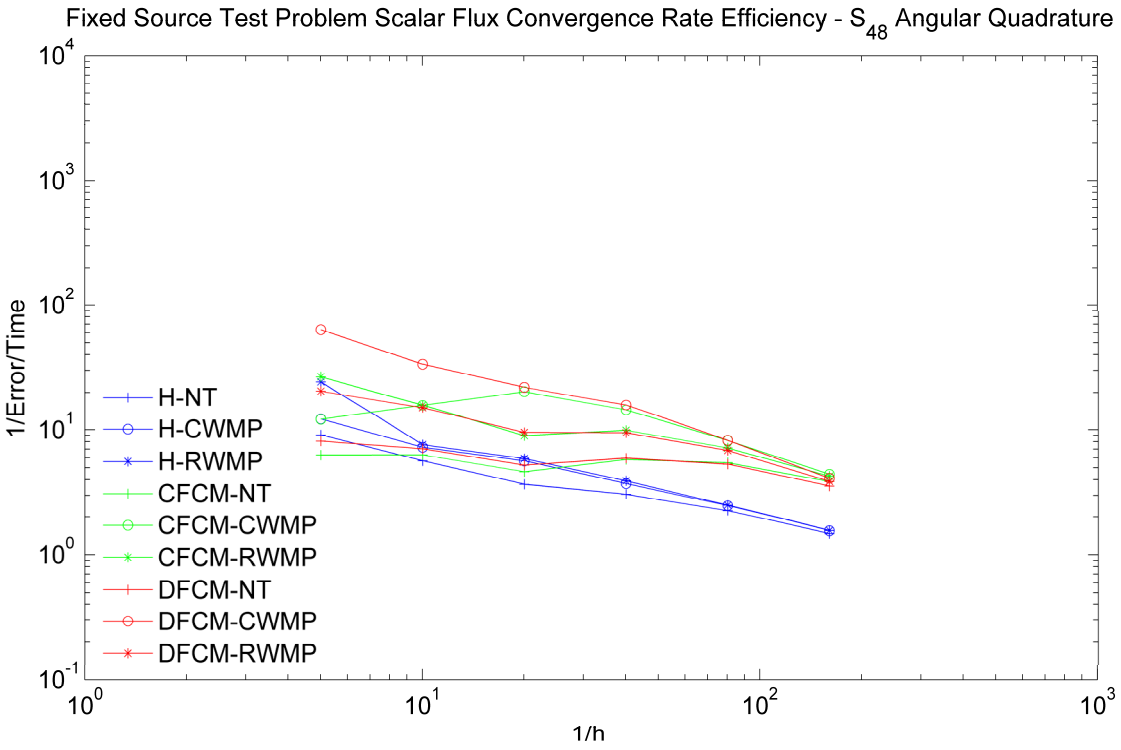

Figure 29. Computational efficiency for scalar flux with S_{48} for fixed source test problem

The computational efficiency results follow those expected from the convergence plot shown in Figure 28, indicating that computational cost is dominated by sweeping on the orthogonal grid. At coarse to moderate mesh sizes the DFCM and CFCM methods are

significantly more efficient than homogenization with typically an order of magnitude difference in efficiency. The efficiency of all methods decays with increasing mesh refinement, which indicates that the interplay of the angular mesh is affecting convergence and efficiency. For a second-order convergent method, we expect efficiency to attain a constant value at coarse mesh sizes removed from processor cache effects.

4.5 Eigenvalue test problem

The eigenvalue test problem is identical to that presented in Section 4.4, but the fixed source material is replaced with a fissile media. For completeness the problem geometry and associated cross-sections are depicted in Figure 30.

The problem domain is the cylinder described by

$$\begin{aligned} r &\in [0, 1.0001], \\ z &\in [0, 2.0002]. \end{aligned} \tag{4.29}$$

The overall geometry is comprised of an inner fuel region with fissile source and a surrounding spherical shell scattering region as indicated in Figure 30. The radius of the fuel region is taken as 0.25, and the thickness of the moderating shell is set to 0.5. The remaining RZ cylinder is void. This problem has two distinct material interfaces occurring between the fuel and moderator and between the moderator and void regions.

Testing for this problem was performed in an analogous way as the fixed source test problem discussed in Section 4.4 with lower angular quadrature orders due to increased computational run times. Solutions were computed on a total of six different spatial meshes ranging from base grids of 5x10 to 160x320. A series of test problems were run versus spatial mesh size for each S_4, S_8, S_{16}, and S_{32} angular quadrature sets. A scalar flux convergence tolerance of $\varepsilon = 10^{-13}$ was used for all test problems. The fine-mesh solution was generated for this problem using a base spatial mesh of 640x1280 with S_{32} angular quadrature and the DFCM method with CWMP. Convergence studies were performed for both the scalar flux error and the underlying eigenvalue.

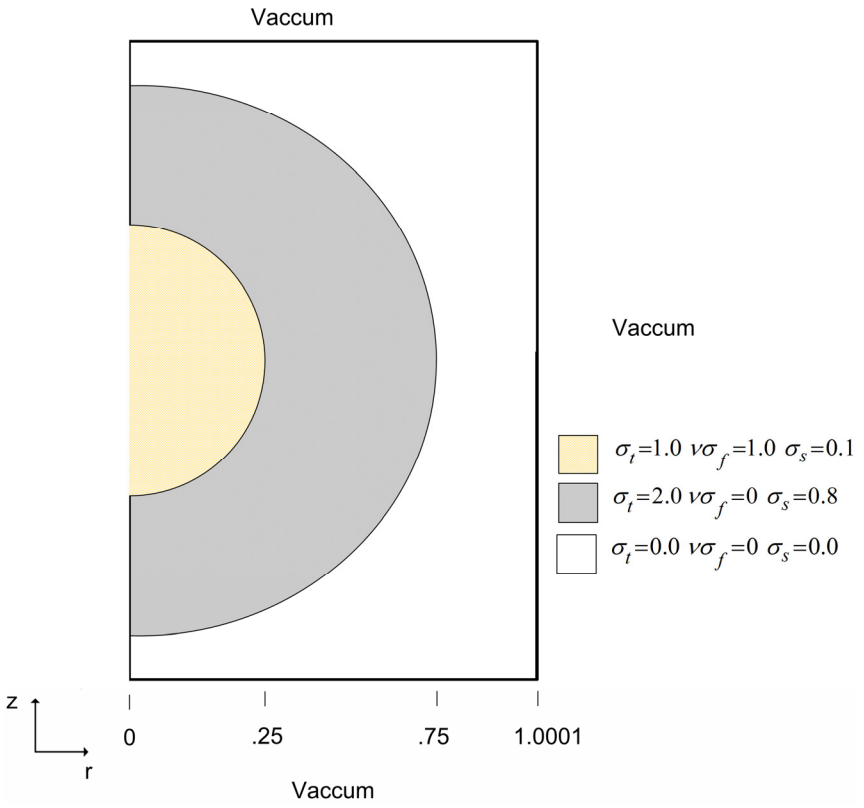

Figure 30. Graphical representation of setup for eigenvalue test problem

A plot depicting the scalar flux convergence for the eigenvalue test problem using S_4 angular quadrature is given in Figure 31. The immediate conclusion seen from this figure is that for mesh sizes finer than 10x20 all methods are essentially equivalent with no further scalar flux convergence occurring at finer mesh sizes. Although little distinction exists between the various methods, DFCM with CFCM is again most accurate.

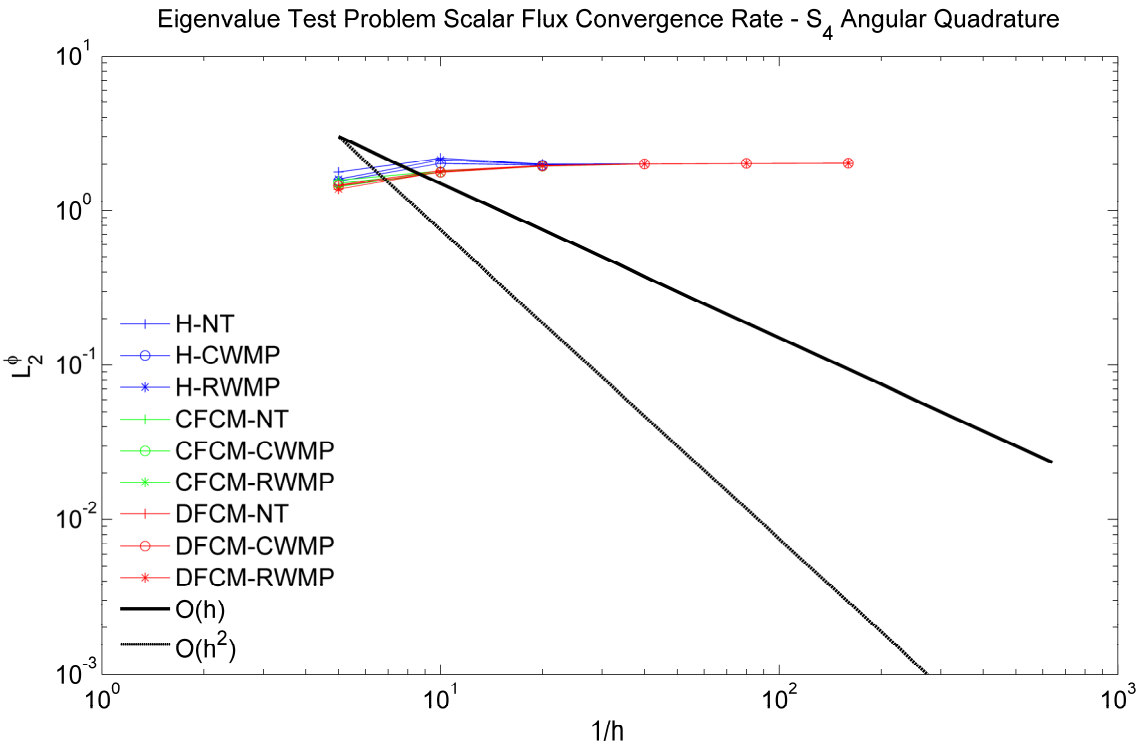

Figure 31. Scalar flux convergence with S_4 for eigenvalue test problem

A plot depicting the scalar flux convergence for the fixed source test problem using S_8 angular quadrature is given in Figure 32. Here we note that only homogenization shows significant convergence across the various spatial meshes. As before, the DFCM cut-cell discretization is clearly the most accurate with CWMP being the most accurate residual technique. For this problem S_8 is still not sufficient angular fidelity to track convergence trends on the selected spatial meshes.

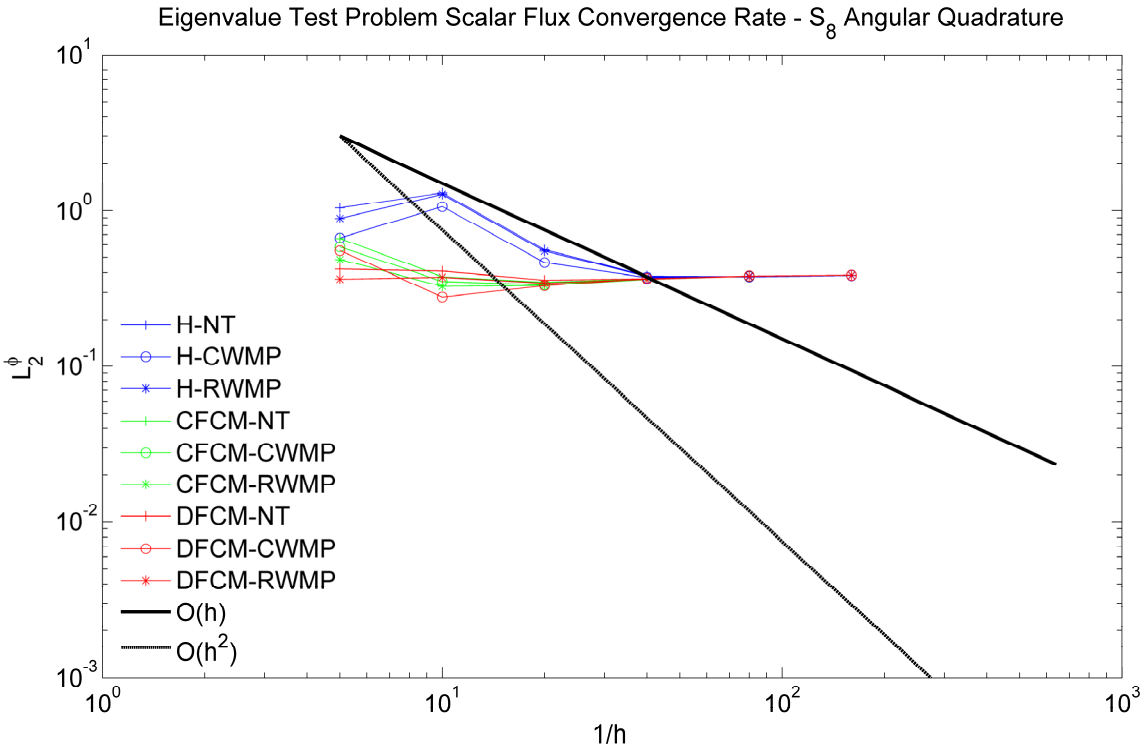

Figure 32. Scalar flux convergence with S_8 for eigenvalue test problem

Continuing with angular refinement, the scalar flux convergence results for the eigenvalue test problem using S_{16} angular quadrature are given in Figure 33. Here we note sufficient angular resolution to observe some amount of convergence across all methods. As before, the DFCM cut-cell discretization is clearly the most accurate with CWMP being the most accurate residual technique. Both DFCM and CFCM are an order of magnitude more accurate than homogenization at coarse to moderate mesh sizes. Though there is still uncertainty in visually determining order of convergence, the moderate mesh sizes appear to show first-order convergence for homogenization.

Though more accurate than homogenization, DFCM and CFCM show flatter convergence than homogenization, indicating that these methods are too accurate to observe order of convergence at S_{16}.

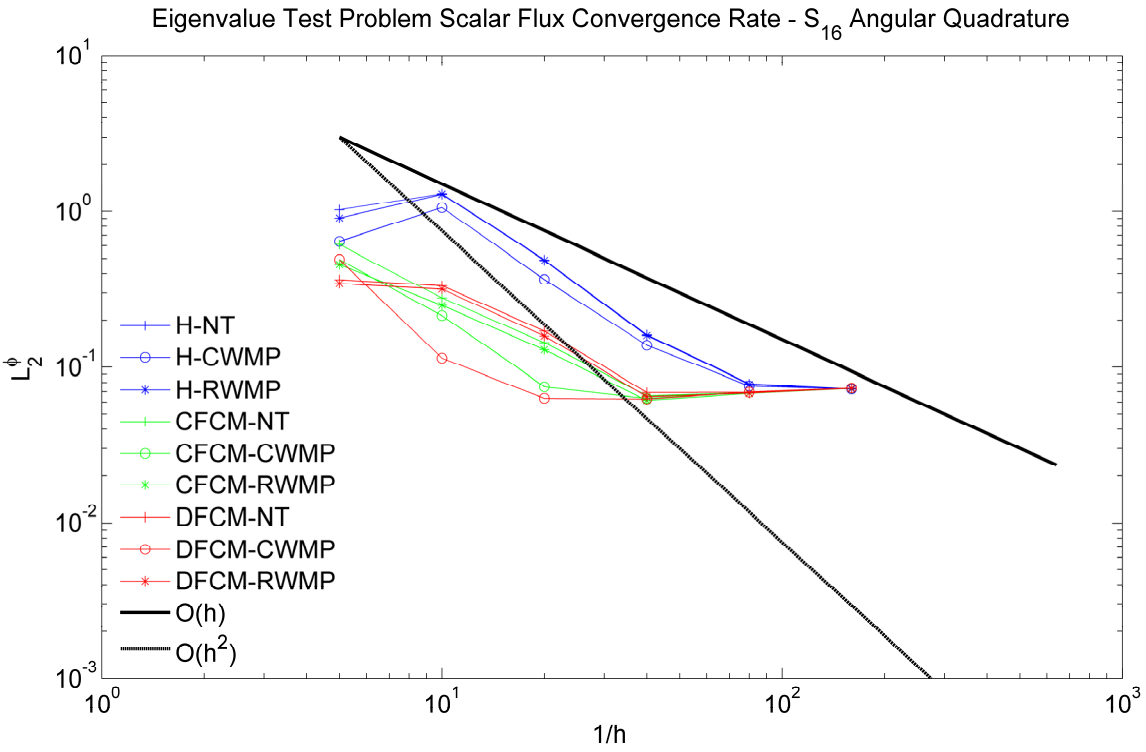

Figure 33. Scalar flux convergence with S_{16} for eigenvalue test problem

Advancing to the finest angular mesh, the scalar flux convergence results for the eigenvalue test problem using S_{32} angular quadrature are given in Figure 34. Here we note that for the first time the angular mesh appears sufficiently fine that spatial error

dominates the majority of the results shown. We note that this is expected since our fine mesh reference solution was computed with the same S_N order. Each method shows distinction throughout all spatial meshes, though convergence begins to occur for CFCM and DFCM at the finest mesh. As before, the DFCM cut-cell discretization is the most accurate with CWMP being the most accurate residual technique. Both DFCM and CFCM are significantly more accurate than homogenization with an order of magnitude better accuracy across most meshes. Though there is still some uncertainty in visually determining order of convergence, the moderate mesh sizes appear to exhibit second-order convergence briefly for DFCM and CFCM while homogenization is somewhat degraded. We note that the second-order convergence trend is most notable for the NT and RWMP residual treatments, while the convergence of DFCM with CWMP is never as rapid in comparison. This suggests that despite using S_{32}, the DFCM/CWMP method is accurate enough that the angular discretization errors degrade the observed order of convergence.

The interpretation of the scar flux convergence rate trends depicted in Figure 31 through Figure 34 are subject to the same space/angle interplay considerations as the fixed source test problem. The results and their interpretation follow the identical discussion laid out in section 4.4.

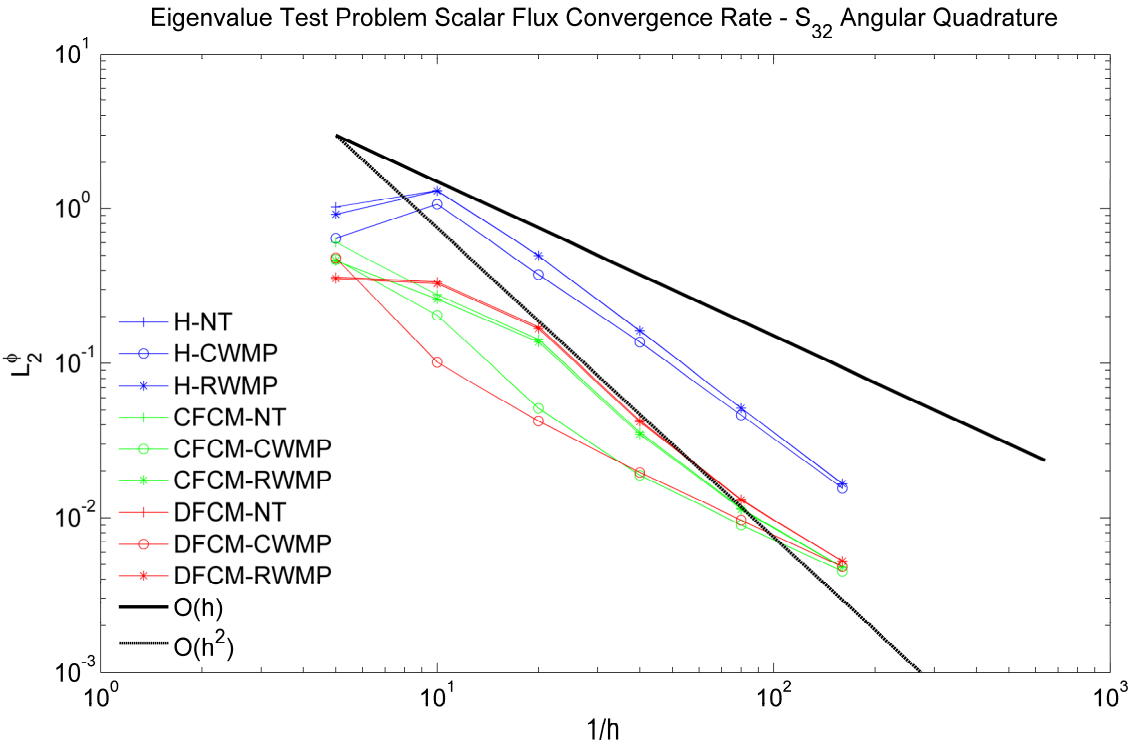

Figure 34. Scalar flux convergence with S_{32} for eigenvalue test problem

The scalar flux convergence plots shown in Figure 31 through Figure 34 show that both the DFCM and the CFCM are significantly more accurate than homogenization. The DFCM is the most accurate method save a few cases with very coarse meshes. In addition, the CWMP appears to be almost universally the most accurate residual treatment. In order to assess computational efficiency for the various methods, the runtime was tabulated for the eigenvalue test problem using the S_{32} angular quadrature. The results for computational efficiency are contained in Figure 35.

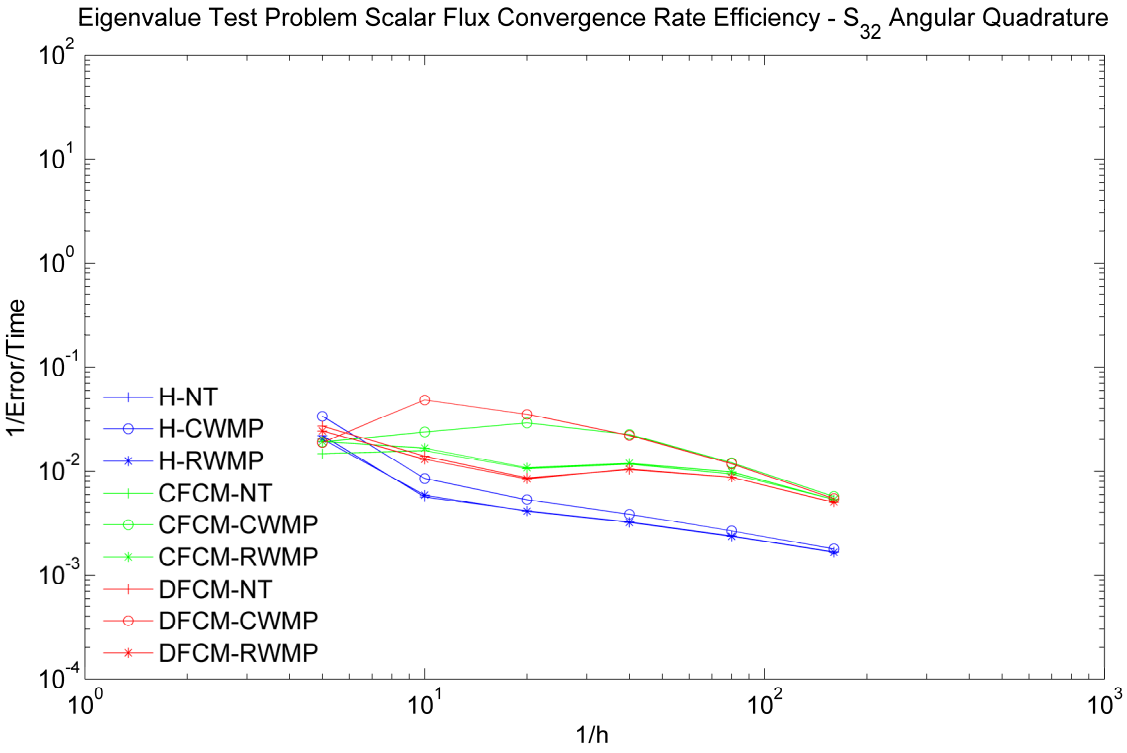

Figure 35. Computational efficiency for scalar flux convergence with S_{32} for eigenvalue test problem

The scalar flux computational efficiency results follow those expected from the convergence plot shown in Figure 34, indicating that computational cost is dominated by sweeping on the orthogonal grid. At coarse to moderate mesh sizes, the DFCM and CFCM methods are significantly more efficient than homogenization, typically with an order of magnitude difference in efficiency. The efficiency of all methods decays with increasing mesh refinement, which indicates that the interplay of the angular mesh is affecting convergence and efficiency. For a second-order convergent method we expect

efficiency to attain a constant value at coarse mesh sizes removed from processor cache effects.

Of primary interest in neutron criticality problems is the eigenvalue itself. For this reason the convergence of the eigenvalue was studied in addition to the scalar flux. In each study we used the same space/angle meshing as was used for the scalar flux convergence study. A plot depicting the eigenvalue convergence for the eigenvalue test problem using the S_4 angular quadrature is given in Figure 36. We immediately note that the convergence results for the eigenvalue are noisy in comparison to the results obtained for the scalar flux. This is to be expected since this quantity is not a mathematical norm and thus error cancellation can occur. It is also clear that convergence of all methods to the same eigenvalue occurs at the finest meshes. Due to the nosiness of the underlying trend data and the sparseness of the results, it is difficult to ascertain anything further from the S_4 convergence data.

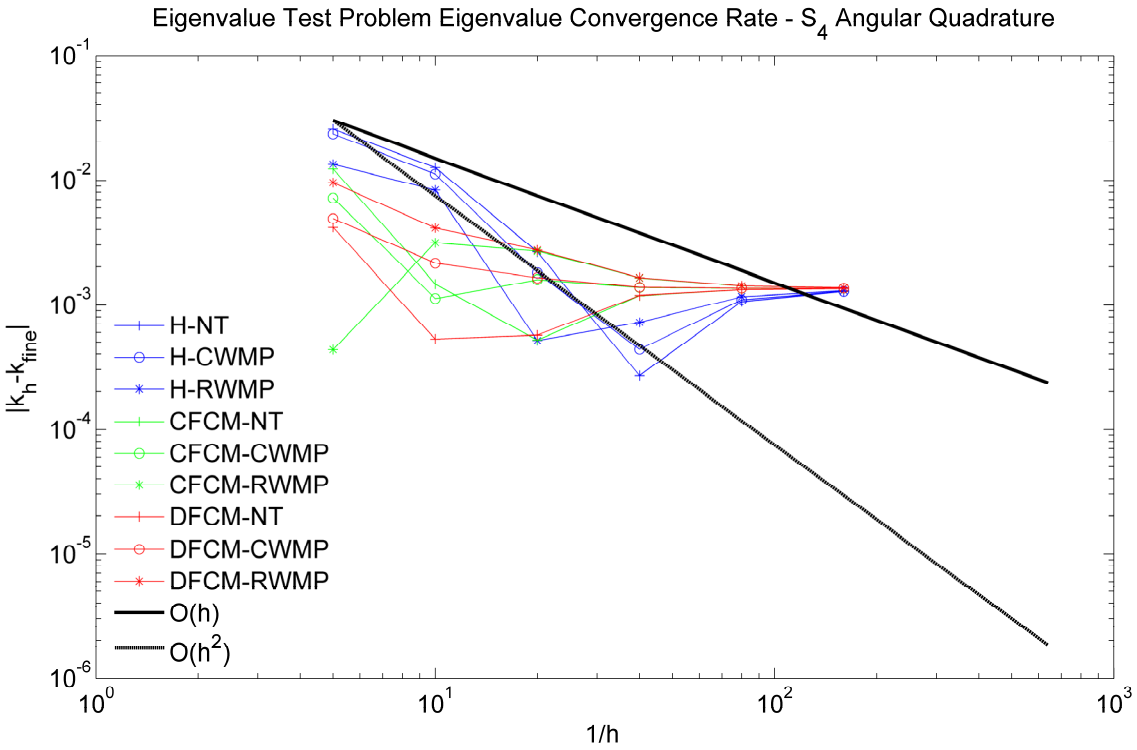

Figure 36. Eigenvalue convergence with S_4 for eigenvalue test problem

The eigenvalue convergence results for S_8 are shown in Figure 37. Despite underlying noisiness we begin to see convergence trend information at this angular resolution. Homogenization in particular seems to be near second-order. DFCM and CFCM are rapidly degraded by lack of angular fidelity. We note that not much may be said of the various residual treatment schemes.

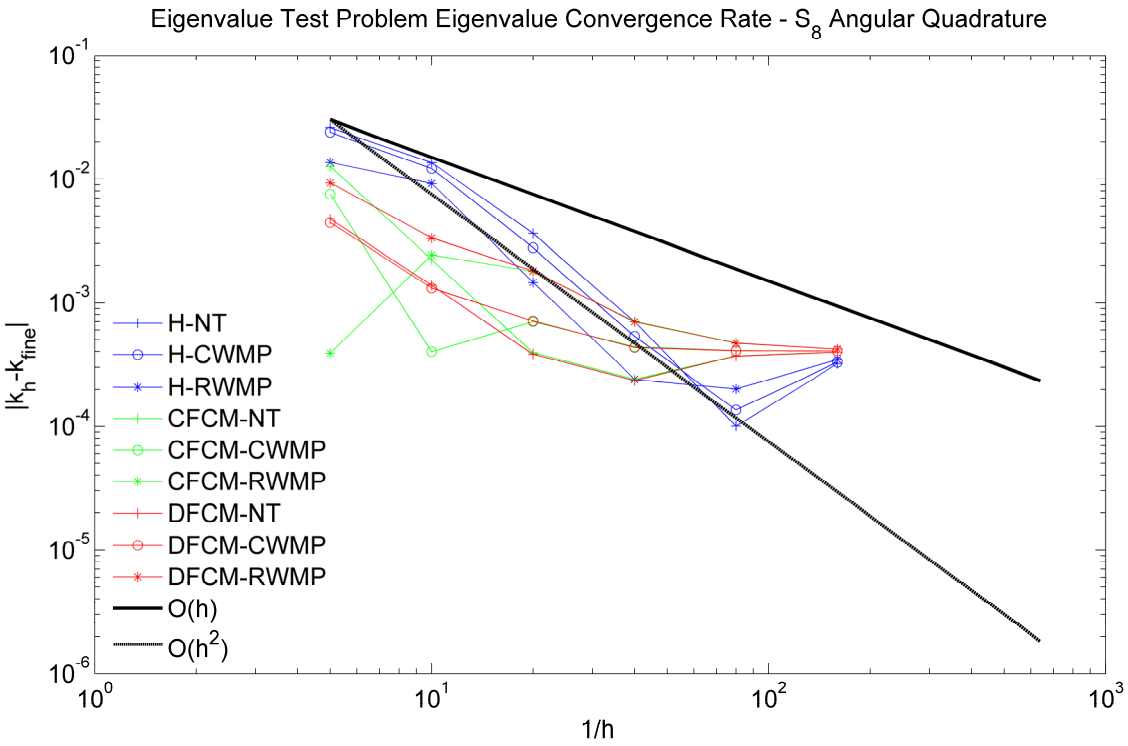

Figure 37. Eigenvalue convergence with S_8 for eigenvalue test problem

The eigenvalue convergence results for S_{16} are shown in Figure 37. These results again indicate that homogenization is near second-order though this is suspect as it appears that the results of this method are passing through the true solution at or near the finest mesh. The noisiness of the DFCM and CFCM make it hard to distinguish a firm order of convergence, although we note that CWMP begins to distinguish itself as the most accurate residual treatment on average.

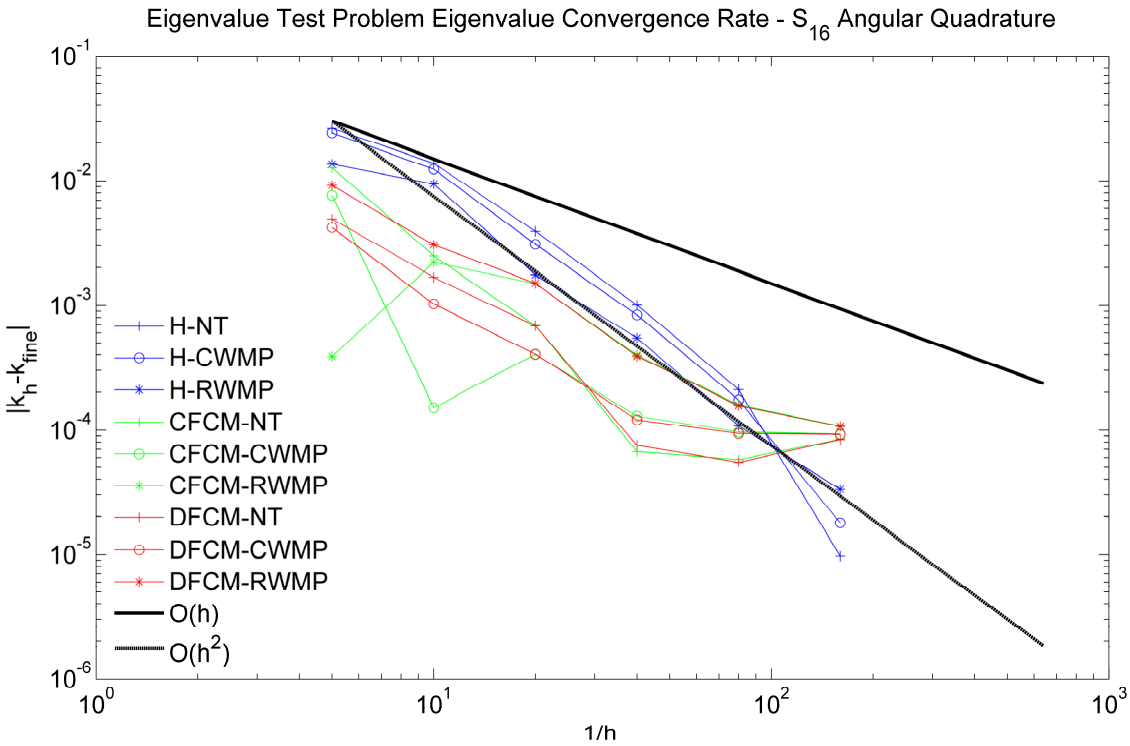

Figure 38. Eigenvalue convergence with S_{16} for eigenvalue test problem

Proceeding to the finest angular mesh, the eigenvalue convergence results for S_{32} are shown in Figure 39. Since these results are computed with an angular quadrature matching that used to generate the fine mesh solution, the space-angle meshing interplay is reduced and some trend information is observable. Most notably the results appear second-order convergent for DFCM and CFCM with homogenization showing slightly degraded effective order of convergence. The effective order of convergence of the eigenvalue for the homogenization treatment is $O\left(h^{1.86}\right)$. The residual treatments distinguish themselves here, and we note that for DFCM and CFCM the CWMP

treatment is most accurate. Finally we note that in this particular plot it is most obvious that the trend behavior exhibits particular meshes where the numerical solution passes through the solution. This is particularly obvious with CFCM and DFCM.

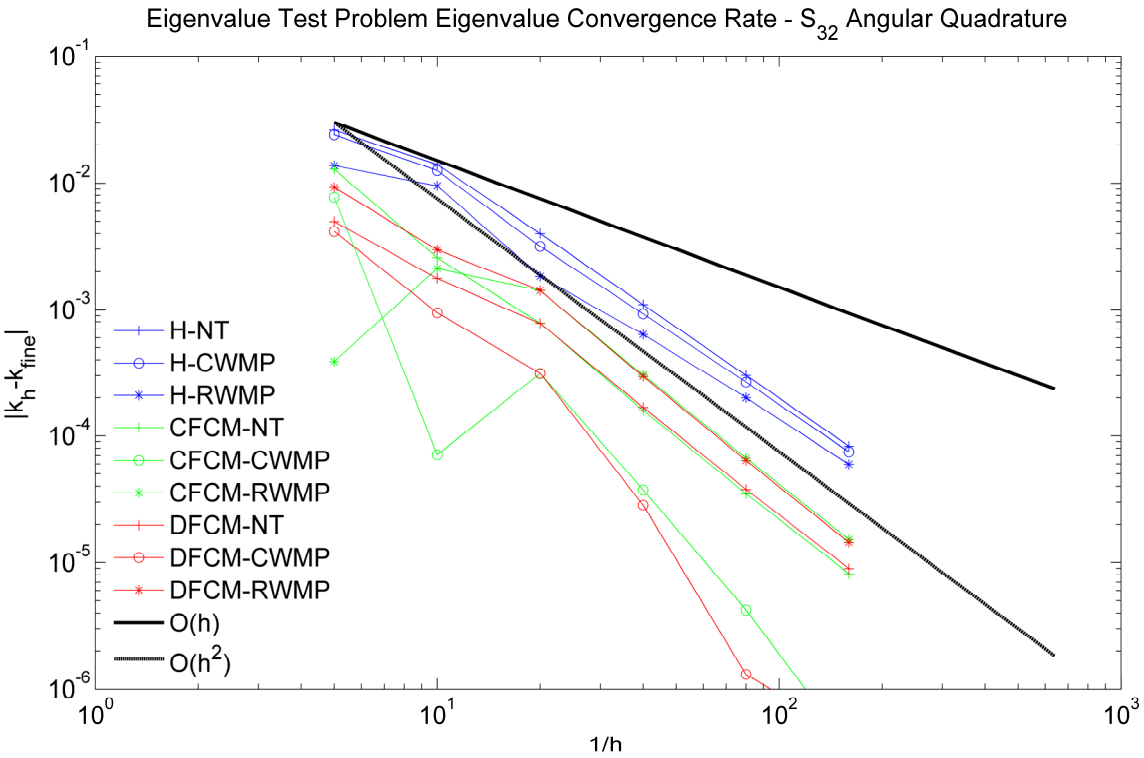

Figure 39. Eigenvalue convergence with S_{32} for eigenvalue test problem

We conclude our study of eigenvalue convergence by presenting a plot showing the computational efficiency for the eigenvalue computation as shown in Figure 40.

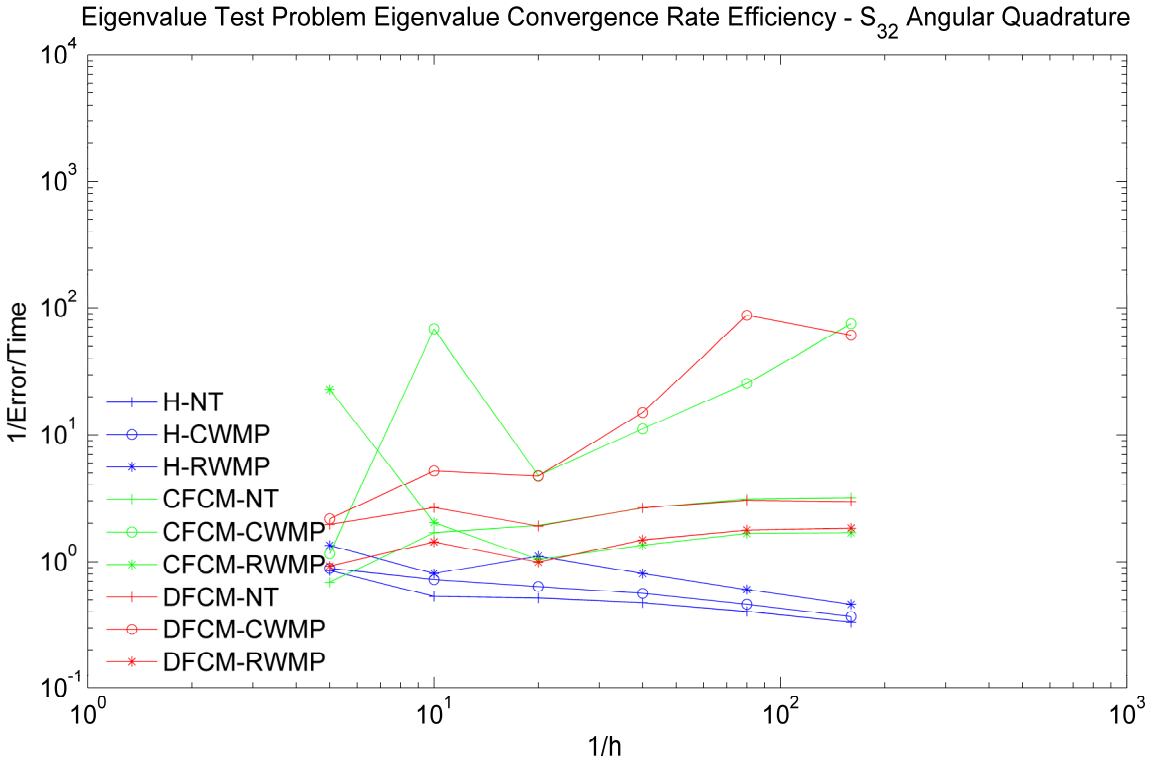

Figure 40. Computational efficiency for eigenvalue computation with S_{48} for eigenvalue test problem

The computational efficiency results follow those expected from the convergence plot shown in Figure 28, indicating that computational cost is dominated by sweeping on the orthogonal grid. At coarse to moderate mesh sizes, the DFCM and CFCM methods are significantly more efficient than homogenization, typically with an order of magnitude difference in efficiency. The efficiency of homogenization degrades as the mesh is refined, indicating that the order of convergence is degraded. The DFCM and CFCM

methods in conjunction with NT or RWMP attain near constant efficiency, indicating the second-order convergence noted previously.

CHAPTER V

DOCUMENT SUMMARY

In this paper we have introduced a new cut-cell discretization for the S_n form of the neutron transport equation. This new method has been developed, implemented, and tested using a test code developed specifically for this purpose. In this section we briefly summarize the major results of each section of this thesis.

In section 1 of this document we introduce the neutron transport equation and briefly outline one technique by which it may be solved using numerical discretization. We then introduce the concept of orthogonal spatial meshing and the issues that arise when this spatial mesh does not conform to the boundaries of the underlying geometry. Next we introduce the concept of a cut-cell taken from the field of fluid mechanics in which the term is used to describe a spatial mesh cell that contains a material boundary. Having introduced the problem of interest, we then summarize some previous work that was performed on the topic of interface reconstruction for the neutron transport equation. The section concludes with a brief summary of our new cut-cell discretization developed in this thesis.

In Section 2 of this document we developed all the equations necessary to solve the one-group LDGFEM equations with isotropic scattering in RZ geometry for basic orthogonal

grids. The section begins by deriving the relevant FEM equations that describe the angular and spatial discretization of the transport equation. Though much of Section 2 is written as if the target is orthogonal grids, these equations are specifically developed such that they may be applied to arbitrary polygons in 2-D. Next the document provides in some detail the description by which inner products may be evaluated for the specific polygonal elements of triangles, quadrilaterals, and pentagons. Section 2 ends with a description of sweeping, which is the process by which the S_N equations are solved on a space-angle mesh.

In Section 3 of this document we introduced three different techniques for handling the spatial discretization of cut-cells using the LDGFEM. The first method is known as homogenization and is the standard historical approach used for treating spatial cells comprised of multiple materials. This method simply replaces the cut-cell with a homogeneous cell consisting of the volume weighted mean of material properties intrinsic to the cut-cell. The second method is an analog of a cut-cell method recently introduced which we refer to as the CFCM. This method employs a piecewise constant cross-section representation of the cut-cell but still uses a single linear flux representation throughout the cut-cell. The third method represents our new technique and is referred to as the DFCM. This new method employs a piecewise constant cross-section representation and a piecewise linear flux representation across the two cut-cells comprising a mixed-cell representing a material interface. Finally, in Section 3 we also

introduce three different methods for treating the material residuals that result in representing curvilinear geometries with linear segments. The simplest residual treatment method is referred to as the NT method and simply ignores the material residual. The RWMP method adjusts the overall material densities such that the total mass of materials is maintained when the geometry is meshed. The CWMP method adjusts material densities in each cut-cell such that the total mass of material present in each mixed-cell is preserved between the original geometry and the mesh geometry.

In Section 4 of this document we introduced various techniques and metrics for evaluating the performance of our new methods. We then introduced a total of four test problems comprised of two manufactured solutions, a fixed source test problem, and a k-eigenvalue problem. The results of the first manufactured solution serve to validate implementation and numerical conditioning of our new methods. The second manufactured solution test problem shows second-order convergence for the scalar flux in problems with curvilinear material interfaces in the absence of singularities for the DFCM method. The fixed source test problem briefly exhibits near second-order convergence for the scalar flux for the CFCM and DFCM before singularities in the underlying solution degrade the order of convergence. Homogenization shows an effective order of convergence for the scalar flux of $O\left(h^{1.6}\right)$. The CFCM and DFCM are both significantly more computationally efficient than homogenization. The eigenvalue problem briefly exhibits second-order eigenvalue convergence for CFCM and DFCM

prior to singularities degrading the convergence. Homogenization again yields an effective scalar flux convergence of $O\left(h^{1.86}\right)$. Both DFCM and CFCM show second order convergence for the eigenvalue itself, while homogenization converges with order $O\left(h^{1.6}\right)$. In all test problems the DFCM method is shown to be the most computationally efficient of all methods evaluated. The CWMP method is almost universally the most accurate residual treatment for all problems run.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In this section we summarize the behavior of the various cut-cell methods developed in this work with some concluding behavior on their performance. We then detail some observations related to future work in this area.

5.1 Conclusions

The performance of the three discretization schemes described in this thesis vary significantly across different problem types and underlying computational result. Below we summarize the behavior of the methods for the problems investigated.

For the manufactured solutions presented the angular flux is smooth and the interpretation of results is straightforward. For this smooth manufactured solution the DFCM technique exhibits second order accuracy for the scalar flux throughout the asymptotic regime. The CFCM and homogenization techniques show degraded order of convergence that are more near first order convergent than second order. Though CFCM is more accurate than homogenization, the difference is minimal. The computational efficiency for DFCM is dramatically better than the other methods.

For the real fixed source and eigenvalue test problems investigated using fine mesh reference solutions the convergence of the methods is less well behaved. At intermediate spatial meshes the CFCM and DFCM methods yield comparable behavior. Although the convergence is not second order, each is more accurate than homogenization in this regime. As the mesh is further refined all methods trend toward comparable accuracy and seem to eventually degrade to first order convergence. This behavior is apparently due to the underlying singularities in the reference solutions used to compute convergence. When the spatial mesh is sufficiently refined the errors due to singularities dominate those associated with the interface techniques and each method becomes equivalent. This conclusion is supported by noting that this behavior is far less pronounced for the convergence of the eigenvalue which is more global in nature.

For the test problems investigated the CWMP interface residual treatment generally performs the best of the three residual treatments investigated. For very coarse meshes the behavior can be sporadic but with relatively moderate mesh refinement the CWMP technique becomes superior to the other treatments.

The computational results provided in this dissertation demonstrate significant advancement with respect to computational transport on meshes exhibiting material interfaces that do not conform to mesh boundaries. In the absence of ray-effects and singularities, the DFCM method has been shown to be second-order convergent for the

scalar flux and significantly more computationally efficient than the current method of homogenization. This result is significant in practice for various reasons. From a practical perspective the increased computational efficiency may be viewed as providing more accurate solutions within a fixed time or providing a solution to a specific tolerance quicker relative to the homogenization treatment. Secondly, the proposed method is particularly relevant because it is a relatively minor modification to the approaches currently implemented in production codes. This means that inclusion of the DFCM method is straight-forward, relatively easy to implement, and provides significant computational advantage. Lastly, the DFCM technique described in this work is readily amenable to other aspects of computational transport of particular concern. The primary objective of concern involves the DFCM's amenability to massively parallel architectures. Since the basic block orthogonal grid is maintained by this method, the currently implemented parallelization schemes may be readily applied. Additionally, the new cut-cell method should be an effective complement easily integrated into adaptive mesh refinement (AMR) schemes in use in production codes.

5.2 Future work

Though the cut-cell methods introduced in this work are essentially generic and have proven quite efficient, there remain several aspects of the general problem that have not been addressed by this work.

The most obvious opportunity for future work involves the application of the DFCM procedure to other geometries. Having been applied in RZ geometry in this work we note that we have already shown implementation in XY geometry via the starting direction flux procedure. However, the method should be readily applicable to 3-D, XYZ geometry as well. In XYZ geometry a mixed-cell will convert a brick element into two general polyhedral cut-cells. The resulting cut-cell mesh may be treated in the same manner described in this work. Implementation and testing in other geometries should be performed to validate this approach and document the computational results.

A second area, and likely the most important work to be done, involves providing an effective source iteration acceleration scheme. We have not tackled this aspect of the problem in this work. There remain at least two obvious ways to approach this problem. The first involves deriving a consistent diffusion cut-cell discretization that may be used as a pre-conditioner to source iteration. The second involves acceleration via the more general Krylov solvers such as the generalized minimal residual (GMRES) algorithm. Though there is no reason to believe that either of these schemes will prove difficult to implement, this has not been proven at this time.

The third area of concern involves the treatment of degenerative cut-cell meshes. Though specifically avoided in this work, in general there is a chance that the meshing procedure described in Section 3 will result in degenerative cut-cells that coincide with

orthogonal mesh vertices. In practice this effect will be seen when the vertices of a linear interface fall arbitrarily close relative to machine precision to the corner of a rectangular grid cell. Though treatment of such conditions should be specifically handled prior to implementation in production codes, there is no reason to believe the underlying performance of the method will be affected. We note that the likely best solution for such occurrences simply involves artificially sliding the interface to a cell corner, thus avoiding conditioning issues and clearly defining inflow/outflows for the sweeping procedure.

The last area for future work that we wish to detail involves the actual mesh generation procedure. For this work we have generated simple meshes whereby the original geometry was exactly known, and we had complete control of how the domain was meshed. Considering various scenarios for the use of production codes, this criterion will in general not be true. For multi-physics calculations in particular, the radiation transport package will note specify the meshing procedures. For complicated mesh generation involving CAD models there are two distinct issues of concern. First one must consider the effect of meshing errors themselves. Although our new cut-cell methods provide second-order spatial discretization convergence, there is no guarantee that this convergence holds for the true geometry since the meshing procedure itself must be second-order. Furthermore we note that the mass preservation treatments employed in this work rely on exact knowledge of the underlying geometry although this information

may not be available in production mesh generation packages. Future work should examine popular mesh generation packages and confirm that they are adequate for the methods presented in this research.

REFERENCES

[1] J.D. Rogers, The Neutron's Discovery – 80 Years On, Physics Procedia 43 (2013) 1-9.

[2] P. Vaz, Neutron Transport Simulation (Selected Topics), Rad. Phys. And Chemistry 78 (2009) 829-842.

[3] K.M. Case, P.F. Zweifel, Linear Transport Theory, Addison-Wesley, Reading, MA, 1967.

[4] E.E. Lewis, W.F. Miller, Computational Methods of Neutron Transport, American Nuclear Society, La Grange Park, IL, 1993.

[5] B.G. Carlson, K.D. Lathrop, Transport Theory – The Method of Discrete Ordinates, J. Quantitative Spectroscopy & Radiative Transfer 145 (2014) 121-146.

[6] E. Boman, J. Tervo, M. Vauhkonen, Modeling the Transport of Ionizing Radiation Using the Finite Element Method, Phys. Med. Biol. 50 (2005) 265-280.

[7] E. Machorro, Discontinuous Galerkin Finite Element Method Applied to the 1-D Spherical Neutron Transport Equation, J. Comput. Phys. 223 (2007) 67-81.

[8] M. Mercimek, H.A. Ozgener, Discontinuous Finite Element Formulations for Neutron Transport in Spherical Geometry, Annals of Nuclear Energy 64 (2014) 244-255.

[9] J. Wood, M.M.R. Williams, Recent Progress in the Application of the Finite Element Method to the Neutron Transport Equation, Progress in Nuclear Energy 14 (1) (1984) 21-40.

[10] R.S. Baker, K.R. Koch, An S_N Algorithm for the Massively Parallel CM200 Computer, Nucl. Sci. and Eng. 128 (1998) 312.

[11] R. Sanchez, Assembly Homogenization Techniques for Core Calculations, Progress in Nuclear Energy 51 (2009) 14-31.

[12] F. Fraysse, J. de Vicente, E. Valero, The Estimation of Truncation Error by τ-Estimation Revisited, J. Comput. Phys. 231 (2012) 3457-3482.

[13] J.I. Duo, Y.Y. Azmy, L.T. Zikatanov, A Posteriori Error Estimator and AMR for Discrete Ordinates Nodal Transport Methods, Annals of Nuclear Energy 36 (2009) 268-273.

[14] J.W. Purvis, J.E. Burkhalter, Prediction of Critical Mach Number for Store Configurations, AIAA Journal 17 (11) (1979) 1170-1176.

[15] D.M. Ingram, D.M. Causon, C.G. Mingham, Developments in Cartesian Cut Cell Methods, Mathematics and Computers in Simulation 61 (2003) 561-572.

[16] M. Meinke, L. Schneiders, C. Günther, W. Schröder, A Cut-Cell Method for Sharp Moving Boundaries in Cartesian Grids, Computers & Fluids 85 (2013) 135-142.

[17] M.W. Johnson, A Novel Cartesian CFD Cut Cell Approach, Computers & Fluids 79 (2013) 105-119.

[18] K.J. Fidkowski, D.L. Darmofal, A Triangular Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations, J. Comput. Phys. 225 (2007) 1653-1672.

[19] F. Bouchon, T. Dubois, N. James, A Second-Order Cut-Cell Method for the Numerical Simulation of 2D Flows Past Obstacles, Computers & Fluids 65 (2012) 80-91.

[20] M.S. Reed, R.S. Baker, An Interface Reconstruction Technique for PARTISN, LANL Internal Memorandum LA-UR-07-7474, 20 November 2007.

[21] W.F. Walters, Use of the Chebyshev-Legendre Quadrature Set in Discrete Ordinates Codes, LANL Internal Memorandum LA-UR-87-3621, 2 February 1987.

[22] J.N. Reddy, An Introduction to the Finite Element Method, Second Edition, Tata McGraw-Hill, New Delhi, 2003.

APPENDIX A

CUT-CELL TYPE DESCRIPTIONS

The sweeping infrastructure used for solution of the S_N equations requires numerical integration in space and along boundary faces as discussed in section 2.5. These integrations as well as the upwinding procedure associated with the LDGFEM equations themselves require specific accounting of the cut-cell infrastructure. In this section we present graphical depictions of the various cut-cell types. We also introduce the vertex and face number schemes used to uniquely define them in code.

Ignoring cut-cells where the linear interface intersects a cell corner, there are two primary types of cut-cells. Cut-cells occurring when the linear cut intersects adjacent cell faces result in a combination of a quadrilateral and pentagonal element. The four distinct possibilities resulting from each of the four corners of the mixed-cell are given cell type numbers one through four. These cut-cell types are shown in Figure 41 below.

Cut-cell types formed by a horizontal interface that intersects opposite sides of the mixed-cell result in cut-cells comprised of two non-orthogonal quadrilateral elements. These cut-cells are given cell type numbers five through seven. The corresponding cut-cell types formed from a horizontal interface are given cell type numbers eight through ten. These cut-cell types are shown in Figure 42.

A numbering system was used in order to facilitate algorithmic procedures associated with sweeping and mapping inflows and outflows. Starting at the lower-left corner of the cut-cell, vertices are numbered sequentially in a counter-clockwise manner starting at 1. These numbers are indicated in red within the cut-cell type diagrams. Cell faces are numbered in the same way with face 1 being that face having outward unit normal directed purely left as indicated in the figures. These numbers are indicated in blue within the cut-cell type diagrams. The sub-element lying in the lower left corner is given the sub-cell number one, while the other sub-cell is given the designation two. These numbers appear in green within the cut-cell depictions. We conclude by noting that both face and vertex numbering is repeated for the vertices/faces shared along the linear interface as shown in the diagrams.

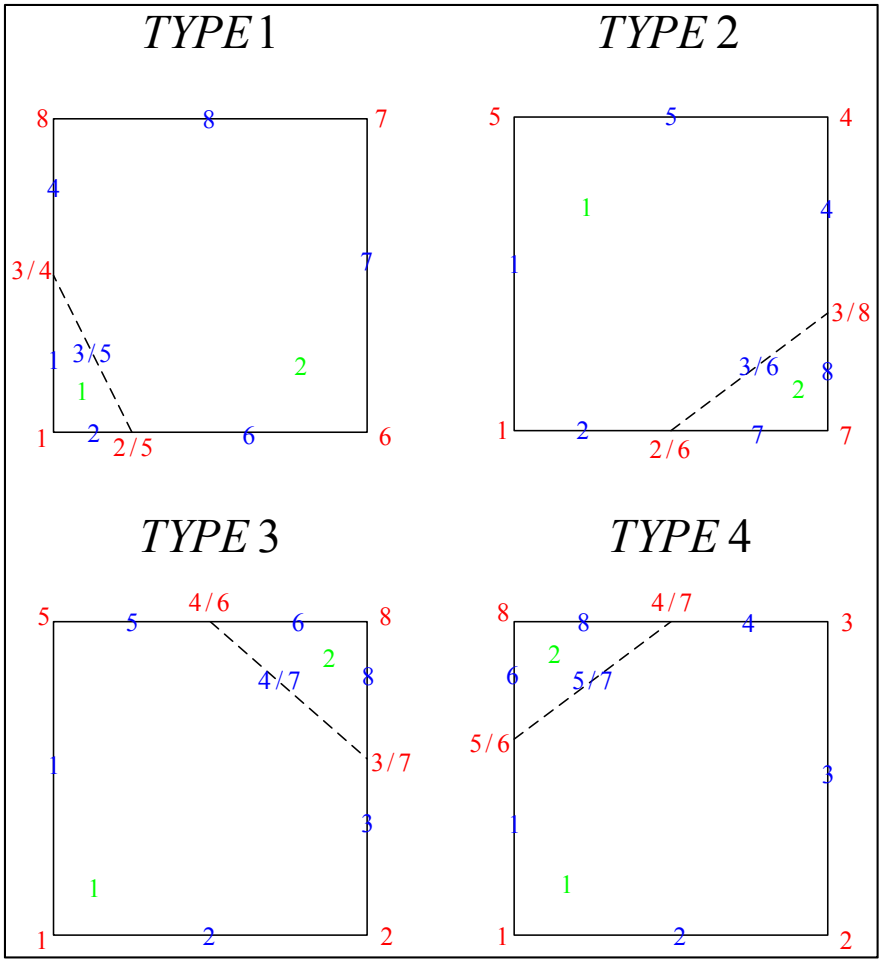

Figure 41. Visual depiction of quadrilateral/pentagon type cut-cells

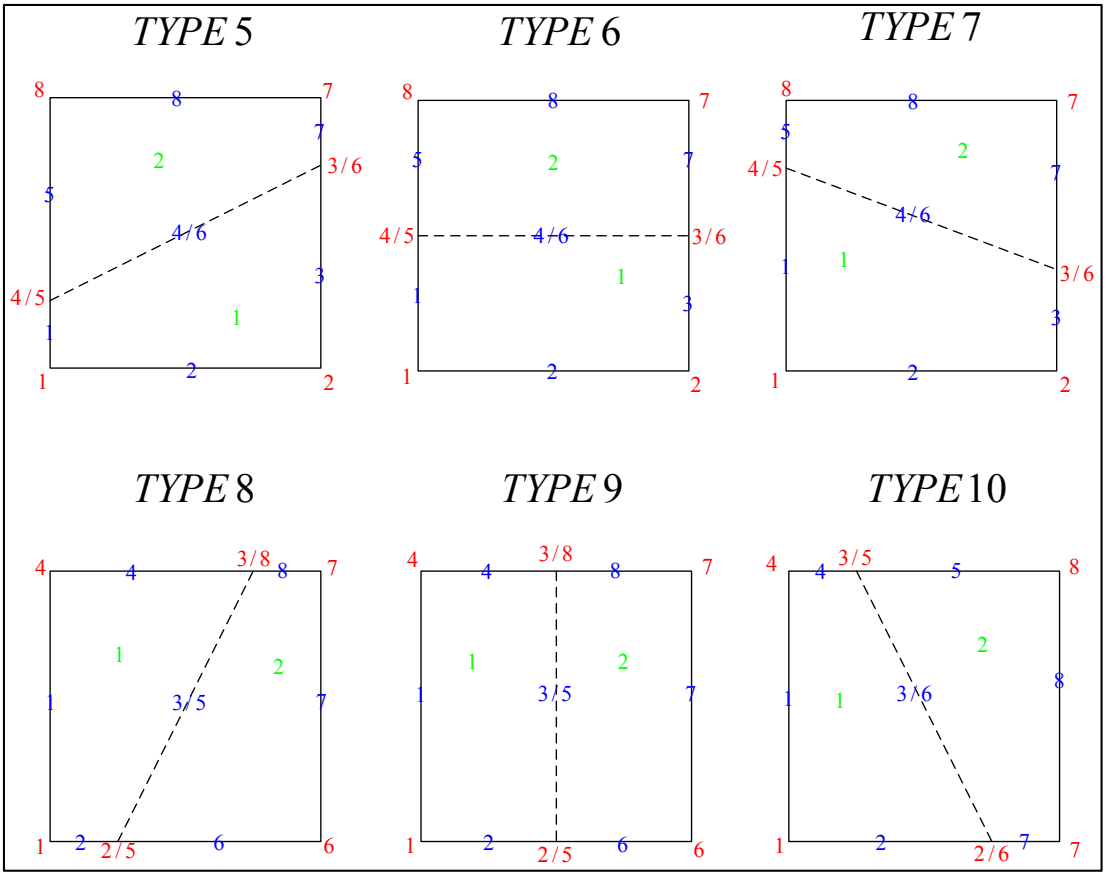

Figure 42. Visual depiction of quadrilateral/quadrilateral type cut-cells