

Dissertation

Discrete Ordinates Radiation Transport using High  
Order Finite Element Spatial Discretizations on  
Meshes with Curved Surfaces

Douglas N. Woods

June 4, 2018

AN ABSTRACT OF THE DISSERTATION OF

Douglas N. Woods for the degree of Doctor of Philosophy in Nuclear Engineering  
presented on June 4, 2018

Title: Discrete Ordinates Radiation Transport using High Order Finite Element Spatial Discretization on Meshes with Curved Surfaces

Abstract approved:

---

Todd S. Palmer

The high-order finite element  $S_N$  transport equations are solved on several test problems to investigate the behavior of the discretization method on meshes with curved edges in X-Y geometry. Simpler problems ensured the correct implementation of MFEM, the general finite element library employed. A convergence study using the method of manufactured solutions demonstrates the convergence rate as a function of number of unknowns for meshes whose sides are described by polynomial curves of increasing order. Optically thick and diffusive problems indicate the DGFEM transport solution trends toward a numerical solution of the diffusion equation, though a rigorous diffusion limit analysis has not yet been performed. A direct solve approach to computing the angular flux unknowns simultaneously is presented as an alternative to source iteration that involves linear algebraically inverting the “streaming plus collision minus scattering” operator. These results serve as a proof-of-concept of this spatial discretization method.

©Copyright by Douglas N. Woods

June 4, 2018

All Rights Reserved

Discrete Ordinates Radiaiton Transport using High Order Finite Element Spatial  
Discretization in *R-Z* Geometry on Meshes with Curved Surfaces

by  
Douglas N. Woods

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of  
the requirements for the  
degree of

Doctor of Philosophy

Presented June 4, 2018  
Commencement June 2018

Doctor of Philosophy dissertation of Douglas N. Woods presented on June 4, 2018.

APPROVED:

---

Major Professor, representing Nuclear Engineering

---

Head of the School of Nuclear Science and Engineering

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

---

Douglas N. Woods, Author

## ACKNOWLEDGEMENTS

I first want to thank my Lord and Savior for the opportunities and gifts I've been given. I want to thank my major advisor, Dr. Todd Palmer, for piquing my interest in transport discretizations during the advanced courses he taught. He found and presented the opportunity to work on this project, brought immense knowledge, and has been encouraging throughout this entire project. Thanks to Dr. Tom Brunner for serving as my mentor during a summer at Lawrence Livermore National Laboratory and giving me an inordinate amount of time and patience. I must thank my parents for being so encouraging, supporting, and loving during my life including. All of my family has been so patient with me – I too often neglected visiting them despite our relatively close proximity at OSU.

TABLE OF CONTENTS use “  
conheading” in .toc

	<u>Page</u>
1 Introduction	1
1.1 Radiation Transport	3
1.2 Discretization	4
1.2.1 Direction of Travel Discretization	5
1.2.2 Spatial Discretization	5
1.2.3 High Order Finite Elements	7
1.2.4 Meshes	8
1.3 Diffusion Limit	9
1.4 Source Iteration Acceleration	11
1.5 $R$ - $Z$ Geometry	12
1.6 Axisymmetry	14
1.7 Outline	15
2 High Order DFEM	16
2.1 Basis Functions	16
3 Meshes with Curved Surfaces	19
3.1 Transformation	19
4 Diffusion Synthetic Acceleration	22
4.1 Modified Interior Penalty DSA	22
4.1.1 Methodology	22
4.1.2 Fourier Analysis	22

## TABLE OF CONTENTS (Continued)

	<u>Page</u>
4.1.3 Results . . . . .	22
4.1.4 As a Preconditioner . . . . .	22
4.2 MIP DSA with Robin Boundary Conditions . . . . .	22
4.2.1 Zero Incident Current . . . . .	24
4.2.2 Fourier Analysis . . . . .	26
4.2.3 Results . . . . .	26
4.2.4 As a Preconditioner . . . . .	26
 5 <i>R-Z</i> Geometry	27
5.1 Angular Discretization . . . . .	28
5.2 Spatial Discretization . . . . .	32
5.3 Lumping . . . . .	35
5.4 Diffusion Synthetic Acceleration . . . . .	35
5.5 Symmetry Preservation . . . . .	36
5.5.1 Linear in Spherical Radius . . . . .	40
5.6 Other . . . . .	41
5.7 Material Discontinuity Stress Test . . . . .	41
5.8 Reflecting Boundary Conditions . . . . .	42
5.9 Reflecting Boundary Conditions . . . . .	42
 6 Conclusions	55
6.1 FutureWork . . . . .	55
 A Implementation in MFEM	62
A.1 Transport Operators . . . . .	63

A.1.1	MIP DSA Operators . . . . .	69
B	Mesh Examples	72
C	Mathematica Scripts	73
C.1	Mass Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals .	73
C.2	Angular Redistribution Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals . . . . .	75
C.3	r-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals	78
C.4	z-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals	81
C.5	z-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals	83
C.6	r Surface-Integral Matrix Entries for RZ BLD Gauss-Legendre on Quadri- laterals . . . . .	86
C.7	z Surface-Integral Matrix Entries for RZ BLD Gauss-Legendre on Quadri- laterals . . . . .	88
C.8	LinearForm Angular Redistribution Matrix Entries for RZ BLD Gauss- Legendre on Quadrilaterals . . . . .	90
C.9	LinearForm Scattering/Fixed Source Matrix Entries for RZ BLD Gauss- Legendre on Quadrilaterals . . . . .	93

## LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1    The positive octant for $S_6$ level-symmetric angular quadrature. . . . .		6
2    Example of mapping the reference element to a physical element. <b>Switch</b> these around to map the physical element to the reference element. . . . .		20
3    Cylindrical space-angle coordinate system showing the position $(r, z)$ and direction of travel $\Omega$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .		28
4    Angular discretization showing $(\xi, \mu)$ pairs; adapted from [1] . . . . .		29
5    Uniform infinite medium solution. . . . . . . . . . . . . . . . . . . . . . . .		34
6    MMS solution for Equation 54. . . . . . . . . . . . . . . . . . . . . . . . .		35
7    MMS solution for Equation 55. . . . . . . . . . . . . . . . . . . . . . . .		36
8    MMS solution to Equation 56. . . . . . . . . . . . . . . . . . . . . . . .		37
9 $L^2$ -norm of the errors from the manufactured solution and reference lines, where $N_{\text{unknowns}} = N_{\text{cells}}(p + 1)^2$ . . . . . . . . . . . . . . . . .		38
10   Relative asymmetry for 1 <sup>st</sup> -order finite elements on a 1 <sup>st</sup> -order mesh for given order of level-symmetric angular quadrature. . . . . . . . . . . . .		44
11   Relative asymmetry for 2 <sup>st</sup> -order finite elements on a 1 <sup>st</sup> -order mesh for given order of level-symmetric angular quadrature. . . . . . . . . . . . .		45
12   Relative asymmetry for 4 <sup>st</sup> -order finite elements on a 1 <sup>st</sup> -order mesh for given order of level-symmetric angular quadrature. . . . . . . . . . . . .		46
13   Relative asymmetry for 8 <sup>st</sup> -order finite elements on a 1 <sup>st</sup> -order mesh for given order of level-symmetric angular quadrature. . . . . . . . . . . . .		47
14   Relative asymmetry for $p = \{1, 2, 4\}$ finite elements on a 1 <sup>st</sup> -order mesh for $S_8$ level-symmetric angular quadrature. . . . . . . . . . . . . . . . . . .		48

15	Relative asymmetry for 1 <sup>st</sup> -order finite elements on a 2 <sup>nd</sup> -order mesh for given order of level-symmetric angular quadrature. . . . .	49
16	Relative asymmetry for 2 <sup>nd</sup> -order finite elements on a 2 <sup>nd</sup> -order mesh for given order of level-symmetric angular quadrature. . . . .	50
17	Relative asymmetry for 4 <sup>th</sup> -order finite elements on a 2 <sup>nd</sup> -order mesh for given order of level-symmetric angular quadrature. . . . .	51
18	Relative asymmetry for $p = \{1, 2, 4\}$ finite elements on a 2 <sup>nd</sup> -order mesh for $S_8$ level-symmetric angular quadrature. . . . .	52
19	Material discontinuity stress test with MIP DSA problem geometry; materials defined in Table 2. . . . .	53
20	Solution to multi-material stress test. White regions indicate negative scalar fluxes. <b>This was solved without DSA and was only allowed 10,000 source iterations to complete.</b> . . . . .	54
21	Flow diagram for solution process. . . . .	71

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Gaussian quadrature locations. . . . .	17
2	Material discontinuity stress test with MIP DSA material properties.	42
3	MFEM PDE function calls where the arguments have been dropped (see Equations ?? and ?? for these details). . . . .	66
4	MFEM PDE function calls where the arguments have been dropped (see Equations ?? and ?? for these details). . . . .	67
5	MFEM default integration orders for transport operators. The nota- tion for the finite element order is $p$ , mesh order is $m$ , and problem dimension is $d$ . . . . .	68
6	MFEM diffusion equation function calls. . . . .	70

# 1 Introduction

The radiation transport equation — nuclear reactors, medical industry, HEDP.

There are several applications for the thermal radiation transport (TRT) equation for nuclear fusion. Two of which include astrophysics such as stars or supernovae [2], and inertial confinement fusion such as the National Ignition Facility<sup>1</sup> (NIF). Nuclear fusion occurs in the high energy density physics (HEDP) regime where mass and energy densities of a material are very high [3, 2]. Since material temperatures are very high, materials emit black body radiation in tremendous quantities. This thermal radiation field deposits energy back to the material influencing the material internal energy, temperature, and density. The radiation field can exchange enough momentum with the material that it directly affects the fluid motion [2]. Radiation transport and material absorption and emission are interdependent mechanisms that must be modeled simultaneously to obtain an energy distribution throughout the material. The material may also be exposed to additional hydrodynamic forces that change pressures, introduce density fluctuations, and cause fluid motion. In turn, the motion and energy density of the fluid affects the location and quantity of radiation emission. This complicated system of thermal radiation transport with hydrodynamics is studied in radiation hydrodynamics. While all of the forces in these HEDP systems are important, we can study some of the effects independently. For instance, we study hydrodynamics separately from TRT.

The staggered grid hydrodynamics (SGH) approach is considered a traditional approach that uses the finite difference or finite volume methods, and has to compensate for calculating the spatial gradients, which are highly dependent on the mesh resolution [4]. The finite element method (FEM) has been employed, approximating the thermodynamic variables as piecewise constant and the kinematic variables as

---

<sup>1</sup><https://lasers.llnl.gov>

linear continuous[5], and higher order finite elements [6].

Lawrence Livermore National Laboratory (LLNL) is developing a hydrodynamics code called BLAST.<sup>2</sup> [4]. It solves the Euler equations using a general finite element method (FEM) on meshes with curved surfaces for the conservation of mass, energy, and momentum of a fluid [6]. Novel features include: higher order elements to represent the thermodynamic and kinematic variables, and meshes with curved surfaces. Compared to SGH, Dobrev et al. [6] demonstrated their method can more accurately model flow geometry, symmetry of radial flow, and an increased resolution of a shock front saying their method has the ability to model the shock within a single mesh element. These improvements have reduced some of the numerical errors that have been exhibited by previous methods.

BLAST utilizes MFEM [7], a general finite element library also being developed at LLNL, to spatially discretize using the high order FEM on meshes with curved surfaces. This proposed TRT research could be combined with a hydrodynamics code such as BLAST to develop a radiation hydrodynamics package to better model HEDP problems such as those produced during inertial confinement fusion at the NIF. The integrated radiation hydrodynamics code will need each physics package to numerically solve the same physical problem and communicate between packages. Hence, it is advantageous to utilize the same general finite element library when considering the integration of code packages.

This chapter introduces TRT and establishes our motivation for using the proposed methods. In Section ??, we give a brief introduction to the TRT equations and nomenclature. In Section 1.2, we describe various discretization methods used to solve the TRT equations. In Section 1.3, we describe the diffusion limit and it's application to optically thick problems found in HEDP regimes. Finally, in Section

---

<sup>2</sup><https://computation.llnl.gov/project/blast/>

1.7, we outline the remained of this research proposal.

## 1.1 Radiation Transport

The steady-state, mono-energetic radiation transport equation,

$$\Omega_m \cdot \nabla \psi(\mathbf{x}, \Omega_m) + \sigma_t(\mathbf{x})\psi(\mathbf{x}, \Omega_m) = \frac{1}{4\pi}\sigma_s(\mathbf{x}) \int_{4\pi} \psi(\mathbf{x}, \Omega') d\Omega' + \frac{1}{4\pi}S_0(\mathbf{x}), \quad (1)$$

describes the angular flux  $\psi(\mathbf{x}, \Omega)$ , which is a function of position ( $\mathbf{x}$ ) and direction of travel ( $\Omega$ ). We also define the scalar flux by

$$\phi(\mathbf{x}) = \int_{4\pi} \psi(\mathbf{x}, \Omega') d\Omega', \quad (2)$$

which accounts for the angular fluxes in all directions at spatial position  $\mathbf{x}$ . Both the angular and scalar fluxes describe a particle density in a phase space denoted by their dependent variables and have units  $\text{cm}^{-2}\text{s}^{-1}$ . In general, the total macroscopic cross section,  $\sigma_t(\mathbf{x}) = \varsigma_t(\mathbf{x})N(\mathbf{x})$ , is a property of the microscopic total cross section,  $\varsigma_t(\mathbf{x}) \text{ cm}^2$ , and the number density of the material,  $N(\mathbf{x}) \text{ cm}^{-3}$ . The absorption and scattering cross sections are related to the total cross section by  $\sigma_t(\mathbf{x}) = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x})$ . Additionally, the isotropic volumetric source,  $S_0(\mathbf{x})/4\pi$  is some isotropically emitting external source (e.g., fission neutrons or blackbody radiation).

There are only a few specific cases where the radiation transport equation can be solved exactly. Problems of interest typically fall outside of this subset and discretization methods must be employed to obtain approximate solutions. There are two general categories of methods used to solve this equation: stochastic and deterministic methods.

Stochastic methods take a statistical approach. In any particular source region, a single particle is emitted and tracked through its “random walk” through the problem domain until it is either absorbed or escapes the domain. One can simply tally the

number of particles that get absorbed into a region of interest. If enough particles are simulated, a statistically significant conclusion can be drawn about the particle density in a region of interest. This is very accurate but also computationally demanding because of the large number of particles required to be simulated. This method does not solve any equation, but rather is based on the physics of the particle-material interactions.

Alternatively, deterministic methods solve the radiation transport equation for the scalar flux by discretizing the equation in each of the dependent variables (i.e., space and direction of travel). The discretized equation results in a linear system of equations that can be solved by linear algebra techniques. Deterministic methods are generally faster than stochastic methods but involve discretization or truncation errors. However, employing certain discretizations can reduce the impact of these errors.

## 1.2 Discretization

Physically, 3-dimensional space is continuous: a particle could exist at any one of an infinite number of locations. Numerically, it is impossible to compute a solution at an infinite number of locations so we discretize the spatial domain into a small subset of locations. Similarly, a particle can travel in an infinite number of directions so we approximate it traveling in only a few. Discretizing the direction of travel by using the discrete ordinates ( $S_N$ ) method [8] is common. Particles with various energies may behave or interact with material differently. It is common to approximate an infinite number of particle energies by grouping them into a finite number of energy groups. If a problem evolves through time, we discretize the time continuum with small discrete steps through time. Ideally, we would model the continuous phase space because it describes the physical behavior of neutral particles. Since we can-

not account for an infinite number of points on a continuum, we must make these discrete approximations. In this light, we discretize each of the continuous variables to approximate each as closely as practical. In general, this can mean refining the discretization, e.g., refining the spatial mesh or using a higher order discrete ordinates approximation. However, the cost of higher accuracy is increased computational time, which may become prohibitively expensive.

This research discretizes the energy domain into one group, thereby making the problem energy independent. We also assume the solution of the transport equation has reached steady-state, making the problem time independent. We discretize the two remaining dependent variables (space and direction of travel) where the spatial discretization is the main focus of this research.

### 1.2.1 Direction of Travel Discretization

We approximate the continuum of directions that a particle can travel by assuming they only travel in discrete ordinate directions. Figure 1 shows an example of a set of discrete ordinates.

We utilize the discrete ordinate weights to approximate the integral in Equation 2 by

$$\phi \approx \sum_m w_m \psi_m. \quad (3)$$

### 1.2.2 Spatial Discretization

There are several methods of spatial discretization that appear in the transport community. Among the most common are characteristic methods [9], finite difference methods [1], finite volume methods, and the finite element methods (FEM) [1]. Typically, the problem domain is divided to a larger number of smaller domains. The equations are then numerically solved on each of these smaller regions where the solution is likely less varying when compared to the entire problem. The FEM was

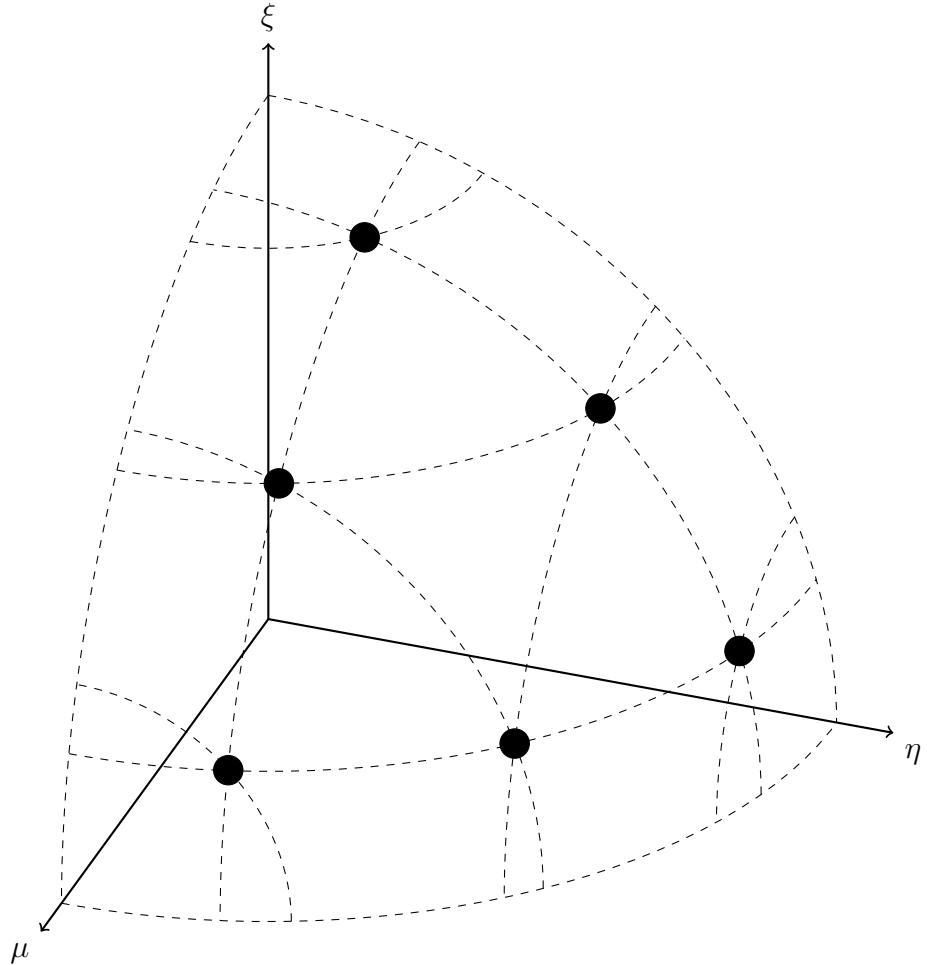


Figure 1: The positive octant for  $S_6$  level-symmetric angular quadrature.

introduced to the transport community in the 1970's. The utility of this method was that it was more accurate than finite differencing methods [10]. Since then, radiation transport research using the FEM has proliferated. In particular, the discontinuous FEM (DFEM) has been favored for its ability to perform in the thick diffusion limit [11] (discussed in Sections 1.3 and ??). Thus, this research uses the DFEM in which the solution is approximated to have a functional form within each mesh element. More discussion of various discretization methods can be found in Lewis and Miller [1].

Additionally, these radiation transport problems are solved in each of Cartesian,

cylindrical ( $R$ - $Z$ ), and spherical coordinates. The present research is concerned with Cartesian and  $R$ - $Z$  geometries. Difficulties arise in  $R$ - $Z$  geometry because of the introduction of angular derivatives. While a particle travels in a straight line in direction  $\Omega$ , the cosines of the angles relative to the coordinate axes change as the  $r$  and  $z$  coordinates change.

### 1.2.3 High Order Finite Elements

The finite element spatial discretization method (FEM) for the radiation transport equation has been continually investigated since the 1970s [10, 12]. Although FEMs require more computer memory than finite-difference methods, the linear discontinuous finite element method (LD) has been used because of its increased accuracy [11] and continues to be used and researched [13, 14, 15]. Studying and implementing more accurate methods becomes increasingly possible as computational hardware performance increases. Computers can process and communicate quicker, and larger memory can store more variables. The FEM has since been used and shown to be accurate in production codes like Attila [16], a three-dimensional transport code. Trilinear DFEM (TLD), the three-dimensional analog to bilinear DFEM (BLD), was later adapted in Attila [17]. Other methods have been studied as well, such as the piece-wise linear DFEM (PWLD) [18].

Higher order DFEMs (polynomial orders  $p \geq 2$ ) have become more popular as modern computer performance improves. Several authors [19, 20, 21, 22] saw increased accuracy for up to 4<sup>th</sup>-order finite elements. Woods et al. [23] saw increased accuracy for up to 8<sup>th</sup>-order finite elements.

Negative solutions can arise from oscillations of higher order finite element solutions near zero [24, 23]. Authors have addressed negative scalar fluxes [14, 15, 25, 26, 27], with methods generally falling into two categories: ad hoc modification af-

ter calculating a solution, and modifying the equations beforehand to yield positive results.

#### 1.2.4 Meshes

There are a wide variety of meshes used with triangular and quadrilateral being the most common. Modeling curved boundaries with quadrilateral spatial meshes can be difficult without significant mesh refinement. Triangular meshes can be used in these cases and are often seen in literature [10, 20, 21, 28]. For example, in three-dimensions, Attila can discretize using unstructured tetrahedral meshes [16]. The PWLD mesh refinement truncation error was first thought to behave as well on unstructured meshes as other methods using structured grids [29].

Recently, meshes with curved surfaces have been investigated using finite elements for radiation transport [23, 24]. The motivation is to model curved boundaries within a problem without requiring additional mesh refinement. Woods et al. [23] performed a spatial convergence study using, meshes with curved surfaces and saw convergence rates of  $(p + 1)$ , which is expected from Lasaint and Raviart [12] and Wang and Ragusa [20].

Other fields have investigated the use of meshes with curved surfaces. Cheng [30] demonstrated that straight-edged meshes, compared to curved meshes, restricted the accuracy of the solution of the compressible Euler equations, alluding to the necessity of curved meshes for higher-order accuracy. Dobrev et al. [4] saw increased accuracy using such meshes for Lagrangian hydrodynamics. They mapped the reference element to the physical element using the basis functions, which results in curved surfaces for  $p > 1$  elements. Subsequently, Dobrev et al. [31] demonstrated curved mesh use in axisymmetric geometries and noted that it more accurately modeled some features of the flow, among other improvements.

### 1.3 Diffusion Limit

In high energy density physics (HEDP) problems, a particle can have a very small mean free path (mfp) compared to the size of a spatial zone. The mfp is the inverse of the total cross section  $\Lambda = \sigma_t^{-1}$ . Therefore, when the total macroscopic cross section  $\sigma_t$  becomes very large, a span of length can be described as being many mfp's long. These problems are called “optically thick”. The transport equation (Eq. 1) resolves optically thin length scales and we require the transport equation to resolve lengths in the optically thick regime. We can assess the behavior of Equation 1 as the problem becomes increasingly optically thick by performing an asymptotic analysis. Specifically, if a small factor  $\varepsilon$  is used to scale the physical processes of Equation 1,

$$\boldsymbol{\Omega}_m \cdot \nabla \psi_m + \frac{\sigma_t}{\varepsilon} \psi_m = \frac{1}{4\pi} \left( \frac{\sigma_t}{\varepsilon} - \varepsilon \sigma_a \right) \int_{4\pi} \psi d\Omega' + \varepsilon S_0 \quad (4)$$

where arguments have been dropped for brevity. Then, as  $\varepsilon \rightarrow 0$  the mean free path  $\Lambda = \varepsilon/\sigma_t \rightarrow 0$ . In this limit, the problem is said to be optically thick and diffusive. It can be shown [32, 33] that this scaled analytic transport equation limits to the analytic radiation diffusion equation to  $O(\varepsilon^2)$ . The radiation diffusion equation provides accurate solutions to optically thick and diffusive problems in the problem interior. Hence, concluding that the analytic transport equation limits to the analytic diffusion equation is physically meaningful. Optically thick and diffusive problems that may be typically solved with the radiation diffusion equation can also be solved using the radiation transport equation.

The diffusion equation has substantially fewer degrees of freedom making it much quicker to solve numerically. This benefit is accompanied by some drawbacks. The diffusion equation cannot resolve optically thin problems nor solutions near highly absorbing regions, voids, nor strong material discontinuities [34]. However, the transport equation can resolve solutions in all of these regimes, including the diffusive

regions. Hence, our present work employs the transport equation to solve problems that are optically thick and diffusive.

The diffusion limit analysis has been applied to spatially discretized transport problems. The LD and lumped LD (LLD) methods were shown to have the diffusion limit in one-dimension by Larsen and Morel [11] and Larsen [35], respectively, which is one of the reasons for their continued popularity in literature. In multiple dimensions, the LD method generally fails in the diffusion limit [36], but the LLD [28, 37], BLD [15], and fully lumped BLD (FLBLD) [38] methods possess the diffusion limit. Lumping the BLD equations in curvilinear geometry has been demonstrated to possess the diffusion limit as well [39, 40]. Attila has been demonstrated to solve problems in the diffusion limit using tri-linear DFEM, despite having the potential for negative fluxes [17].

The PWLD has also been popular for its performance in the diffusion limit. Stone and Adams [29] concluded the PWLD should behave as well as BLD but on unstructured meshes. Bailey et al. [18, 41] corroborated this and extended the PWLD method to RZ geometry and concluded this method has the diffusion limit. Bailey et al. [42] then introduced piece-wise BLD (PWBLD), which allows for more curvature in a solution and has properties favorable to having the diffusion limit, although they do not perform the asymptotic analysis to be conclusive.

Woods et al. [23] numerically demonstrate that two-dimensional Cartesian high order finite elements trend toward the diffusion limit despite not being conclusive about possessing the diffusion limit. Their study was performed without a source iteration acceleration technique so they were unable to perform calculations for optically thick media. This was performed on a quadrilateral mesh with curved surfaces. Their results indicated their method may converge toward the diffusion limit at  $O(\varepsilon)$  rather than  $O(\varepsilon^2)$ , which was suggested may occur due to the inclusion of curved

surfaces [15].

## 1.4 Source Iteration Acceleration

The source iteration (SI) method [1] is commonly employed to solve the discretized transport equation. The algorithm

$$\Omega \cdot \nabla \psi^{(\ell+1/2)} + \sigma_t \psi^{(\ell+1/2)} = \frac{1}{4\pi} \sigma_s \phi^{(\ell)} + S_0 \quad (5a)$$

$$\phi^{(\ell+1/2)} = \sum_m w_m \psi_m^{(\ell+1/2)} \quad (5b)$$

$$\phi^{(\ell+1)} = \phi^{(\ell+1/2)} \quad (5c)$$

describes the calculation of the angular flux using the lagged scalar flux followed by an update to the scalar flux using Equation 3.

The transport equation can converge arbitrarily slowly in highly scattering problems [43]. If computational efficiency is important, one option would be to solve the problem using the diffusion equation instead of the transport equation. The radiation diffusion equation can resolve these problems well and it does not require the source iteration method. However, the diffusion equation cannot resolve the solution in highly absorbing regions nor at material discontinuities including vacuum boundaries [34].

We want to preserve the transport solution to these problems so we opt to speed up the SI. One option is to refine the mesh until the optical thickness of a typical mesh cell is on the order of a mean-free-path to, effectively, solve an optically thin problem in each mesh zone. This option is not very efficient because it can introduce a large number of degrees of freedom to the problem, thereby increasing the solution time. Alternatively, acceleration techniques can be applied to the SI to compensate for slow convergence. Diffusion synthetic acceleration (DSA) is commonly used to

accelerate the source iteration and is discussed in more detail below.

Several source iteration acceleration schemes have been developed. The Wareing-Larsen-Adams (WLA) method [44], accelerates the source iterations but its effectiveness was found to degrade as cells become optically thick [45]. Another alternative, the modified four-step (M4S) [46], is effective in 1-D but is conditionally stable in 2-D for unstructured meshes. Diffusion synthetic acceleration (DSA) is a very common method. Using DSA, Larsen [43] and Larsen and McCoy [47] showed that one-dimensional LD is unconditionally accelerated. Wareing et al. [44] and Adams and Martin [48] extended unconditionally accelerating DSA to two-dimensions.

Recently, Wang and Ragusa [19] developed the modified interior penalty (MIP) form for the DSA equations. Considered a partially consistent scheme, it was developed from the interior penalty form of a discretized diffusion equation and was originally applied to high-order DFEM  $S_N$  transport on triangular meshes. It has since been used with PWLD on arbitrary polygon meshes [49] and later with BLD [50]. The MIP DSA equations were derived with Dirichlet boundary conditions according to Kanschat [51] for incompressible flow.

The DSA implementation originally utilized Robin (Marshak, zero incident current, vacuum) boundary conditions [48, 45], whereas the MIP DSA equations, as developed, employed homogeneous Dirichlet boundary conditions. It has been acknowledged that homogeneous Dirichlet boundary conditions may degrade performance [22].

## 1.5 *R-Z Geometry*

The radiation transport equation has been solved in *R-Z* geometry for many years. One particular difficulty with cylindrical coordinates is that an angular derivative is present within the streaming term. Hence, the spatial discretization is inherently

dependent upon the direction of travel discretization. Despite a particle traveling in a straight line, the coordinate system describing that direction changes with position. The common method, described in Section ??, is to perform an angular differencing that requires a “starting direction” equation, closing the system of equations with a weighted diamond difference scheme.

Palmer and Adams [39] and Palmer [52] solved the  $R$ - $Z$   $S_N$  equations using BLD, MLBLD, SLBLD, FLBLD, and SCB methods but found that only the FLBLD and SCB methods are accurate in the thick diffusion limit, as predicted by their asymptotic diffusion limit analysis. They determined that the support points must be sufficiently “local” to achieve a reasonable discretization of the diffusion equation.

Bailey et al. [18] derived the PWLD transport equation for  $R$ - $Z$  geometry on arbitrary polygonal meshes and showed that their method is accurate in the diffusion limit. Bailey [41] performed an asymptotic diffusion limit analysis for PWLD in  $R$ - $Z$  geometry on an arbitrary polygonal mesh and found the leading order angular flux is isotropic. The PWLD scheme also displayed  $O(h^2)$  convergence rates in several test problems, as expected.

Morel et al. [28] performed an asymptotic diffusion limit analysis for the LLD method in  $R$ - $Z$  geometry on a triangular mesh. The LLD equations satisfy a LLC diffusion discretization to leading order on the mesh interior. They also numerically demonstrate a  $O(h^2)$  spatial convergence in the thick diffusion limit.

Efforts to maintain positivity on non-orthogonal meshes in  $R$ - $Z$  geometry have been successful. Morel et al. [40] derived a lumped BLD scheme for quadrilaterals that “is conservative, preserves the constant solution, preserves the thick diffusion limit, behaves well with unresolved boundary layers, and gives second-order accuracy in both the transport and thick diffusion-limit regimes.”

At this time, no application of high order finite elements in  $R$ - $Z$  geometry on

meshes with curved surfaces has been observed.

## 1.6 Axisymmetry

It can be convenient to solve a spherical problem using *R-Z* geometry. Specifically because the streaming term of the radiation transport equation is more complicated in spherical coordinates than in cylindrical. Thus, the discretization methodology need not be as complicated. Despite the convenience of describing a physical position within a sphere using spherical coordinates, the spatial discretizaiton methodology becomes much more complicated. The increased complexity is analogous to the increased complexity of using *R-Z* geometry from Cartesian coordinates.

It can be simpler to solve a spherical problem using *R-Z* geometry. The question of whether or not this can be modeled accurately naturally arises. Two ways of considering how well the *R-Z* discretized system can model a spherical system are using the method of manufactured solutions (MMS) [53] with a manufactured solution in spherical geometry, and considering the spherical symmetry preservation from an *R-Z* geometry discretization. We consider both methods in this dissertation.

There has been relatively little research in this vein. In hydrodynamics, Dobrev et al. [31] demonstrated this spherical symmetry preservation. In radiation diffusion, Brunner et al. [54] demonstrated conditional spherical symmetry preservation. To date, we are only aware of one published paper discussing the ability of the *R-Z* transport equation to preserve 1-D spherical symmetry. Chaland and Samba [55] qualitatively demonstrated spherical symmetry preservation in a void with an initial condition of scalar flux in the center of the “sphere”. They demonstrated that ray-effects are created from the  $S_N$  transport equations, whereas they present an alternative angular discretization that qualitatively preserves the 1-D spherical symmetry in a void.

## 1.7 Outline

The remainder of this dissertation is outlined as follows. We discuss the finite element method in Section 2, including high-order basis functions. In Section 3, we discuss the meshes with curved surfaces and their transformations. Section 4 discusses the DSA method, our modifications, and results we obtained. Finally, Section 5 discusses the *R-Z* discretization scheme, solution methodology, and results using our methodology.

## 2 High Order DFEM

This section describes the methods to be carried out to accomplish the proposed research. We also consider some potential issues that we may encounter with some initial thoughts about mitigation.

### 2.1 Basis Functions

The basis functions are mesh zone dependent. So, by transforming each physical element into the reference element, the basis functions are uniform for each physical element. The basis functions are allowed to be unique to each element. However, in practice, each mesh element is transformed to the reference element, removing the mesh dependence for the basis functions.

One requirement for the basis functions is that they are unity at the integration point they “live” at and zero at all of the others. In 2-D, the general form of the first-order polynomial basis function is

$$b(x, y) = axy + bx + cy + d, \quad (6)$$

and the second-order basis function,

$$b(x, y) = ax^2y^2 + bx^2y + cx^2 + dxy^2 + exy + fx + gy^2 + hy + j. \quad (7)$$

The location of the integration points is important. We may place these evenly across the reference element or at Gaussian quadrature locations. On the unit square  $[0, 1] \times [0, 1]$ , the Gaussian integration points are located at the cross product of the nodes on  $[0, 1]$ . Listed for linear, quadratic, and cubic finite elements in Table 1, the Gaussian integration point locations were transformed from the traditional  $[-1, 1]$  to our reference element length  $[0, 1]$ .

Once the finite element order and the set of integration points is determined,

Table 1: Gaussian quadrature locations.

FE order	Points on $[-1, 1]$	Points on $[0, 1]$
1	$\pm \frac{1}{\sqrt{3}}$	$\frac{1}{3 + \sqrt{3}}$ $1 - \frac{1}{3 + \sqrt{3}}$
2	0 $\pm \sqrt{\frac{3}{5}}$	$\frac{1}{5 + \sqrt{15}}$ 0.5 $1 - \frac{1}{5 + \sqrt{15}}$
3	$\pm \sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}$ $\pm \sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{10 + \sqrt{30}}{35 + \sqrt{35(15 - 2\sqrt{30})}}$ $\frac{10 - \sqrt{30}}{35 + \sqrt{35(15 - 2\sqrt{30})}}$ $1 - \frac{10 + \sqrt{30}}{35 + \sqrt{35(15 - 2\sqrt{30})}}$ $1 - \frac{10 - \sqrt{30}}{35 + \sqrt{35(15 - 2\sqrt{30})}}$

the linear system can be arranged to determine the coefficients. For example, to determine the first-order basis function that “lives” at integration point  $(x_1, y_1)$  (and hence is equal to unity at that integration point and is zero at the remaining three), we assemble the system of equations

$$\begin{bmatrix} x_1 y_1 & x_1 & y_1 & 1 \\ x_2 y_2 & x_2 & y_2 & 1 \\ x_3 y_3 & x_3 & y_3 & 1 \\ x_4 y_4 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (8)$$

Moving the 1 inside the solution vector on the right-hand-side determines which basis

function coefficients will be obtained. The vector  $[0, 0, 1, 0]^T$  will return the basis function for the third integration point at  $(x_3, y_3)$ .

## 3 Meshes with Curved Surfaces

The complicated shapes of each mesh zone create a challenge by having to solve the discretized equations for each unique mesh zone. We can avoid having to solve a unique set of equations for each mesh zone by transforming the mesh zone into the reference element. Each mesh zone will have a unique transformation but an identical set of basis functions to obtain the solution on the reference element.

These meshes add complications to the numerical methods used to solve the transport equation. In particular, cycles can be present during the spatial sweep of the mesh. It is common to solve for a single mesh zone using incident angular flux information and propagate that angular flux from mesh zone to mesh zone, sweeping through the grid. However, if any particular mesh zone has both incident and outgoing angular fluxes to another mesh zone, they depend upon each other in a cyclic manner. “Breaking the cycle” in some fashion is necessary to perform the numerical computation. Alternatively, instead of sweeping through the grid, we utilize MFEM to generate the solution matrix for all of the mesh zones for the entire problem. This is computationally intensive, but it bypasses the need to consider any cycles that may occur.

### 3.1 Transformation

We set up the system of equations (Section ??) on each individual mesh zone after we transform it to the reference element. After performing the following integrations, we map the solution back to the physical element. The bi-quadratic mapping from the reference element to the physical element, shown in Figure 2, has the following

functional form

$$\begin{bmatrix} x(\rho, \kappa) \\ y(\rho, \kappa) \end{bmatrix} = \sum_{i=1}^{J_k} \sum_{j=1}^{J_k} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} N_i(\rho) N_j(\kappa) \quad (9)$$

where

$$N_l(\xi) = \begin{cases} (2\xi - 1)(\xi - 1), & l = 1 \\ 4\xi(1 - \xi), & l = 2 \\ \xi(2\xi - 1), & l = 3 \end{cases} \quad (10)$$

are the quadratic basis functions that have support points at typical locations shown in the left image of Figure 2. The  $(x_{ij}, y_{ij})$  coordinates are the locations of the support points in the physical element and are generally known. For example, the node  $(x_{12}, y_{12})$  is the location on the physical zone that is mapped from  $(\rho, \kappa) = (0, 0.5)$  on the reference element.

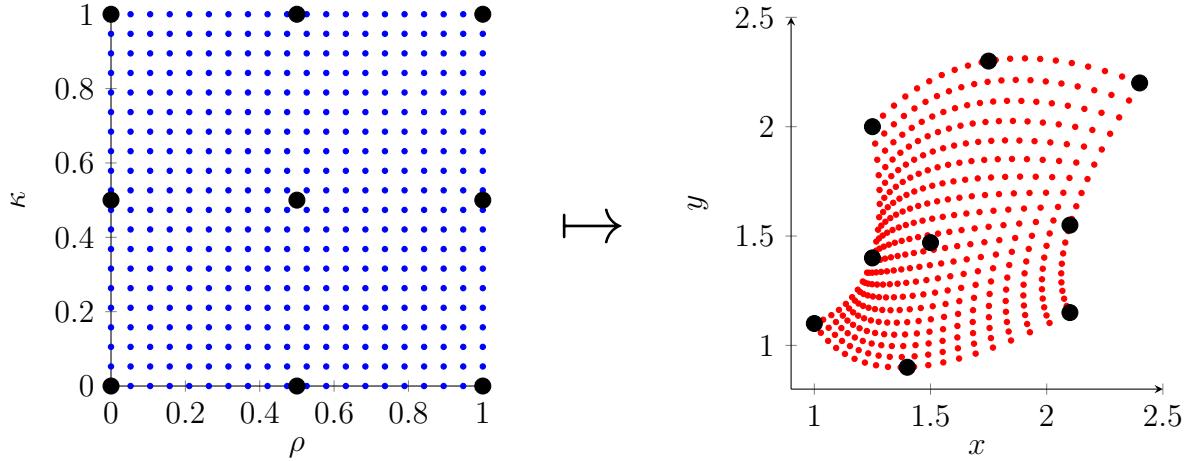


Figure 2: Example of mapping the reference element to a physical element. **Switch these around to map the physical element to the reference element.**

The determinant of the Jacobian of the transformation,

$$\det(J) = \begin{vmatrix} \frac{\partial x}{\partial \rho} & \frac{\partial y}{\partial \rho} \\ \frac{\partial x}{\partial \kappa} & \frac{\partial y}{\partial \kappa} \end{vmatrix}, \quad (11)$$

is required for the volume integrations in the next section.

## 4 Diffusion Synthetic Acceleration

This section describes the methods to be carried out to accomplish the proposed research. We also consider some potential issues that we may encounter with some initial thoughts about mitigation.

### 4.1 Modified Interior Penalty DSA

#### 4.1.1 Methodology

#### 4.1.2 Fourier Analysis

#### 4.1.3 Results

#### 4.1.4 As a Preconditioner

### 4.2 MIP DSA with Robin Boundary Conditions

The diffusion synthetic acceleration (DSA) algorithm first solves the RTE, called a “transport sweep”, followed by a diffusion solve for each source iteration. The RTE solve is performed exactly as before (Eqs. 5) with notation changing to indicate a “half step”:

$$\Omega \cdot \nabla I^{(l+1/2)} + \sigma_t I^{(l+1/2)} = \sigma_s E^{(l)} + S_0 \quad (12a)$$

$$E^{(l+1/2)} = \sum_{m=1}^M w_m I_m^{(l+1/2)}(\mathbf{r}, \Omega) \quad (12b)$$

(Recall the energy density  $E$  defined by Equation ??.) Next, we solve the diffusion equation with a modified source term utilizing the half step and previous iteration RTE solutions.

$$-\nabla \cdot D \nabla \phi^{(l+1/2)} + \sigma_a \phi^{(l+1/2)} = \sigma_s (E^{(l+1/2)} - E^{(l)}) \quad (12c)$$

The diffusion solution,  $\phi^{(l+1/2)}$ , becomes the correction factor at the half step.

This correction factor is added to the energy density at the half step to complete the full DSA iteration.

$$E^{(l+1)} = E^{(l+1/2)} + \phi^{(l+1/2)} \quad (12d)$$

This iterative process continues until the convergence criteria

$$\|E^{(l+1)} - E^{(l)}\|_{\infty} < \varepsilon_{\text{conv}} (1 - \rho) \|E^{l+1}\|_{\infty}, \quad (13)$$

is met, where  $\varepsilon_{\text{conv}}$  is a small number.

Wang and Ragusa [19] proposed the modified interior penalty (MIP) equations for discretizing the DSA equation using the DGFEM:

$$\begin{aligned} b_{MIP}(\phi, w) = & (\sigma_a \phi, w)_{\mathbb{V}} + (D \nabla \phi, \nabla w)_{\mathbb{V}} \\ & + (\kappa_e [\![\phi]\!], [\![w]\!])_{\partial \mathbb{V}^i} + ([\![\phi]\!], \{D \partial_n w\})_{\partial \mathbb{V}^i} + (\{D \partial_n \phi\}, [\![w]\!])_{\partial \mathbb{V}^i} \\ & + (\kappa_e \phi, w)_{\partial \mathbb{V}^d} - \frac{1}{2} (\phi, D \partial_n w)_{\partial \mathbb{V}^d} - \frac{1}{2} (D \partial_n \phi, w)_{\partial \mathbb{V}^d} \end{aligned} \quad (14)$$

and

$$l_{MIP}(w) = (Q_0, w)_{\mathbb{V}} \quad (15)$$

where  $b_{MIP}$  is the bilinear form,  $l_{MIP}$  is the linear form,  $w$  is the weight/test function,  $\mathbb{V}$  denotes the element volume,  $\partial \mathbb{V}^i$  is the internal edges,  $\partial \mathbb{V}^d$  is on the problem boundary,  $\partial_n$  is the partial derivative perpendicular to the edge (i.e.  $\nabla \cdot \hat{n}$ ). The following definitions accompany Equations 14 and 15.

$$[\![\phi]\!] = \phi^+ - \phi^- \quad (16)$$

$$\{\!\!\{\phi\}\!\!\} = \frac{(\phi^+ + \phi^-)}{2} \quad (17)$$

$$Q_0 = \sigma_s (E^{(l+1/2)} - E^{(l)}) \quad (18)$$

$$\kappa_e^{IP} = \begin{cases} \frac{c(p^+)}{2} \frac{D^+}{h_\perp^+} + \frac{c(p^-)}{2} \frac{D^-}{h_\perp^-}, & \text{on interior edges (i.e., } \mathbf{r} \in E_h^i) \\ c(p) \frac{D}{h_\perp}, & \text{on boundary edges (i.e., } \mathbf{r} \in \partial D^d) \end{cases} \quad (19)$$

$$\kappa_e = \max \left( \kappa_e^{IP}, \frac{1}{4} \right) \quad (20)$$

$$c(p) = Cp(p+1) \quad (21)$$

where  $C$  is a an arbitrary constant,  $D^\pm$  is the diffusion coefficient,  $\phi^\pm$  is the scalar flux,  $p^\pm$  is the order of finite elements, and  $h_\perp^\pm$  is the perpendicular length of the cell, where the  $\pm$  denotes either side of an element boundary. Equation 20 is a “switch” between two methods (interior penalty and diffusion conforming form), one of which is stable for optically thin regions and the other for optically thick regions. Wang and Ragusa [19] used  $C = 2$  and Turcksin and Ragusa [49] used  $C = 4$ .

#### 4.2.1 Zero Incident Current

Kanschat [51] shows that Equations 14 and 15 employ Nitsche’s method for “a fully conforming method of treating Dirichlet boundary values.” The boundary terms  $(\partial \mathbb{V}^d)$  in this form are homogeneous Dirichlet boundary conditions. The result is that the DSA correction for the energy density at the problem boundaries is zero, so the energy density is only updated by the RTE solution. That is, the DSA correction only accelerates the interior solution. Consequently, the energy densities on the problem boundary is only subjected to the RTE solution source iterations.

Instead, a DSA update equation should incorporate Robin boundary conditions

(zero incident partial current) on the boundaries,

$$\mathbf{J}_- = 0 = \frac{1}{4}\phi + \frac{1}{2}D\boldsymbol{\nabla}\phi \cdot \hat{n}, \quad (22)$$

$$-\frac{1}{2}\phi = D\boldsymbol{\nabla}\phi \cdot \hat{n}, \quad (23)$$

thereby allowing a correction of the boundary energy densities. This boundary condition requires modification of Equation 14. Three implementaion methods are proposed here. Method 1 substitutes Equation 23 into Equation 14:

$$\begin{aligned} b_{MIP,1}(\phi, w) &= (\sigma_a\phi, w)_{\mathbb{V}} + (D\boldsymbol{\nabla}\phi, \boldsymbol{\nabla}w)_{\mathbb{V}} \\ &+ (\kappa_e[\![\phi]\!], [\![w]\!])_{\partial\mathbb{V}^i} + ([\![\phi]\!], \{D\partial_n w\})_{\partial\mathbb{V}^i} + (\{D\partial_n \phi\}, [\![w]\!])_{\partial\mathbb{V}^i} \\ &+ (\kappa_e\phi, w)_{\partial\mathbb{V}^d} - \frac{1}{2}(\phi, D\partial_n w)_{\partial\mathbb{V}^d} + \frac{1}{4}(\phi, w)_{\partial\mathbb{V}^d} \end{aligned} \quad (24)$$

Method 2 is very similar to Method 1 with one term removed:

$$\begin{aligned} b_{MIP,2}(\phi, w) &= (\sigma_a\phi, w)_{\mathbb{V}} + (D\boldsymbol{\nabla}\phi, \boldsymbol{\nabla}w)_{\mathbb{V}} \\ &+ (\kappa_e[\![\phi]\!], [\![w]\!])_{\partial\mathbb{V}^i} + ([\![\phi]\!], \{D\partial_n w\})_{\partial\mathbb{V}^i} + (\{D\partial_n \phi\}, [\![w]\!])_{\partial\mathbb{V}^i} \\ &+ (\kappa_e\phi, w)_{\partial\mathbb{V}^d} + \frac{1}{4}(\phi, w)_{\partial\mathbb{V}^d} \end{aligned} \quad (25)$$

Method 3 stems from a different approach. After performing the integration by parts on the diffusion term,

$$-(\boldsymbol{\nabla} \cdot D\boldsymbol{\nabla}\phi, w)_{\mathbb{V}} = (D\boldsymbol{\nabla}\phi, \boldsymbol{\nabla}w)_{\mathbb{V}} - (D\boldsymbol{\nabla}\phi \cdot \hat{n}, w)_{\partial\mathbb{V}^i} - (D\boldsymbol{\nabla}\phi \cdot \hat{n}, w)_{\partial\mathbb{V}^d}, \quad (26)$$

we employ Equation 23:

$$-(\boldsymbol{\nabla} \cdot D\boldsymbol{\nabla}\phi, w)_{\mathbb{V}} = (D\boldsymbol{\nabla}\phi, \boldsymbol{\nabla}w)_{\mathbb{V}} - (D\boldsymbol{\nabla}\phi \cdot \hat{n}, w)_{\partial\mathbb{V}^i} + \frac{1}{2}(\phi, w)_{\partial\mathbb{V}^d} \quad (27)$$

Thus, Method 3 is

$$\begin{aligned}
 b_{MIP,3}(\phi, w) = & (\sigma_a \phi, w)_{\mathbb{V}} + (D \nabla \phi, \nabla w)_{\mathbb{V}} \\
 & + (\kappa_e [\![\phi]\!], [\![w]\!])_{\partial \mathbb{V}^i} + ([\![\phi]\!], \{ \! \{ D \partial_n w \! \} \}_{\partial \mathbb{V}^i}) + (\{ \! \{ D \partial_n \phi \! \} \}, [\![w]\!]_{\partial \mathbb{V}^i} \\
 & + \frac{1}{2} (\phi, w)_{\partial \mathbb{V}^d} \quad (28)
 \end{aligned}$$

We will implement these three methods and investigate their impact on the solution behavior. We will first solve a 1-D analytic diffusion equation problem with zero incident current boundary conditions to evaluate our implementation.

#### 4.2.2 Fourier Analysis

#### 4.2.3 Results

#### 4.2.4 As a Preconditioner

## 5 R-Z Geometry

Solving the transport equation in different coordinate systems may provide simpler ways of modeling a particular geometry or symmetry. In this section, we derive the *R-Z* transport equation to be solved. It assumes there is no variation in the azimuthal direction (of a cylinder), hence problems in *R-Z* geometry look very similar to problems in *X-Y* geometry. The streaming operator in cylindrical geometry is [1]

$$\boldsymbol{\Omega} \cdot \nabla \psi = \frac{\mu}{r} \frac{\partial}{\partial r}(r\psi) + \frac{\eta}{r} \frac{\partial \psi}{\partial \zeta} + \xi \frac{\partial \psi}{\partial z} - \frac{1}{r} \frac{\partial}{\partial \omega}(\eta\psi), \quad (29)$$

where  $\boldsymbol{\Omega}$  is the direction of travel unit vector,  $\psi$  is the angular flux, and

$$\mu \equiv \boldsymbol{\Omega} \cdot \hat{e}_r = \sqrt{1 - \xi^2} \cos \omega = \sin(\theta) \cos(\omega), \quad (30)$$

$$\eta \equiv \boldsymbol{\Omega} \cdot \hat{e}_\theta = \sqrt{1 - \xi^2} \sin \omega = \sin(\theta) \sin(\omega), \quad (31)$$

$$\xi \equiv \boldsymbol{\Omega} \cdot \hat{e}_z = \cos(\theta). \quad (32)$$

The variables  $\mu$ ,  $\eta$ ,  $\xi$ ,  $\omega$ , and  $\theta$  are shown in the cylindrical coordinate system in Figure 3. We assume there is no solution variation in the azimuthal direction, i.e.

$$\frac{\partial \psi}{\partial \zeta} \equiv 0, \quad (33)$$

which simplifies the streaming term to

$$\boldsymbol{\Omega} \cdot \nabla \psi = \frac{\mu}{r} \frac{\partial}{\partial r}(r\psi) + \xi \frac{\partial \psi}{\partial z} - \frac{1}{r} \frac{\partial}{\partial \omega}(\eta\psi). \quad (34)$$

The transport equation in *R-Z* geometry is then

$$\begin{aligned} \frac{\mu}{r} \frac{\partial}{\partial r} r\psi(r, z, \boldsymbol{\Omega}) + \xi \frac{\partial}{\partial z} \psi(r, z, \boldsymbol{\Omega}) - \frac{1}{r} \frac{\partial}{\partial \omega} \eta\psi(r, z, \boldsymbol{\Omega}) + \sigma_t(r, z) \psi(r, z, \boldsymbol{\Omega}) \\ = \frac{1}{4\pi} \int_{4\pi} \sigma_s(r, z) I(r, z, \boldsymbol{\Omega}') d\Omega' + S_0(r, z, \boldsymbol{\Omega}) \end{aligned} \quad (35)$$

where  $\sigma_t$  is the total cross section,  $\sigma_s$  is the scattering cross section, and  $S_0$  is an isotropic source as before.

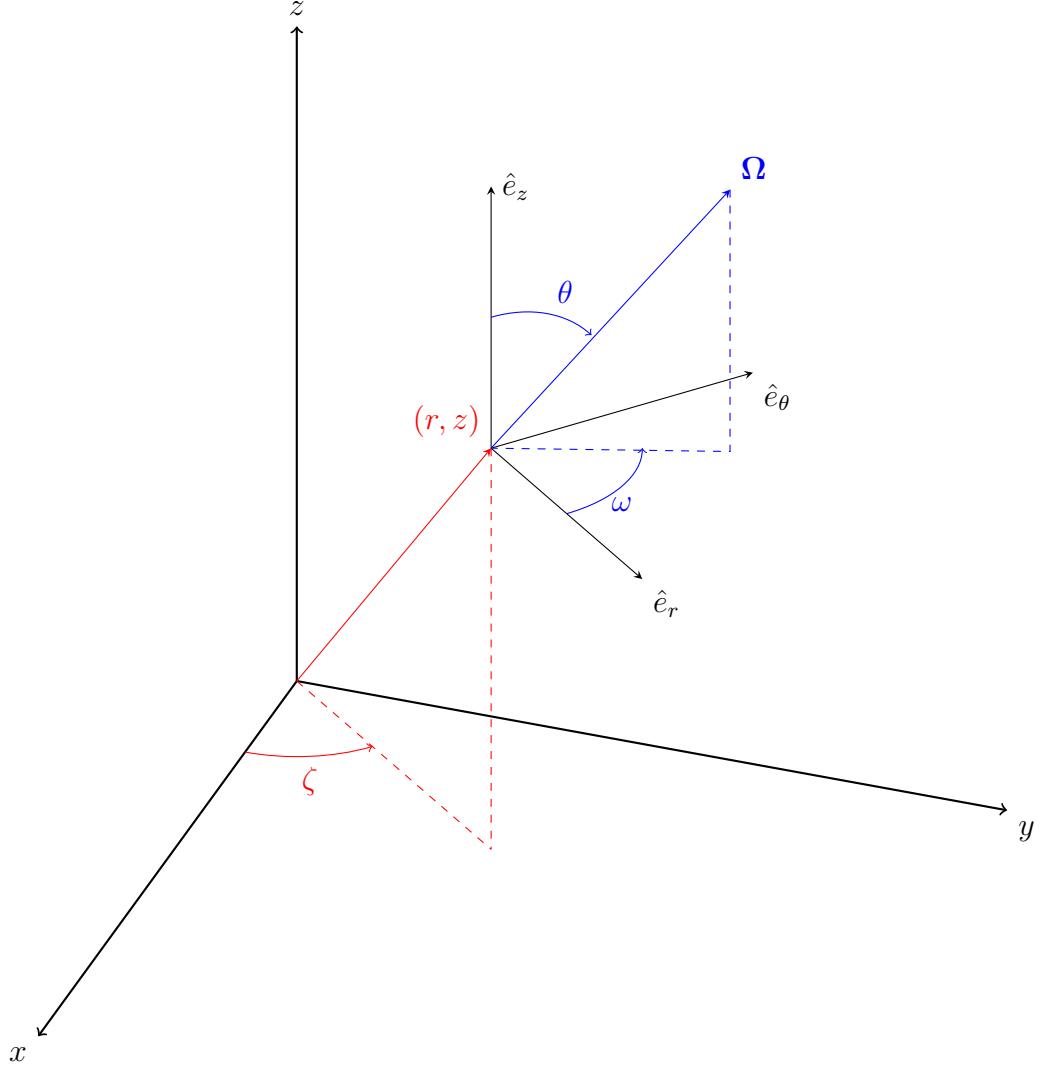


Figure 3: Cylindrical space-angle coordinate system showing the position  $(r, z)$  and direction of travel  $\Omega$ .

## 5.1 Angular Discretization

Discretizing Equation 35 with a level-symmetric angular quadrature results in

$$\begin{aligned} \frac{\mu_{n,m}}{r} \frac{\partial}{\partial r} r \psi_{n,m}(r, z) + \xi_n \frac{\partial}{\partial z} \psi_{n,m}(r, z) - \frac{1}{r} \frac{\partial}{\partial \omega} \eta_{n,m} \psi_{n,m}(r, z) + \sigma_t(r, z) \psi_{n,m}(r, z) \\ = \frac{1}{4\pi} \int_{4\pi} \sigma_s(r, z) I(r, z, \Omega') d\Omega' + S_0(r, z, \Omega) \quad (36) \end{aligned}$$

for direction  $\Omega_{n,m}$ , where index \$n\$ describes a level of quadrature with constant \$\xi\$ and

the  $m$  index denotes the quadrature point on that level. The  $(n, m)$  indexing is shown in Figure 4.

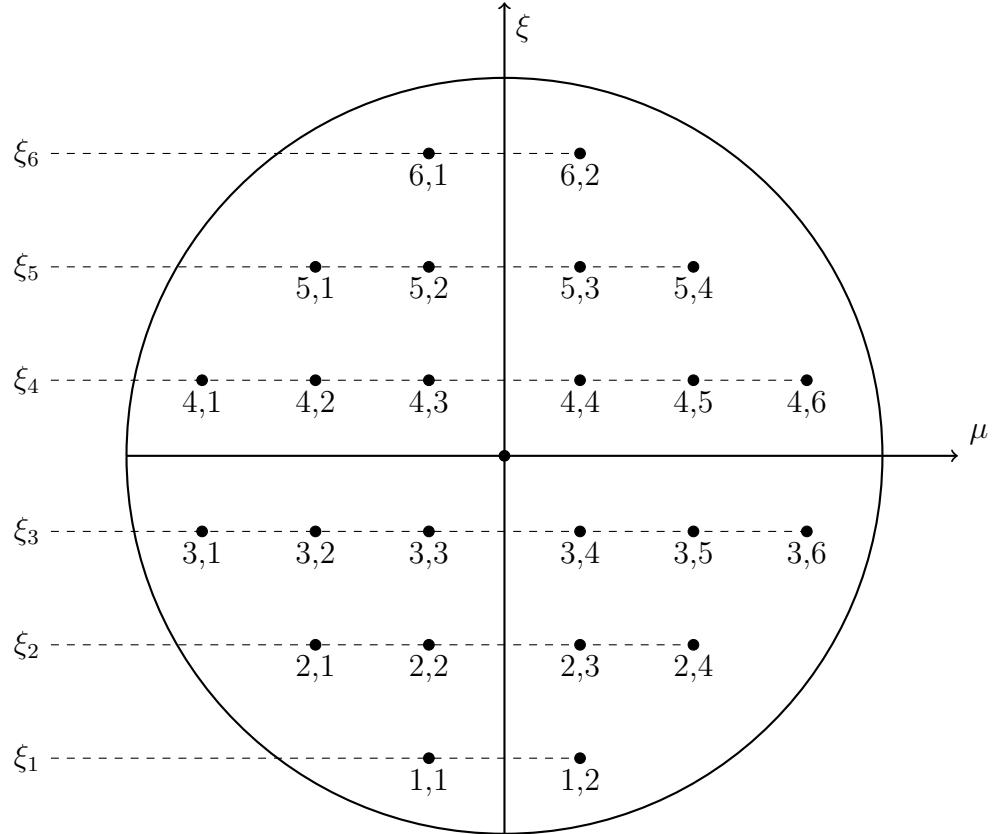


Figure 4: Angular discretization showing  $(\xi, \mu)$  pairs; adapted from [1]

One of the major challenges is handling the angular derivative term. Lewis and Miller [1] describes an approximation for the partial derivative of the intensity with respect to  $\omega$ :

$$-\frac{1}{r} \frac{\partial}{\partial \omega} \eta_{m,n} \psi_{n,m}(r, z) = \frac{\alpha_{m+1/2}^n \psi_{n,m+1/2}(r, z) - \alpha_{m-1/2}^n \psi_{n,m-1/2}(r, z)}{rw_{n,m}} \quad (37)$$

where  $\alpha_{m+1/2}^n$  and  $\alpha_{m-1/2}^n$  are angular differencing coefficients, and  $w_{n,m}$  is the angular

quadrature weight. We substitute this into Equation 46,

$$\begin{aligned} & \frac{\mu_{n,m}}{r} \frac{\partial}{\partial r} r \psi_{n,m}(r, z) + \xi_n \frac{\partial}{\partial z} \psi_{n,m}(r, z) \\ & + \frac{\alpha_{m+1/2}^n \psi_{m+1/2,n}(r, z) - \alpha_{m-1/2}^n \psi_{m-1/2,n}(r, z)}{rw_{n,m}} + \sigma_t(r, z) \psi_{n,m}(r, z) \\ & = \frac{1}{4\pi} \int_{4\pi} \sigma_s(r, z) \psi(r, z, \Omega') d\Omega' + \frac{1}{4\pi} S_0(r, z) \quad (38) \end{aligned}$$

Here, we pause to notice that there are similarities and differences between our Cartesian discretization. The absorption term, axial derivative term, and right-hand-side are the same in both coordinate systems. The differences arise in the radial and angular derivative terms.

After multiplying through by the radius  $r$ , the radial derivative term has a factor of  $r$  inside the derivative. The angular derivative term is also new and does not resemble a mass matrix so MFEM will require additional modification.

Requiring Equation 38 to satisfy the uniform infinite medium solution results in the condition,

$$\alpha_{m+1/2}^n = \alpha_{m-1/2}^n - \mu_{n,m} w_{n,m} \quad (39)$$

If  $\alpha_{1/2}^n$  is known, then the remaining coefficients are uniquely determined. To find  $\alpha_{1/2}^n$ , we require that Equation 38 satisfy the conservation equation (Eq. 35). Given a quadrature set with an even number of  $\mu_{n,m}$  values, setting  $\alpha_{1/2}^n = 0$  results in  $\alpha_{M_n+1/2}^n = 0$  per Equation 39 and the conservation equation is satisfied.

A relationship between  $\psi_{n,m}$ ,  $\psi_{n,m+1/2}$ , and  $\psi_{n,m-1/2}$  must be established. A weighted diamond difference scheme has been established by Morel and Montry [56],

$$\psi_{n,m}(r, z) = \tau_{n,m} \psi_{n,m+1/2} + (1 - \tau_{n,m}) \psi_{n,m-1/2} \quad (40)$$

where  $\tau_{n,m}$  linearly interpolates  $\mu$ :

$$\tau_{n,m} = \frac{\mu_{n,m} - \mu_{n,m-1/2}}{\mu_{n,m+1/2} - \mu_{n,m-1/2}} \quad (41)$$

with

$$\mu_{n,m+1/2} = \sqrt{1 - \xi_n^2} \cos(\varphi_{n,m+1/2}) \quad (42)$$

$$\varphi_{n,m+1/2} = \varphi_{n,m-1/2} + \pi \frac{w_{n,m}}{w_n} \quad (43)$$

$$w_n = \sum_{m=1}^{M_n} w_{n,m} \quad (44)$$

We take Equation 38, multiply through by  $r$  and perform a product rule on the radial derivative term,

$$\begin{aligned} & \mu_{n,m} \left[ \psi_{n,m}(r, z) + r \frac{\partial}{\partial r} \psi_{n,m}(r, z) \right] + r \xi_n \frac{\partial}{\partial z} \psi_{n,m}(r, z) \\ & + \frac{\alpha_{m+1/2}^n \psi_{m+1/2,n}(r, z) - \alpha_{m-1/2}^n \psi_{m-1/2,n}(r, z)}{w_{n,m}} + r \sigma_t(r, z) \psi_{n,m}(r, z) \\ & = \frac{r}{4\pi} \int_{4\pi} \sigma_s(r, z) \psi(r, z, \Omega') d\Omega' + \frac{r}{4\pi} S_0(r, z). \end{aligned} \quad (45)$$

We solve Equation 40 for  $\psi_{n,m+1/2}$ , perform a substitution, and move the known quantities to the right-hand-side,

$$\begin{aligned} & \mu_{n,m} r \frac{\partial}{\partial r} \psi_{n,m}(r, z) + r \xi_n \frac{\partial}{\partial z} \psi_{n,m}(r, z) + \mu_{n,m} \psi_{n,m}(r, z) \\ & + \frac{\alpha_{m+1/2}^n}{\tau_{n,m} w_{n,m}} \psi_{n,m}(r, z) + r \sigma_t(r, z) \psi_{n,m}(r, z) \\ & = \frac{r}{4\pi} \int_{4\pi} \sigma_s(r, z) \psi(r, z, \Omega') d\Omega' + \frac{r}{4\pi} S_0(r, z) \\ & + \left( \frac{1 - \tau_{n,m}}{\tau_{n,m}} \frac{\alpha_{m+1/2}^n}{w_{n,m}} + \frac{\alpha_{m-1/2}^n}{w_{n,m}} \right) \psi_{n,m-1/2}(r, z). \end{aligned} \quad (46)$$

Given a level-symmetric quadrature set, all of the  $\alpha_{n,m\pm 1/2}^n$  and  $\tau_{n,m}$  values can be computed. We solve the starting direction equation to obtain  $\psi_{n,1/2}$ . That is, we

solve the  $X$ - $Y$  system for directions  $\Omega_{n,1/2}$ ,

$$\Omega_{n,1/2} \cdot \nabla \psi_{n,1/2} + \sigma_t \psi_{n,1/2} = \frac{1}{4\pi} \sigma_s \phi + \frac{1}{4\pi} S_0 \quad (47)$$

There is an alternate angular discretization method developed by Warsa and Prinja [57]. Instead of finding an approximation for the angular derivative, they perform a product rule:

$$\frac{\partial \psi}{\partial \omega} \equiv \frac{\partial \mu}{\partial \omega} \frac{\partial \psi}{\partial \mu} \quad (48)$$

Since,

$$\frac{\partial \mu}{\partial \omega} \equiv -\xi, \quad (49)$$

The angular derivative can be written

$$\frac{\partial \psi}{\partial \omega} \equiv -\xi \frac{\partial \psi}{\partial \mu} \quad (50)$$

Here, an approximation for the  $\mu$ -derivative must be established.

## 5.2 Spatial Discretization

The finite element discretization is performed here. The methodology is similar to the Cartesian geometry. First, we subdivide a problem domain using a spatial mesh. Then, we multiply Equation 46 by a test function and integrate over the volume of a single mesh zone,

$$\begin{aligned} & (r \Omega_{n,m} \cdot \nabla \psi_{n,m}, v_i)_{\mathbb{D}} + (\mu_{n,m} \psi_{n,m}, v_i)_{\mathbb{D}} \\ & + \left( \frac{\alpha_{m+1/2}^n}{\tau_{n,m} w_{n,m}} \psi_{n,m}, v_i \right)_{\mathbb{D}} + (r \sigma_t \psi_{n,m}, v_i)_{\mathbb{D}} \\ & = \left( \frac{r}{4\pi} \int_{4\pi} \sigma_s \psi d\Omega', v_i \right)_{\mathbb{D}} + \left( \frac{r}{4\pi} S_0, v_i \right)_{\mathbb{D}} \\ & + \left( \left( \frac{1 - \tau_{n,m}}{\tau_{n,m}} \frac{\alpha_{m+1/2}^n}{w_{n,m}} + \frac{\alpha_{m-1/2}^n}{w_{n,m}} \right) \psi_{n,m-1/2}, v_i \right)_{\mathbb{D}}, \end{aligned} \quad (51)$$

where the Cartesian gradient operator is used and the inner product notation,

$$(a, b)_{\mathbb{D}} \equiv \int_{\mathbb{D}} ab, \quad (52)$$

is used. We perform an integration by parts,

$$\begin{aligned} & (r\boldsymbol{\Omega}_{n,m} \cdot \hat{n}\psi_{n,m}, v_i)_{\partial\mathbb{D}} - (r\psi_{n,m}, \boldsymbol{\Omega}_{n,m} \cdot \nabla v_i)_{\mathbb{D}} + (\mu_{n,m}\psi_{n,m}, v_i)_{\mathbb{D}} \\ & + \left( \frac{\alpha_{m+1/2}^n}{\tau_{n,m}w_{n,m}} \psi_{n,m}, v_i \right)_{\mathbb{D}} + (r\sigma_t\psi_{n,m}, v_i)_{\mathbb{D}} \\ & = \left( \frac{r}{4\pi} \int_{4\pi} \sigma_s \psi d\Omega', v_i \right)_{\mathbb{D}} + \left( \frac{r}{4\pi} S_0, v_i \right)_{\mathbb{D}} \\ & + \left( \left( \frac{1 - \tau_{n,m}}{\tau_{n,m}} \frac{\alpha_{m+1/2}^n}{w_{n,m}} + \frac{\alpha_{m-1/2}^n}{w_{n,m}} \right) \psi_{n,m-1/2}, v_i \right)_{\mathbb{D}}, \end{aligned} \quad (53)$$

to obtain our angular and spatially discretized  $R$ - $Z$  transport equation.

#### WE PERFORMED SOME STUDIES TO MAKE SURE IT'S RIGHT...

We first solved a uniform infinite medium problem with  $\sigma_t = 1.0$ ,  $\sigma_s = 0.3$ , and  $S_0 = 0.7$  for 1<sup>st</sup>-order FEM on a 2<sup>nd</sup>-order mesh using  $S_4$  level-symmetric angular quadrature. The solution, shown in Figure 5, demonstrates we get the exact flat solution of  $\phi = 1.0$ .

We tested several MMS problems as well. First, we defined the manufactured solution

$$\psi = (1 - \mu^2)(1 - \xi^2) \sin\left(\frac{\pi}{2}r\right) \cos(\pi z) \quad (54)$$

with 2<sup>nd</sup>-order FEM, Orthogonal quadrilateral mesh,  $\sigma_t = 1.0$ ,  $\sigma_s = 0.3$ ,  $S_0 = 0.7$ ,  $S_4$  level-symmetric angular quadrature. The solution is shown in Figure 6 and the L<sup>2</sup>-error was 0.132.

Removing the angular dependence in the manufactured solution,

$$\psi = \sin\left(\frac{\pi}{2}r\right) \cos(\pi z), \quad (55)$$

increased the accuracy of our DGFEM approximation. Shown in Figure 7, the L<sup>2</sup>-

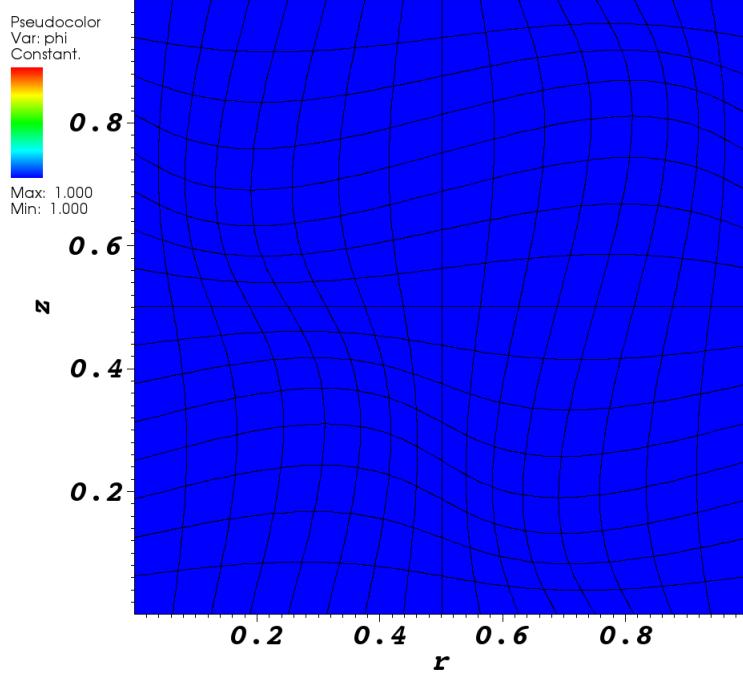


Figure 5: Uniform infinite medium solution.

error was reduced to  $4.59 \times 10^{-5}$ .

Bailey et al. [18] showed 2<sup>nd</sup>-order convergence using PWLD and BLD using the manufactured solution

$$\psi_{\text{MMS}}(r, z) = (\sin(\pi r) + 1 - r) \sin(\pi z), \quad (56)$$

for  $\sigma_t = 3 \text{ cm}^{-1}$  and  $\sigma_s = 0.9999\sigma_t$ . We solved this same problem using  $p = \{1, 2, 4, 6, 8\}$  on an orthogonal and 2<sup>nd</sup>-order curved mesh using  $S_8$  level-symmetric angular quadrature. The incident angular flux is equal to Equation 56. Figure 8 shows the  $p = 2$  solution on a 2<sup>nd</sup>-order mesh.

The spatial convergence study performed by Bailey et al. demonstrated 2<sup>nd</sup>-order converge for their 1<sup>st</sup>-order methods. Figures 9a and 9b demonstrate  $O(p+1)$  convergence on an orthogonal mesh and 2<sup>nd</sup>-order mesh, respectively. Reference lines

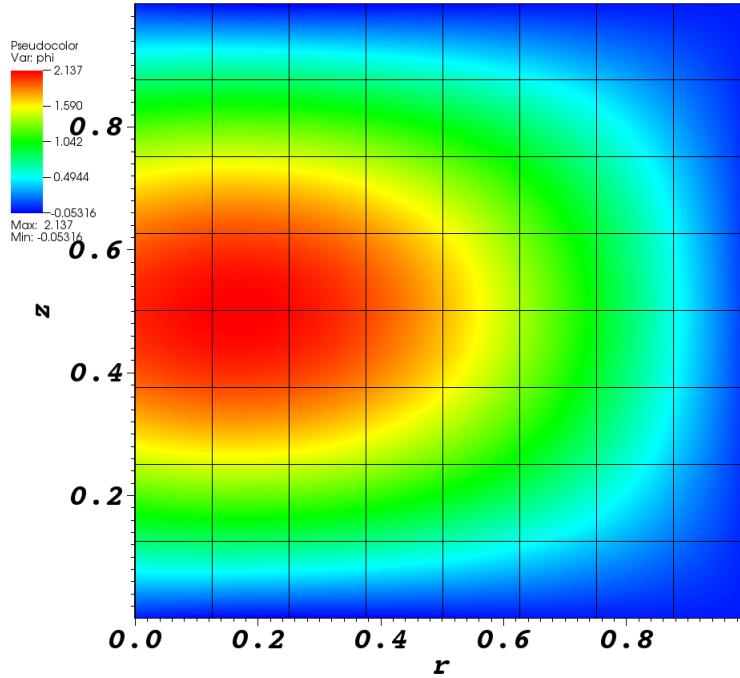


Figure 6: MMS solution for Equation 54.

are also provided for comparison.

### 5.3 Lumping

### 5.4 Diffusion Synthetic Acceleration

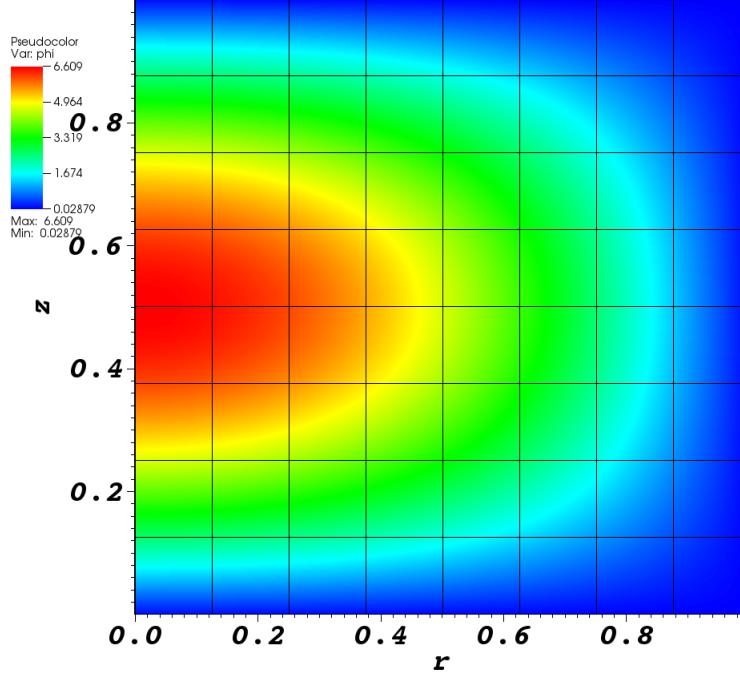


Figure 7: MMS solution for Equation 55.

## 5.5 Symmetry Preservation

We want  $R$ - $Z$  geometry to solve and preserve 1-dimensional spherical solutions. The transport equation in spherical geometry is

$$\Omega \cdot \nabla \psi = \frac{\partial \psi}{\partial r} \Omega \cdot \nabla r + \frac{\partial \psi}{\partial \mu} \Omega \cdot \nabla \mu \quad (57)$$

$$= \mu \frac{\partial \psi}{\partial r} + \frac{(1 - \mu^2)}{r} \frac{\partial \psi}{\partial \mu} \quad (58)$$

$$= \frac{\mu}{r^2} \frac{\partial}{\partial r} (r^2 \psi) + \frac{\partial}{\partial \mu} \left[ \frac{(1 - \mu^2) \psi}{r} \right] \quad (59)$$

$$\mu \frac{\partial}{\partial r} (r^2 \psi) + r \frac{\partial}{\partial \mu} ((1 - \mu^2) \psi) + r^2 \sigma_t \psi = \frac{1}{4\pi} \sigma_s \phi + \frac{1}{4\pi} S_0. \quad (60)$$

In 1-dimensional geometry, this simplifies to

$$\mu \frac{\partial \psi}{\partial r} + \sigma_t \psi = \frac{1}{4\pi} \sigma_s \phi + \frac{1}{4\pi} S_0. \quad (61)$$

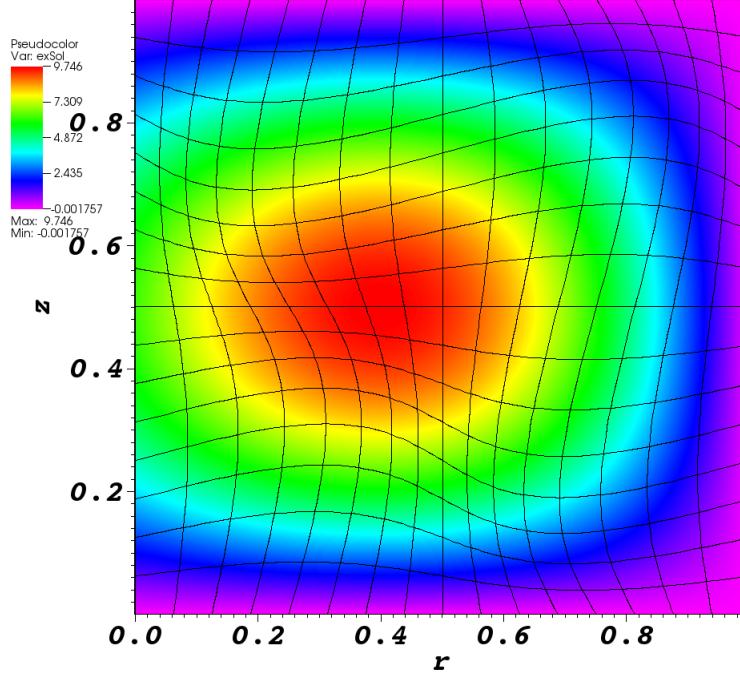


Figure 8: MMS solution to Equation 56.

We can solve this for  $\psi$ .

$$\frac{\partial \psi(r)}{\partial r} e^{\sigma_t r / \mu} + \frac{\sigma_t}{\mu} \psi(r) e^{\sigma_t r / \mu} = \frac{1}{4\pi\mu} (\sigma_s \phi + S_0) e^{\sigma_t r / \mu} \quad (62)$$

$$\frac{\partial}{\partial r} (\psi(r) e^{\sigma_t r / \mu}) = \frac{1}{4\pi\mu} (\sigma_s \phi + S_0) e^{\sigma_t r / \mu} \quad (63)$$

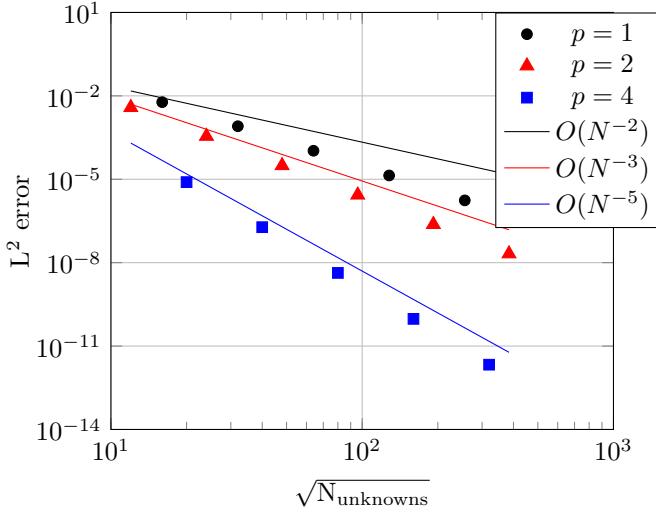
$$\int \frac{\partial}{\partial r'} (\psi(r') e^{\sigma_t r' / \mu}) dr' = \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' \quad (64)$$

$$\psi(r) e^{\sigma_t r / \mu} - c = \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' \quad (65)$$

$$\psi(r) = e^{-\sigma_t r / \mu} \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' + ce^{-\sigma_t r / \mu} \quad (66)$$

$$\psi(1) = 1 = e^{-\sigma_t / \mu} \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' + ce^{-\sigma_t / \mu} \quad (67)$$

$$c = e^{\sigma_t / \mu} - \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' \quad (68)$$



(a) Orthogonal quadrilateral mesh.

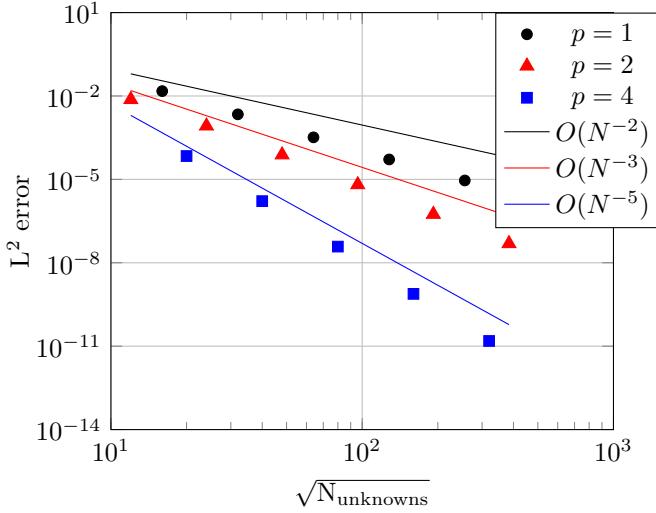
(b) 2<sup>nd</sup>-order curved mesh.

Figure 9:  $L^2$ -norm of the errors from the manufactured solution and reference lines, where  $N_{\text{unknowns}} = N_{\text{cells}}(p + 1)^2$ .

$$\begin{aligned} \psi(r) = e^{-\sigma_t r / \mu} & \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' + \\ & \left[ e^{\sigma_t / \mu} - \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' \right] e^{-\sigma_t r / \mu} \quad (69) \end{aligned}$$

$$\psi(r) = e^{-\sigma_t r / \mu} \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' + e^{(1-r)\sigma_t / \mu} - e^{-\sigma_t r / \mu} \frac{1}{4\pi\mu} \int (\sigma_s \phi(r') + S_0) e^{\sigma_t r' / \mu} dr' \quad (70)$$

$$\psi(r) = e^{(1-r)\sigma_t / \mu} \quad (71)$$

We use MMS to solve a 1-D spherical problem using the *R-Z* geometry spatial discretization. The manufactured solution is

$$\psi_{\text{MMS}}(\rho) = \sin(\pi\rho) + 2 - \rho, \quad (72)$$

where  $\rho = \sqrt{r^2 + z^2}$  is the distance from the origin (i.e. the spherical radius). First, we integrate the streaming term by parts,

$$\boldsymbol{\Omega} \cdot \nabla \psi = \mu \frac{\partial \psi}{\partial r} + \frac{\mu}{r} \psi + \xi \frac{\partial \psi}{\partial r} - \frac{\mu}{r} \psi - \frac{\eta}{r} \frac{\partial \psi}{\partial \omega} \quad (73)$$

$$= \mu \frac{\partial \psi}{\partial r} + \xi \frac{\partial \psi}{\partial z} - \frac{\eta}{r} \frac{\partial \psi}{\partial \omega} \quad (74)$$

Substituting the manufactured solution into this *R-Z* transport equation,

$$\begin{aligned} & \mu \frac{\partial}{\partial r} (\sin(\pi\rho) + 2 - \rho) + \xi \frac{\partial}{\partial z} (\sin(\pi\rho) + 2 - \rho) \\ & - \frac{\eta}{r} \frac{\partial}{\partial \omega} [(\sin(\pi\rho) + 2 - \rho)] + \sigma_t (\sin(\pi\rho) + 2 - \rho) \\ & = \frac{1}{2\pi} \sigma_s \phi_{\text{MMS}} + \frac{1}{2\pi} S_0. \end{aligned} \quad (75)$$

Integrating  $\psi_{\text{MMS}}$  over all directions reveals  $\phi_{\text{MMS}}$ ,

$$\begin{aligned} & \mu \frac{\partial}{\partial r} (\sin(\pi\rho) + 2 - \rho) + \xi \frac{\partial}{\partial z} (\sin(\pi\rho) + 2 - \rho) \\ & - \frac{\eta}{r} \frac{\partial}{\partial \omega} [(\sin(\pi\rho) + 2 - \rho)] + \sigma_t (\sin(\pi\rho) + 2 - \rho) \\ & = \sigma_s (\sin(\pi\rho) + 2 - \rho) + \frac{1}{2\pi} S_0. \end{aligned} \quad (76)$$

We perform some simplifications,

$$\frac{\partial \psi}{\partial r} = \frac{\pi r \cos(\pi \sqrt{r^2 + z^2})}{\sqrt{r^2 + z^2}} - \frac{r}{\sqrt{r^2 + z^2}} \quad (77)$$

$$\frac{\partial \psi}{\partial z} = \frac{\pi z \cos(\pi \sqrt{r^2 + z^2})}{\sqrt{r^2 + z^2}} - \frac{r}{\sqrt{r^2 + z^2}} \quad (78)$$

$$\begin{aligned} & \mu \frac{\partial}{\partial r} (\sin(\pi\rho) + 2 - \rho) + \xi \frac{\partial}{\partial z} (\sin(\pi\rho) + 2 - \rho) \\ & - \frac{\eta}{r} \frac{\partial}{\partial \omega} [(\sin(\pi\rho) + 2 - \rho)] + \sigma_t (\sin(\pi\rho) + 2 - \rho) \\ & = \sigma_s (\sin(\pi\rho) + 2 - \rho) + \frac{1}{2\pi} S_0. \end{aligned} \quad (79)$$

Because of the angular derivative in the streaming term, we reduce the influence of the direction dependence as much as possible by using higher order level-symmetric angular quadrature.

We evaluate the relative asymmetry by calculating the averages of all nodes at each  $\rho$  value and

$$\phi_{\text{sym}}(\rho, \theta) = \frac{\phi_{\text{code}}(\rho, \theta) - \phi_{\text{avg}}(\rho)}{\phi_{\text{avg}}(\rho)}, \quad (80)$$

where

$$\phi_{\text{avg}}(\rho) = \frac{1}{N_{\text{nodes}}(\rho)} \sum_{i=1}^{N_{\text{nodes}}(\rho)} \phi(\rho, \theta_i) \quad (81)$$

is the average scalar flux at all nodes at the same spherical radius  $\rho$ .

### 5.5.1 Linear in Spherical Radius

The three traditional coordinate systems are Cartesian, cylindrical, and spherical. It is straight forward to describe a spatial position using any one of them. However, challenges arise when spatially discretizing. Specifically, angular derivatives arise inside

the streaming term in cylindrical and spherical geometries. In modeling a spherical problem, it would seem best to use a spherical coordinate system to describe the spatial domain. However, there are two angular derivatives that must now be handled with a numerical scheme. Cylindrical geometry may be used in lieu of spherical coordinates to describe a spherical problem. In this section, we demonstrate the use of  $R$ - $Z$  geometry to preserve a 1-D spherical solution.

We use the MMS with

$$\psi_{\text{MMS}}(r, z) = \rho = \sqrt{r^2 + z^2}. \quad (82)$$

This solution is linear in the 1-D spherical coordinate  $\rho$ . We solve this for  $p = \{1, 2, 4\}$ , using  $S_N$  level symmetric angular quadrature with  $N = \{4, 6, 8, 10, 12\}$ , on a 1<sup>st</sup>- and 2<sup>nd</sup>-order mesh. The problem has physical parameters  $\sigma_t = 5.0$  and  $\sigma_a = 2.0$ . Figures 10 - 12 show each finite element order  $p$  for select angular quadrature solutions.

We see from Figures ?? - ?? that increasing the angular quadrature order does qualitatively reduce the general asymmetry of the solution. Increasing the finite element order, however, reduces the asymmetry significantly more. Figure ?? compares  $p = \{1, 2, 4\}$  for  $S_8$  level-symmetric angular quadrature. We notice that the solution around  $\rho = 0$  persistently has the largest asymmetries. We note that this mesh is made up entirely of quadrilaterals, even the inner-most radial zones.

## 5.6 Other

## 5.7 Material Discontinuity Stress Test

We adapted this problem from Palmer [52] and solved it without DSA in Woods et al. [23]. There are five different material regions described in Table 2 and Figure 19.

This problem has opacities that range several orders of magnitude, resulting in

Table 2: Material discontinuity stress test with MIP DSA material properties.

Material Region	$\sigma_t \text{ cm}^{-1}$	$\sigma_s \text{ cm}^{-1}$	$S_0 \text{ cm}^{-2} \text{ s}^{-1}$
Source	1.0	1.0	1.0
Very thin absorber	0.0001	0.0	0.0
Thick absorber	10.0	0.0	0.0
Very thick absorber	100.0	0.0	0.0
Very thick scatterer	1000.0	1000.0	0.0

strong material discontinuities. We also introduce anisotropic incident intensities into the scattering region by preferentially attenuating intensities that are not perpendicular to the thick absorber. We expect some degradation in the DSA in problems with strong material discontinuities [22]. We also expect boundary layers to form from the anisotropic incident intensities [38]. The solution is shown in Figure 20.

## 5.8 Reflecting Boundary Conditions

To incorporate reflecting boundary conditions, we will “guess” the incident angular fluxes, update them with outgoing angular fluxes from the previous iteration, and adapt a convergence criterion for those fluxes. Along the z-axis, the reflection for direction  $\Omega = (\mu, \eta, \xi)$  is  $\Omega_R = (-\mu, \eta, \xi)$ .

## 5.9 Reflecting Boundary Conditions

This may not warrant an entire subsection.

Reflecting boundaries are dependent upon the direction of the outgoing angular flux,  $\Omega_m$ . The reflected incident direction is

$$\Omega^R = \Omega_m - 2(\Omega_m \cdot \hat{n})\hat{n} \quad (83)$$

where  $\Omega_m$  is the outgoing direction and  $\hat{n}$  is the unit normal vector on the boundary (pointing outward). We apply a Dirichlet boundary condition for the angular flux in the reflected direction,

$$\psi^b(\mathbf{r}, \Omega^R) = \psi_m \quad (84)$$

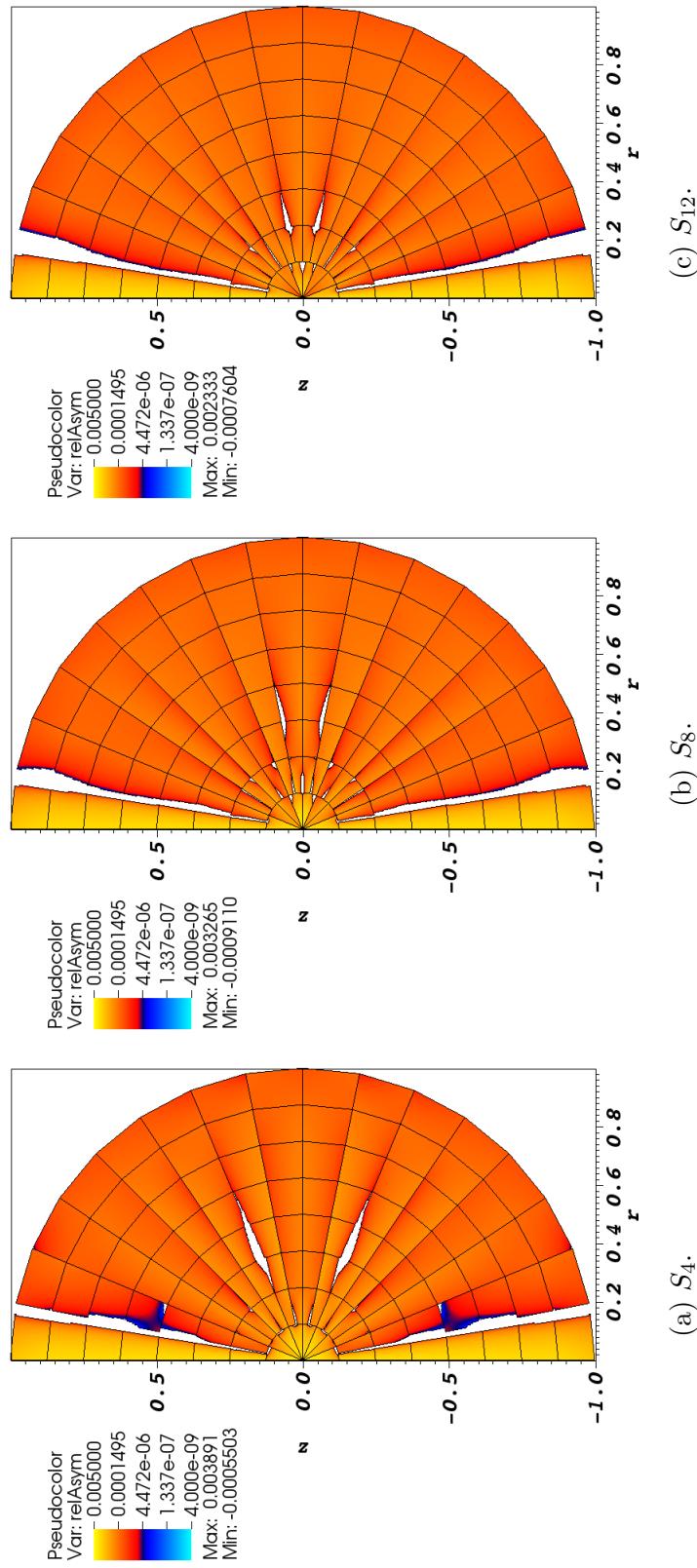


Figure 10: Relative asymmetry for 1<sup>st</sup>-order finite elements on a 1<sup>st</sup>-order mesh for given order of level-symmetric angular quadrature.

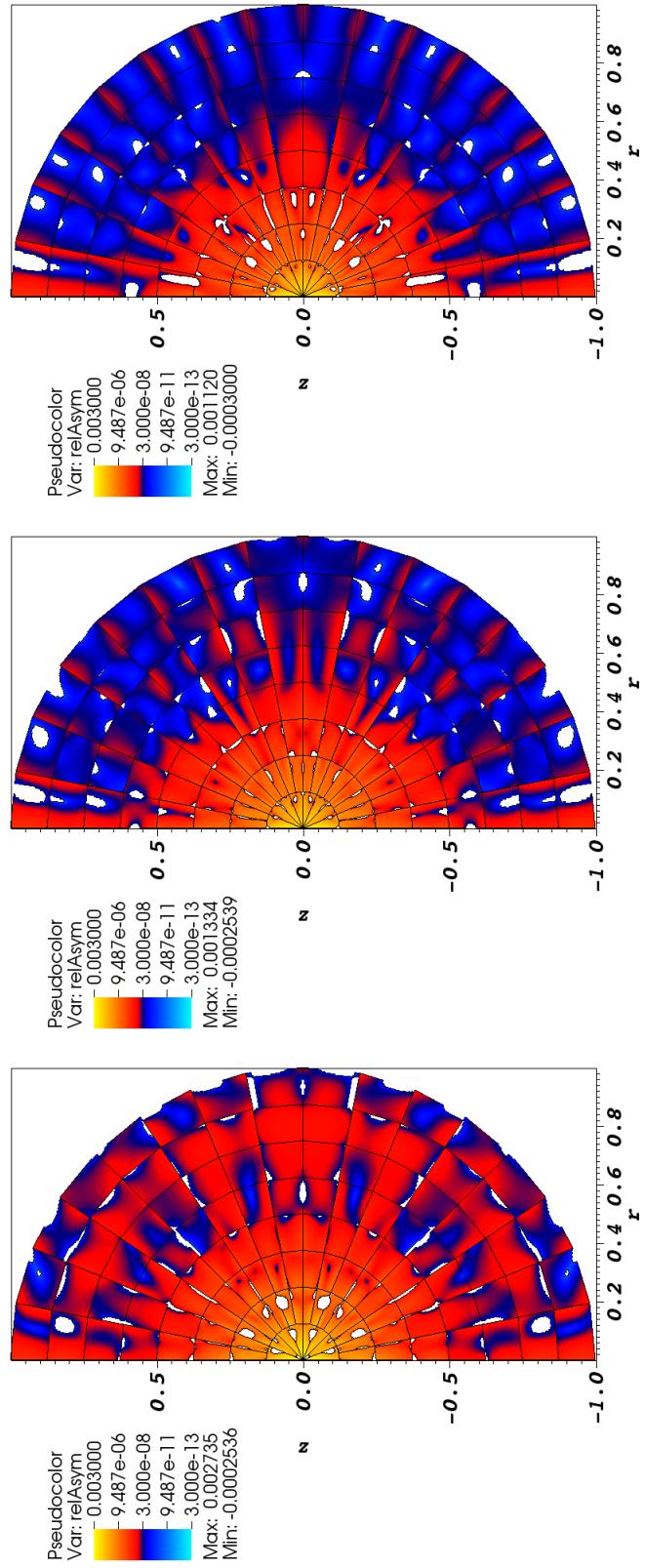


Figure 11: Relative asymmetry for 2<sup>nd</sup>-order finite elements on a 1<sup>st</sup>-order mesh for given order of level-symmetric angular quadrature.

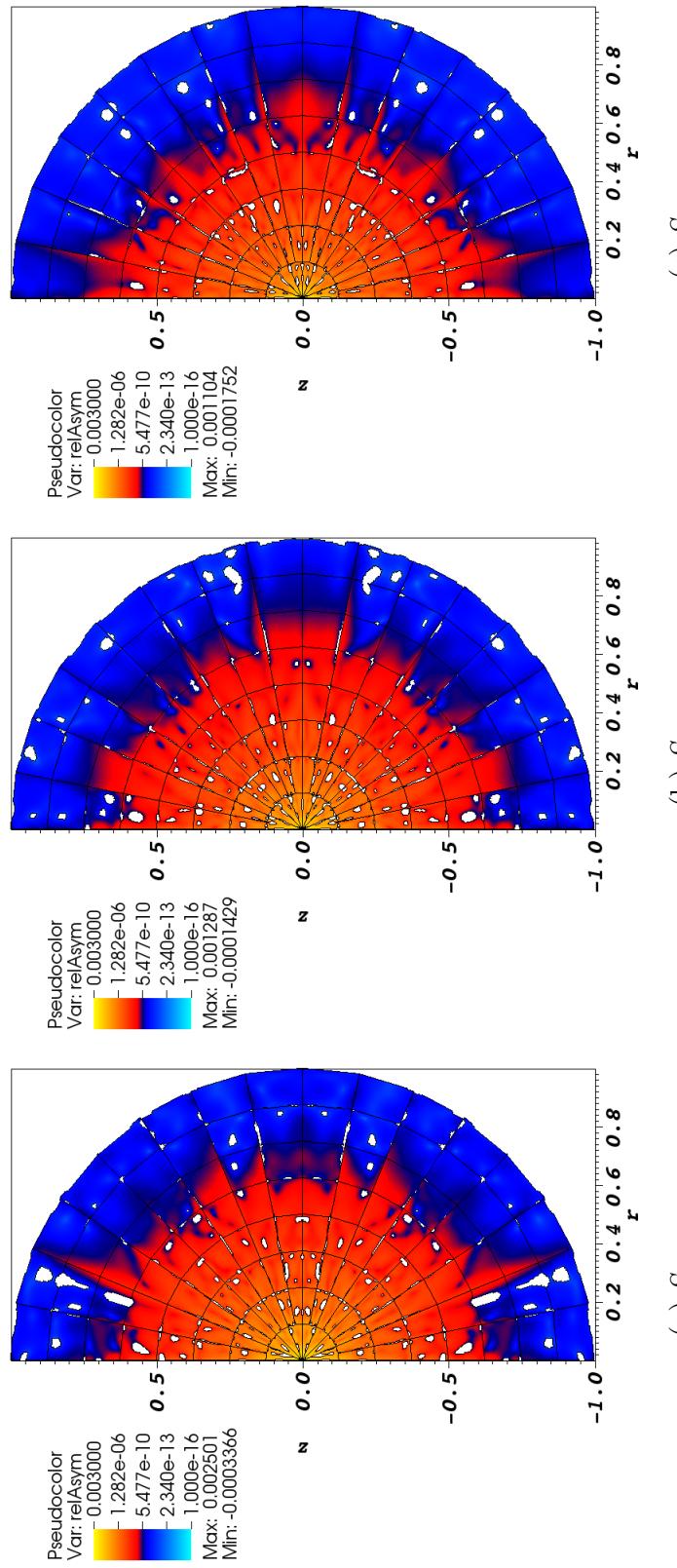


Figure 12: Relative asymmetry for 4<sup>st</sup>-order finite elements on a 1<sup>st</sup>-order mesh for given order of level-symmetric angular quadrature.

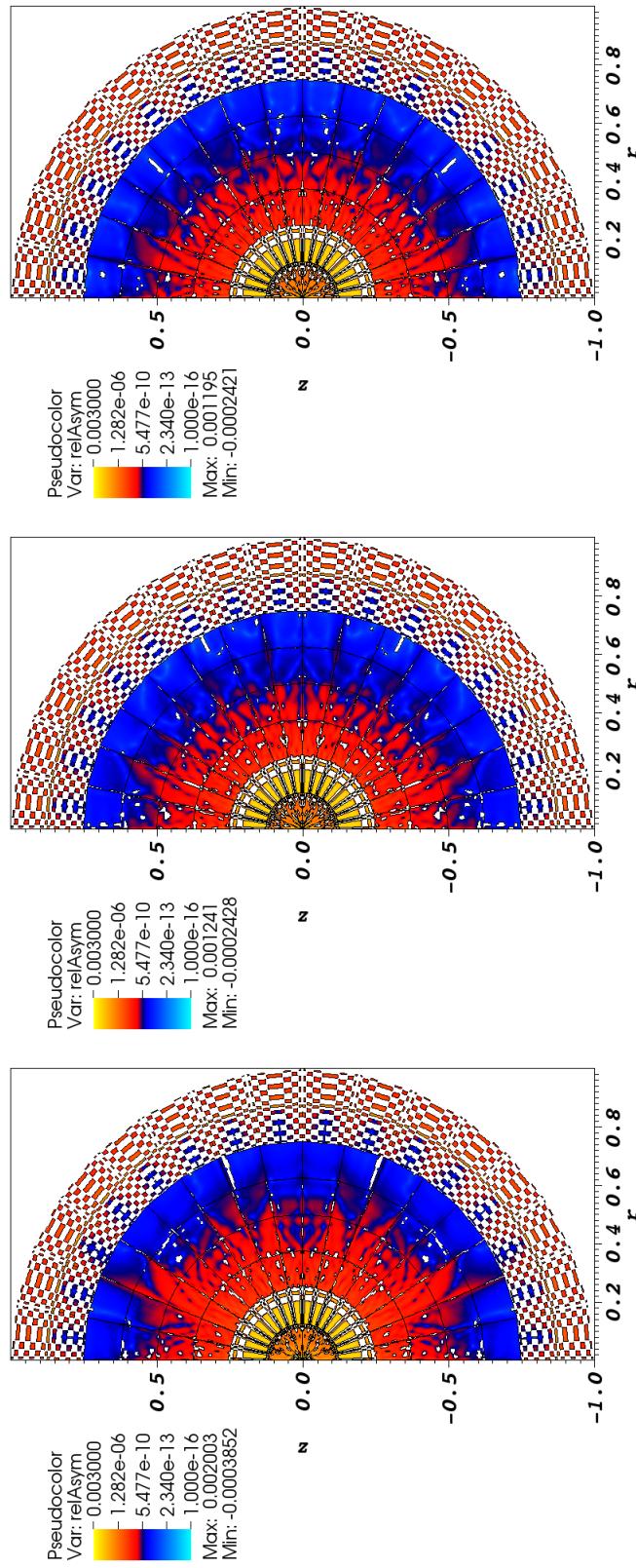


Figure 13: Relative asymmetry for 8<sup>st</sup>-order finite elements on a 1<sup>st</sup>-order mesh for given order of level-symmetric angular quadrature.

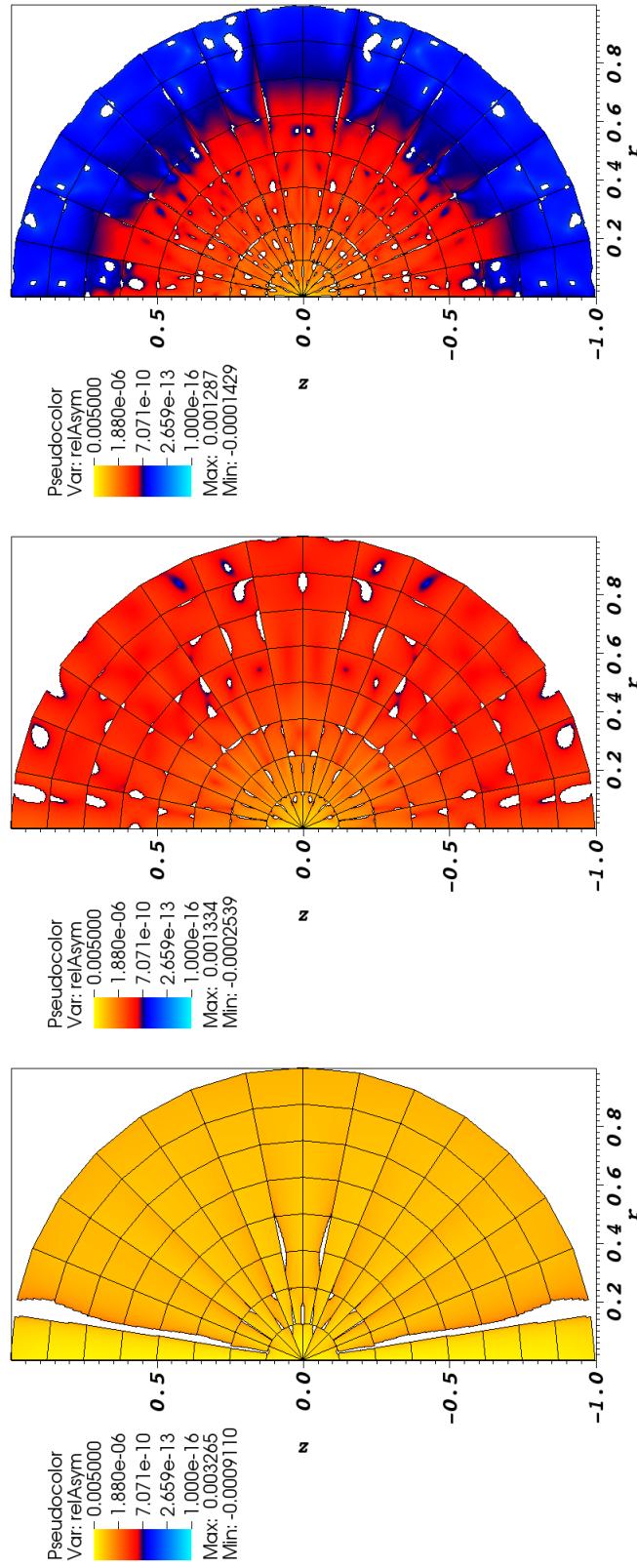


Figure 14: Relative asymmetry for  $p = \{1, 2, 4\}$  finite elements on a 1<sup>st</sup>-order mesh for  $S_8$  level-symmetric angular quadrature.

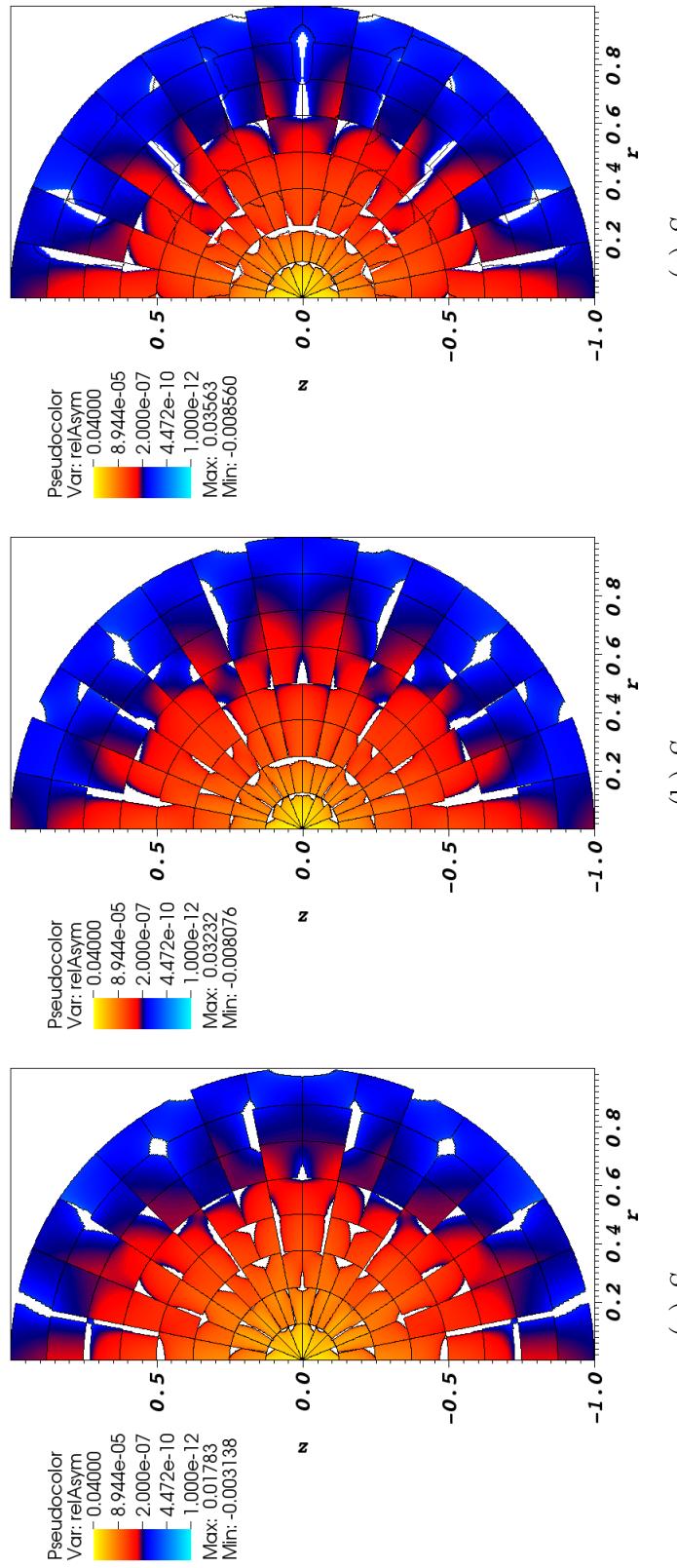


Figure 15: Relative asymmetry for 1<sup>st</sup>-order finite elements on a 2<sup>nd</sup>-order mesh for given order of level-symmetric angular quadrature.

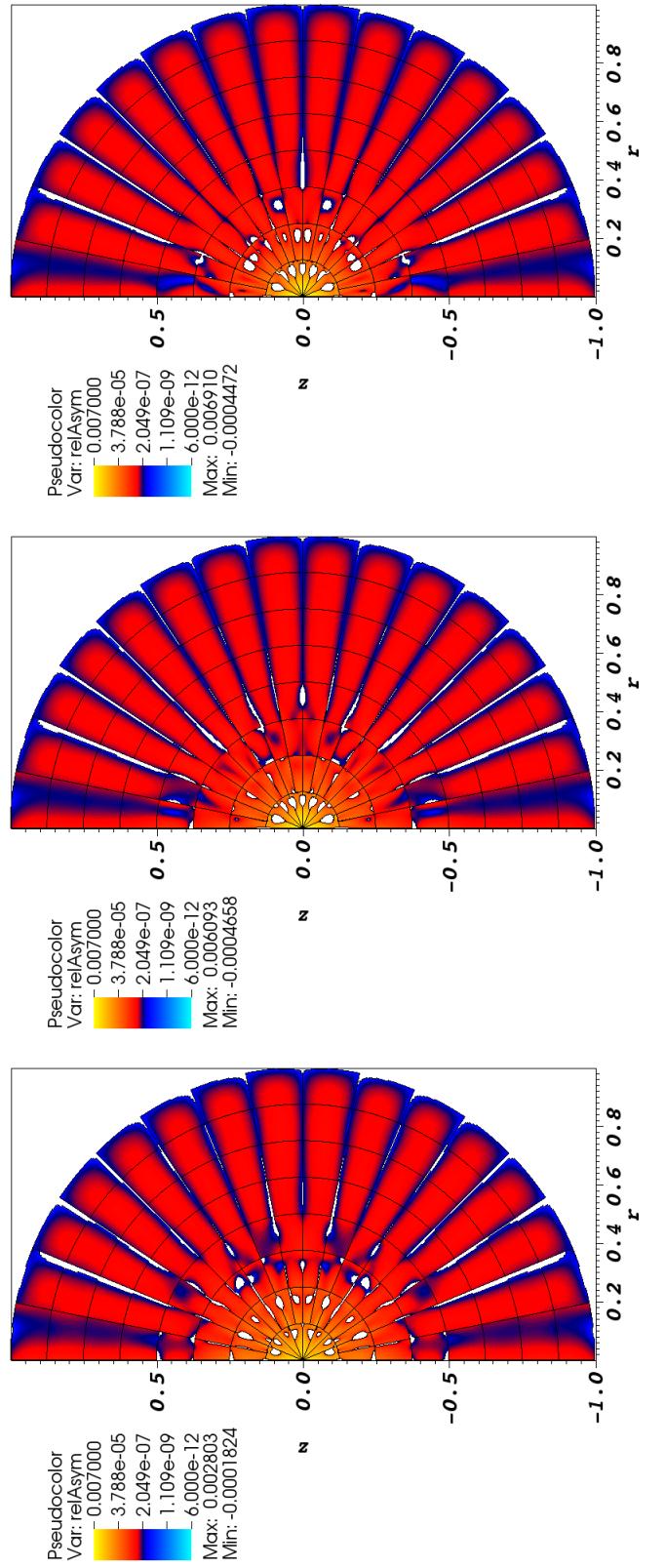


Figure 16: Relative asymmetry for 2<sup>nd</sup>-order finite elements on a 2<sup>nd</sup>-order mesh for given order of level-symmetric angular quadrature.

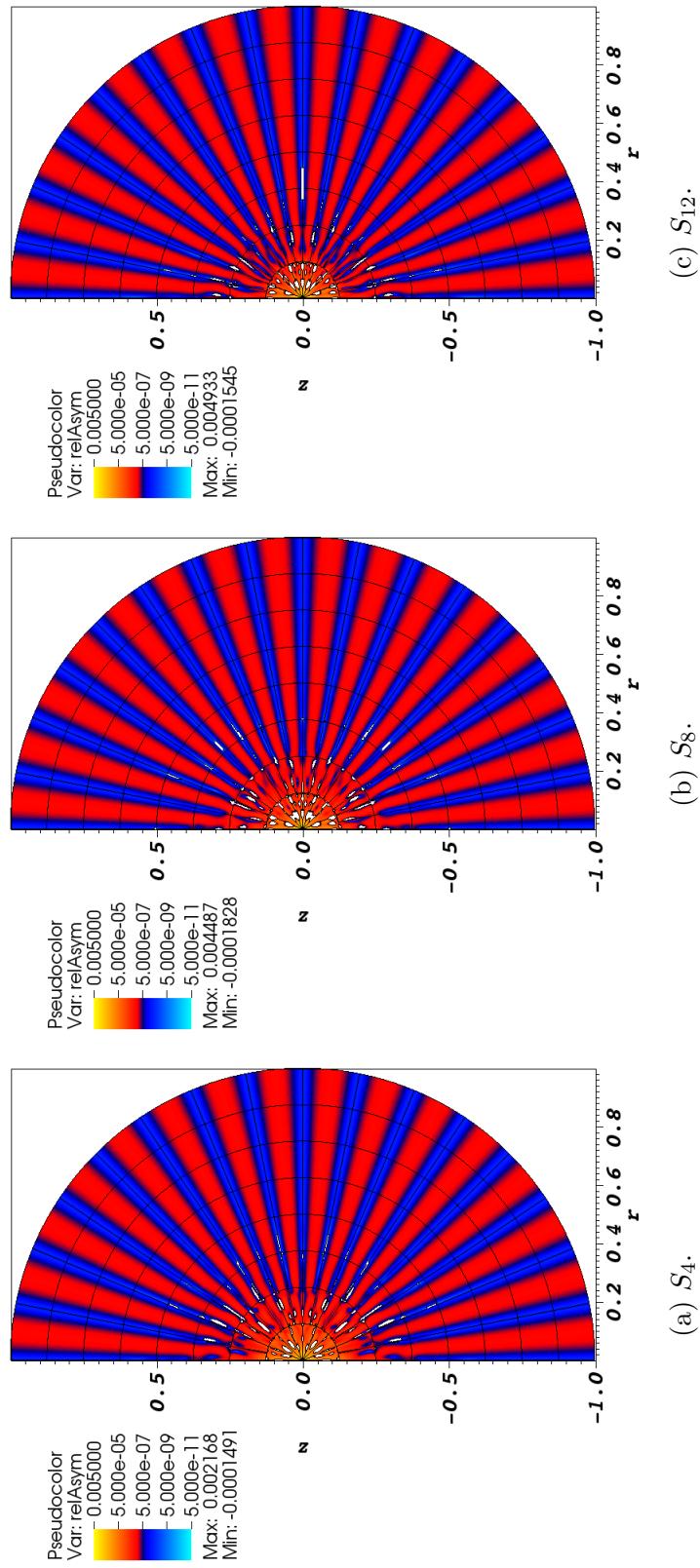


Figure 17: Relative asymmetry for 4<sup>th</sup>-order finite elements on a 2<sup>nd</sup>-order mesh for given order of level-symmetric angular quadrature.

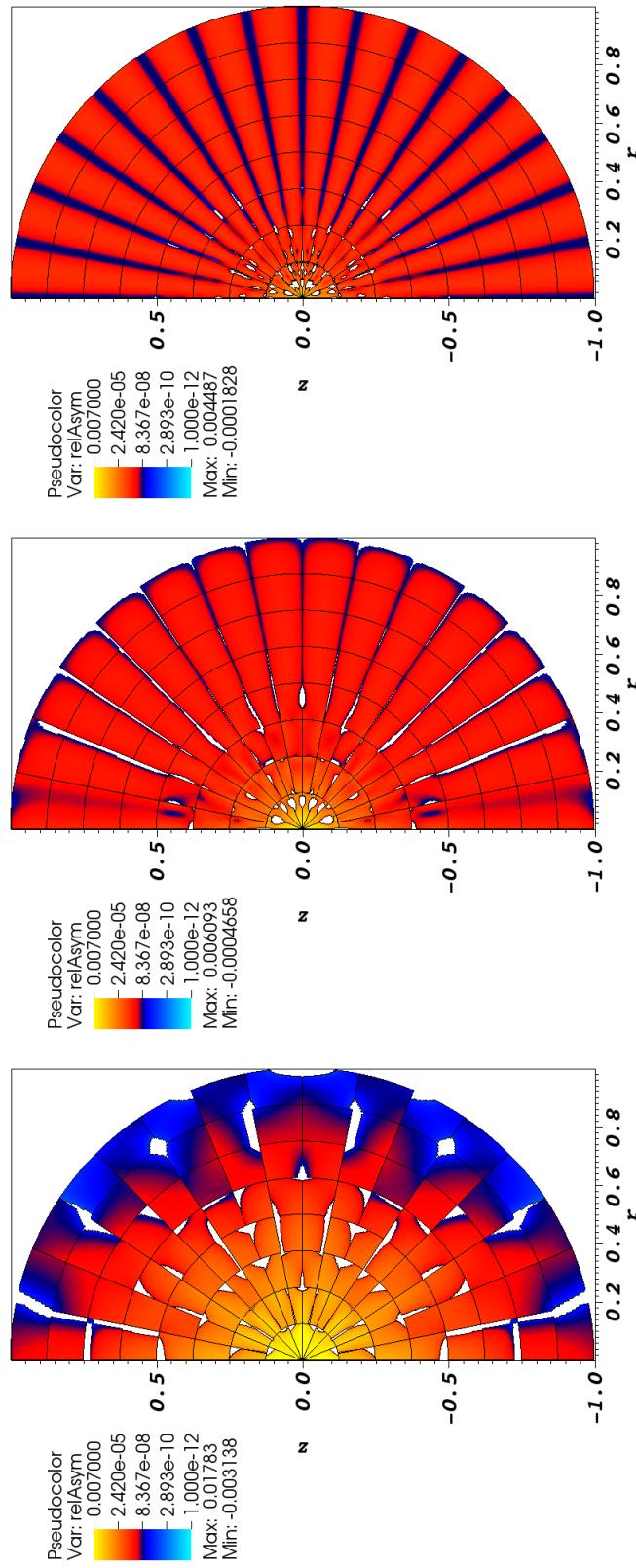


Figure 18: Relative asymmetry for  $p = \{1, 2, 4\}$  finite elements on a 2<sup>nd</sup>-order mesh for  $S_8$  level-symmetric angular quadrature.

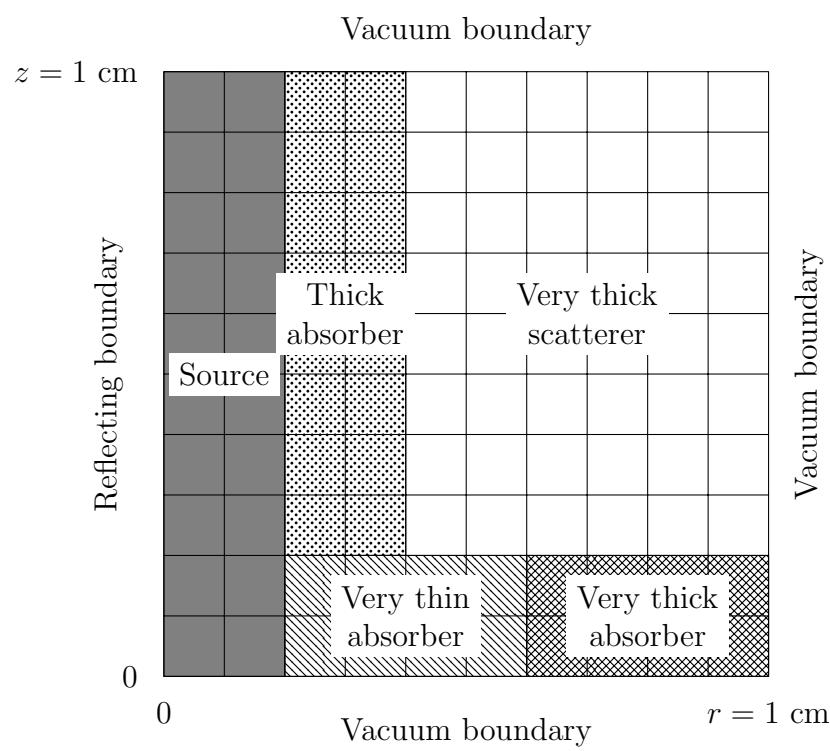
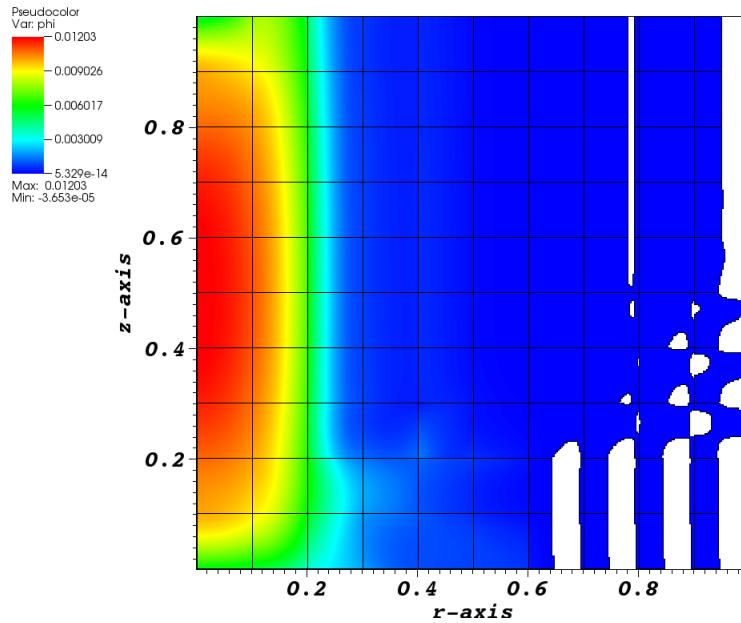
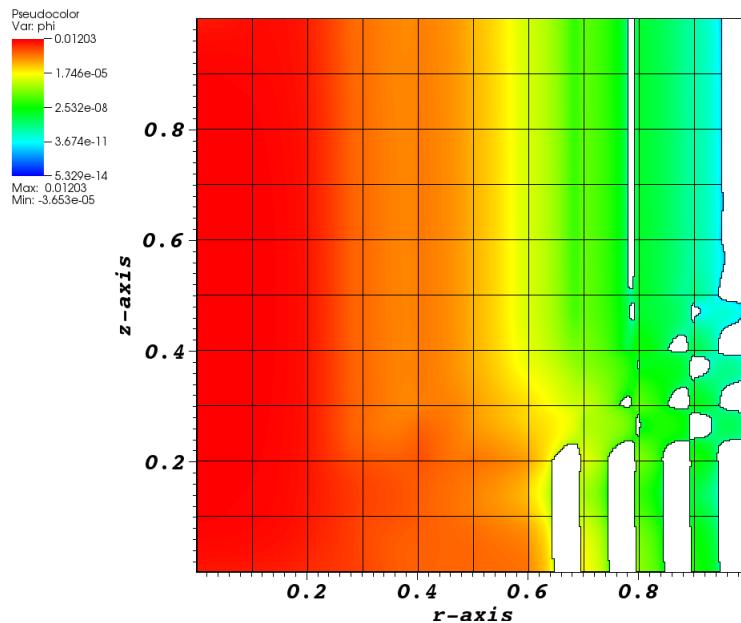


Figure 19: Material discontinuity stress test with MIP DSA problem geometry; materials defined in Table 2.



(a) Scalar flux.



(b) Log of scalar flux.

Figure 20: Solution to multi-material stress test. White regions indicate negative scalar fluxes. This was solved without DSA and was only allowed 10,000 source iterations to complete.

## 6 Conclusions

### 6.1 Future Work

Future work beyond the scope of these research objectives could include methods for solving the system of equations. Currently we simultaneously solve for every degree of freedom in the problem. This limits the overall number of unknowns that we can accommodate because they all get stored in memory. However, if we solved individual mesh cells and systematically “swept through the mesh” (solve the sparse system of equations for each quadrature direction, in parallel or sequentially), we eliminate this limitation. This allows for cycles (discussed in Section ??) in the mesh, which require careful handling.

Negative energy densities, observed in some of the test problems above, may contribute to negative mass densities in the equations of state of multiphysics problems. This must be addressed. Mentioned previously are lumping the matrices that constitute the bilinear form or perform a negative energy density fix up. The latter results in a nonlinear system of equations, which may be satisfactory given the TRT equations are already nonlinear. Oscillations could be reduced by using lower order elements in susceptible regions. Investigating an adaptive element could prove beneficial.

Coupling the TRT equations with the hydrodynamics equations is a logical progression toward modeling a realistic problem. Understanding the coupling between equations may be challenging given the variety of physical phenomena in these problems. The increase in the number of degrees of freedom upon this coupling begs for an increase in efficiency. An investigation into increased parallelization is warranted along with solution methods such as sweeping through the mesh.

[Mesh sweeping, breaking cycles.](#)

## References

- [1] E. E. Lewis and W. F. Miller, Jr. *Computational Methods of Neutron Transport*. American Nuclear Society, 1993.
- [2] John Castor. *Radiation Hydrodynamics*. Cambridge, 2007.
- [3] R. Paul Drake. High-energy-density physics. *Physics Today*, 63(3):8–9, June 2010.
- [4] Veselin A. Dobrev, Tzanio V. Kolev, and Robert N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal of Scientific Computing*, 34(5):B606 – B641, 2012.
- [5] G. Scovazzi, E. Love, and M. J. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Computer Methods in Applied Mechanics and Engineering*, 197:1056–1079, 2008.
- [6] V. A. Dobrev, T. E. Ellis, Tz. V. Kolev, and R. N. Rieben. Curvilinear finite elements for lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 2010.
- [7] MFEM: Modular finite element methods. [mfem.org](http://mfem.org), 2015.
- [8] K. D. Lathrop and B. G. Carlson. Discrete ordinates angular quadrature of the neutron transport equation. Technical report, Los Alamos Scientific Laboratory of the University of California, 1965.
- [9] Marvin L. Adams, Todd A. Wareing, and Wallace F. Walters. Characteristic methods in thick diffusive problems. *Nuclear Science and Engineering*, 130:18–46, 1998.
- [10] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, 1973.
- [11] Edward W. Larsen and J. E. Morel. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes II. *Journal of Computational Physics*, 83:212–236, 1989.
- [12] P. Lasaint and P. A. Raviart. On a finite element method for solving the neutron transport equation. In Carl de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–123. Academic Press, 1974.

- [13] Edward W. Larsen and Warren F. Miller, Jr. Convergence rates of spatial difference equations for the discrete-ordinates neutron transport equations in slab geometry. *Nuclear Science and Engineering*, 1980.
- [14] Steven Hamilton, Michele Benzi, and Jim Warsa. Negative flux fixups in discontinuous finite element  $S_N$  transport. *International Conference on Mathematics, Computational Methods & Reactor Physics*, May 2009.
- [15] Marvin L. Adams. Discontinuous finite element transport solutions in thick diffusive problems. *Nuclear Science and Engineering*, 137(3):298–333, 2001.
- [16] Todd A. Wareing, John M. McGhee, and Jim E. Morel. ATTILA: A three-dimensional, unstructured tetrahedral mesh discrete ordinates transport code. *Transactions of the American Nuclear Society*, 75:146–147, 1996.
- [17] J. M. McGhee and T.A. Wareing. *Attila Version 2: User's Manual*. Los Alamos National Laboratory, 2000.
- [18] Teresa S. Bailey, Marvin L. Adams, Jae H. Chang, and James S. Warsa. A piecewise linear discontinuous finite element spatial discretization of the transport equation in 2D cylindrical geometry. *International Conference on Mathematics, Computational Methods & Reactor Physics*, 2009.
- [19] Yaqi Wang and Jean C. Ragusa. Diffusion synthetic acceleration for high-order discontinuous finite element  $S_N$  transport schemes and application to locally refined unstructured meshes. *Nuclear Science and Engineering*, 166:145–166, 2010.
- [20] Yaqi Wang and Jean C. Ragusa. A high-order discontinuous Galerkin method for the  $S_N$  transport equations on 2D unstructured triangular meshes. *Annals of Nuclear Energy*, 36:931–939, 2009.
- [21] Yaqi Wang and Jean C. Ragusa. On the convergence of DGFEM applied to the discrete ordinates transport equation for structured and unstructured triangular meshes. *Nuclear Science and Engineering*, 163:56–72, 2009.
- [22] Yaqi Wang. *Adaptive Mesh Refinement Solution Techniques for the Multigroup  $S_N$  Transport Equation Using a Higher-Order Discontinuous Finite Element Method*. PhD thesis, Texas A&M University, 2009.
- [23] Douglas N. Woods, Thomas A. Brunner, and Todd S. Palmer. High order finite elements  $S_N$  transport in X-Y geometry on meshes with curved surfaces. *Transactions of the American Nuclear Society*, 114:377 – 380, June 2016.
- [24] Douglas N. Woods. High order finite elements  $S_N$  transport in X-Y geometry on meshes with curved surfaces in the thick diffusion limit. Master's thesis, Oregon State University, February 2016.

- [25] Peter G. Maginot, Jim E. Morel, and Jean C. Ragusa. A non-negative moment-preserving spatial discretization scheme for the linearized Boltzmann transport equation in 1-D and 2-D Cartesian geometries. *Journal of Computational Physics*, 231:6801–6826, 2012.
- [26] Peter G. Maginot, Jean C. Ragusa, and Jim E. Morel. Lumping techniques for DFEM  $S_N$  transport in slab geometry. *Nuclear Science and Engineering*, 2015.
- [27] Thomas A. Brunner. Preserving positivity of solutions to the diffusion equation for higher-order finite elements in under resolved regions. In *ANS M&C - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, April 2015.
- [28] Jim E. Morel, Alejandro Gonzales-Aller, and James S. Warsa. A lumped linear-discontinuous spatial discretization scheme for triangular-mesh  $S_N$  calculations in r-z geometry. *Nuclear Science and Engineering*, 155:168–178, 2007.
- [29] Hiromi G. Stone and Marvin L. Adams. A piecewise linear finite element basis with application to particle transport. *Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew*, 2003.
- [30] Juan Cheng and Chi-Wang Shu. A third order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations. *Communications in Computational Physics*, 4(5):1008–1024, 2008.
- [31] Veselin A. Dobrev, Truman E. Ellis, Tzanio V. Kolev, and Robert N. Rieben. High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics. *Computers & Fluids*, 2012.
- [32] Edward W. Larsen, J. E. Morel, and Warren F. Miller, Jr. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *Journal of Computational Physics*, 69:283–324, 1987.
- [33] F. Malvgi and G. C. Pomraning. Initial and boundary conditions for diffusive linear transport problems. *Journal of Mathematical Physics*, 32(3):805–820, 1991.
- [34] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. Wiley-Interscience, 1976.
- [35] Edward W. Larsen. The asymptotic diffusion limit of discretized transport problems. *Nuclear Science and Engineering*, 112:336–346, 1992.
- [36] Christoph Börgers, Edward W. Larsen, and Marvin L. Adams. The asymptotic diffusion limit of a linear discontinuous discretization of a two-dimensional linear transport equation. *Journal of Computational Physics*, 98(2):285–300, 1992.

- [37] Jim E. Morel and James S. Warsa. An  $S_N$  spatial discretization scheme for tetrahedral meshes. *Nuclear Science and Engineering*, 151:157–166, 2005.
- [38] Marvin L. Adams. Discontinuous finite-element transport solutions in the thick diffusion limit in Cartesian geometry. *The American Nuclear Society International Topical Meeting*, 1991.
- [39] Todd S. Palmer and Marvin L. Adams. Curvilinear geometry transport discretization in the "thick" diffusion limit. *Proceedings of the International Conference Math. Methods & Supercomputing in Nuc. Appl.*, 1993.
- [40] J. E. Morel and J. S. Warsa. A lumped bilinear-discontinuous  $S_N$  spatial discretization for R-Z quadrilateral meshes. *Transactions of the American Nuclear Society*, 95:873, 2006.
- [41] Teresa Bailey. *A Piecewise Linear Discontinuous Finite Element Method Applied to the RZ and XYZ Transport Equations*. PhD thesis, Texas A&M University, 2008.
- [42] T. S. Bailey, J. H. Chang, J. S. Warsa, and M. L. Adams. A piecewise bi-linear discontinuous finite element spatial discretization of the  $S_N$  transport equation. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.
- [43] Edward W. Larsen. Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. part I: Theory. *Nuclear Science and Engineering*, 82(1):47–63, 1982.
- [44] Todd A. Wareing, Edward W. Larsen, and Marvin L. Adams. Diffusion accelerated discontinuous finite element schemes for the  $S_N$  equations in slab and X,Y geometries. 1991.
- [45] James S. Warsa, Todd A. Wareing, and Jim E. Morel. Fully consistent diffusion synthetic acceleration of linear discontinuous  $S_N$  transport discretizations on unstructured tetrahedral meshes. *Nuclear Science and Engineering*, 141:236–251, 2002.
- [46] Marvin L. Adams and Edward W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40(1):3–159, 2002.
- [47] Donald R. McCoy and Edward W. Larsen. Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. part II: Numerical results. *Nuclear Science and Engineering*, 82(1):64–70, 1982.

- [48] Marvin L. Adams and William R. Martin. Diffusion synthetic acceleration of discontinuous finite element transport iterations. *Nuclear Science and Engineering*, 111:145–167, 1992.
- [49] Bruno Turcksin and Jean C. Ragusa. Discontinuous diffusion synthetic acceleration for  $S_N$  transport on 2D arbitrary polygonal meshes. *Journal of Computational Physics*, 274:356–369, 2014.
- [50] Bruno Turcksin and Jean C. Ragusa. A diffusion synthetic acceleration scheme for rectangular geometries based on bilinear discontinuous finite elements. In *International Conference on Mathematics and Computational Methods, Applied to Nuclear Science and Engineering*. Sun Valley, Idaho, 2013.
- [51] Guido Kanschat. *Discontinuous Galerkin Methods for Viscous Incompressible Flow*. Deutscher Universitäts-Verlag, 2007.
- [52] Todd S. Palmer. *Curvilinear Geometry Transport Discretization in Thick Diffusive Regions*. PhD thesis, University of Michigan, 1993.
- [53] C. Lingus. Analytical test case's for neutron and radiation transport codes. In *Second Conference on Transport Theory*, pages 655–659, Los Alamos, New Mexico, January 1971. United States Atomic Energy Commission - Division of Technical Information.
- [54] Thomas A. Brunner, Tzanio V. Kolev, and Teresa S. Bailey. Preserving spherical symmetry in axisymmetric coordinates for diffusion problems. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2013.
- [55] F. Chaland and G. Samba. Discrete ordinates method for the transport equation preserving one-dimensional spherical symmetry in two-dimensional cylindrical geometry. *Nuclear Science and Engineering*, 182:417–434, 2016.
- [56] J. E. Morel and G. R. Montry. Analysis and elimination of the discrete-ordinates flux dip. *Transport Theory and Statistical Physics*, 13(5):615–633, 1984.
- [57] J. S. Warsa and A. K. Prinja. Differential quadrature approximation of the angular derivative terms in the curvilinear-coordinates  $S_N$  equations. *Transactions of the American Nuclear Society*, 111:689–692, 2014.
- [58] Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006.
- [59] Timothy A. Davis. Algorithm 832: Umfpack v4.3 — an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, June 2004.

- [60] Hank Childs, Eric Brugger, Brad Whitlock, Jeremy Meredith, Sean Ahern, David Pugmire, Kathleen Biagas, Mark Miller, Cyrus Harrison, Gunther H. Weber, Hari Krishnan, Thomas Fogal, Allen Sanderson, Christoph Garth, E. Wes Bethel, David Camp, Oliver Rübel, Marc Durant, Jean M. Favre, and Paulr Navrátil. Visit: An end-user tool for visualizing and analyzing very large data. In *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

## A Implementation in MFEM

The open source finite element library Modular Finite Elements Method (MFEM<sup>3</sup>) [7] was used to create the system of linear equations to be solved by a linear algebra solver. The user chooses various parameters to create the system of equations and passes them into MFEM as arguments (i.e. the order of finite elements, the mesh, the number of times to refine the mesh, the order of the mesh, any mesh transformations, the linear algebra solver method, the source iteration convergence criteria, the maximum number of source iterations to perform, the initial guess for the scalar flux, and, in diffusion limit problems, the scaling factor to be applied).

MFEM creates the matrices and a linear solver computes the angular flux. This research utilizes the serial version of MFEM (as opposed to the parallel version where the spatial domain is solved in parallel) and the direct solver UMFPack<sup>4</sup> [58, 59] to solve the equations using a *LU* decomposition. It is common for transport solvers to solve the local system of equations for an individual spatial cell and sweep through the problem domain, propagating information from one cell to the next. Instead, we use MFEM to create the system of equations for the entire problem domain and solve for all of the unknowns in all cells simultaneously. This is more computationally intensive because all of the degrees of freedom in the entire problem domain must be solved for simultaneously. However, mesh zone complications, such as cycles, may be present in the mesh. We avoid having to “break the cycles” by solving for the entire problem simultaneously.

---

<sup>3</sup>[mfem.org](http://mfem.org)

<sup>4</sup><http://faculty.cse.tamu.edu/davis/suitesparse.html>

## A.1 Transport Operators

Shown in Table 4 are the functions within MFEM that integrate and assemble the various components of the transport equation (Equation ??) to the linear algebraic system. The functions are displayed along with the general form of their equation and their translation to the applicable component of the transport equation. The last two entries of Table 4 are the interior boundaries using the upstream values (no. 5) and the problem boundary (no. 6). Several of the MFEM equations have coefficients,  $\alpha$  and  $\beta$ , that are required input. For item number 1, using  $\alpha = 1$  sets the MFEM equation equal to the discretized equation. Similarly, for item number 5 using  $\alpha = -1$  and  $\beta = 1/2$ ,

$$\alpha \int_{\partial\mathbb{V}} \boldsymbol{\Omega} \cdot (-\hat{n}) \{ \psi \} w + \beta \int_{\partial\mathbb{V}} |\boldsymbol{\Omega} \cdot (-\hat{n})| [\![ \psi ]\!] w = \int_{\partial\mathbb{V}_k} (\boldsymbol{\Omega} \cdot \hat{n}) \psi_k w_{k,i} \quad (85)$$

where  $\{ \psi \} = 1/2 (\psi_u + \psi_k)$  and  $[\![ \psi ]\!] = \psi_u - \psi_k$ , where  $\psi_u$  is the upwind angular flux and  $\psi_k$  is the angular flux in cell  $k$ .

$$\begin{aligned} -1 \int_{\partial\mathbb{V}} [\boldsymbol{\Omega} \cdot (-\hat{n})] \left[ \frac{1}{2} (\psi_u + \psi_k) \right] w + \frac{1}{2} \int_{\partial\mathbb{V}} |\boldsymbol{\Omega} \cdot (-\hat{n})| (\psi_u - \psi_k) w \\ = \int_{\partial\mathbb{V}_k} (\boldsymbol{\Omega} \cdot \hat{n}) \psi_k w_{k,i} \end{aligned} \quad (86)$$

$$\frac{1}{2} \int_{\partial\mathbb{V}} (\boldsymbol{\Omega} \cdot \hat{n}) (\psi_u + \psi_k) w + \frac{1}{2} \int_{\partial\mathbb{V}} |\boldsymbol{\Omega} \cdot (-\hat{n})| (\psi_u - \psi_k) w = \int_{\partial\mathbb{V}_k} (\boldsymbol{\Omega} \cdot \hat{n}) \psi_k w_{k,i} \quad (87)$$

For  $\Omega \cdot \hat{n} < 0$  (incident to cell  $k$ ),

$$\frac{1}{2} \int_{\partial\mathbb{V}} (\Omega \cdot \hat{n}) (\psi_u + \psi_k) w - \frac{1}{2} \int_{\partial\mathbb{V}} (\Omega \cdot \hat{n}) (\psi_u - \psi_k) w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_k w_{k,i} \quad (88)$$

$$\frac{1}{2} \int_{\partial\mathbb{V}} (\Omega \cdot \hat{n}) \psi_k w + \frac{1}{2} \int_{\partial\mathbb{V}} (\Omega \cdot \hat{n}) \psi_k w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_k w_{k,i} \quad (89)$$

The normal vector  $\hat{n}$  in MFEM is outward of the upwind mesh surface so a negative was applied to the normal vector to convert it to be the outward normal of the surface of cell  $k$  like it has been previously defined in this thesis. Similarly, for item number 6,  $\alpha = -1$  and  $\beta = -1/2$ ,

$$\frac{\alpha}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} \Omega \cdot (-\hat{n}) w - \beta \int_{\partial\mathbb{V}} \psi_{\text{inc}} |\Omega \cdot (-\hat{n})| w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_{\text{inc},k} w_{k,i} \quad (90)$$

$$-\frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} \Omega \cdot (-\hat{n}) w + \frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} |\Omega \cdot (-\hat{n})| w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_{\text{inc},k} w_{k,i} \quad (91)$$

$$\frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} (\Omega \cdot \hat{n}) w + \frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} [\Omega \cdot (-\hat{n})] w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_{\text{inc},k} w_{k,i} \quad (92)$$

$$\frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} (\Omega \cdot \hat{n}) w + \frac{1}{2} \int_{\partial\mathbb{V}} \psi_{\text{inc}} (\Omega \cdot \hat{n}) w = \int_{\partial\mathbb{V}_k} (\Omega \cdot \hat{n}) \psi_{\text{inc},k} w_{k,i} \quad (93)$$

MFEM automatically determines the degree of numerical integration to integrate each of the integrals of Table 4. These default integration orders are shown in Table 5. It was discovered that integrating all of the terms consistently was important for numeric conservation. For simplicity, each of the integration orders were set to the largest of the default integration orders. Table 5 shows that the integration order is the same for all of the integrators except DomainLFIntegrator, which is the largest integration order only if  $p = 0$  (piecewise constant). While the results presented in this thesis do not consider the circumstance of piecewise constant finite elements, this integration order was included in the code for future use.

MFEM is equipped to visualize data using various tools requiring additional user

input. The images presented in this thesis were produced with VisIt, an open source visualization analysis tool [60].

no.	Discretized Equation	MFEM Equation	MIFEM Integrator Function
1	$(\boldsymbol{\Omega} \cdot \nabla \psi_j, v_i)_{\mathbb{D}_k}$	$(\alpha \boldsymbol{\Omega} \cdot \nabla \psi, v)_{\mathbb{D}_k}$	ConvectionIntegrator( $\boldsymbol{\Omega}, \alpha$ )
2	$(\sigma_t \psi_j, v_i)_{\mathbb{D}_k}$	$(\sigma_t \psi, v)_{\mathbb{D}_k}$	MassIntegrator( $\sigma_t$ )
3	$(\sigma_s \phi, v_i)_{\mathbb{D}_k}$	$(\varphi, v)_{\mathbb{D}_k}$	DomainLFIIntegrator( $\varphi$ )
4	$(S_0, v_i)_{\mathbb{D}_k}$	$(S_0, v)_{\mathbb{D}_k}$	DomainLFIIntegrator( $S_0$ )
5	$(\boldsymbol{\Omega} \cdot \hat{n} \psi_j, v_i)_{\partial \mathbb{D}_k}$	$\alpha (\boldsymbol{\Omega} \cdot \hat{n} \psi, v)_{\partial \mathbb{D}_k}$ $+ \beta ( \boldsymbol{\Omega} \cdot \hat{n}  \psi, v)_{\partial \mathbb{D}_k}$	DGTraceIntegrator( $\boldsymbol{\Omega}, \alpha, \beta$ )
6	$(\boldsymbol{\Omega} \cdot \hat{n} \psi_{inc}, v_i)_{\partial \mathbb{D}_k}$	$\frac{\alpha}{2} (\psi_{inc} \boldsymbol{\Omega} \cdot \hat{n}, v)_{\partial \mathbb{D}_k}$ $- \beta (\psi_{inc}  \boldsymbol{\Omega} \cdot \hat{n} , v)_{\partial \mathbb{D}_k}$	BoundaryFlowIntegrator( $\psi_{inc}, \boldsymbol{\Omega}, \alpha, \beta$ )

Table 3: MFEM PDE function calls where the arguments have been dropped (see Equations ?? and ?? for these details).

no.	Discretized Equation	MFEM Equation	MFEM Integrator Function
	$(r \Omega \cdot \nabla \psi_j, v_i)_{\mathbb{D}_k}$	$(r \alpha \Omega \cdot \nabla \psi, v)_{\mathbb{D}_k}$	RZConvectionIntegrator( $\Omega, \alpha$ ) <sup>†</sup>
	$(r \Omega \cdot \hat{n} \psi_j, v_i)_{\partial \mathbb{D}_k}$	$\alpha (r \Omega \cdot \hat{n} \psi, v)_{\partial \mathbb{D}_k}$ $+ \beta (r  \Omega \cdot \hat{n}  \psi, v)_{\partial \mathbb{D}_k}$	RZDGTraceIntegrator( $\Omega, \alpha, \beta$ ) <sup>†</sup>
	$(\mu_{n,m} \psi_j, v_i)_{\mathbb{D}_k}$		MassIntegrator( $\mu_{n,m}$ )
	$\left( \frac{\alpha_{m+1/2,n}}{\tau_{n,m} w_{n,m}} \psi_j, v_i \right)_{\mathbb{D}_k}$		MassIntegrator( $\frac{\alpha_{m+1/2,n}}{\tau_{n,m} w_{n,m}}$ )
	$(r \sigma_t \psi_j, v_i)_{\mathbb{D}_k}$	$(r \sigma_t \psi, v)_{\mathbb{D}_k}$	RZMassIntegrator( $\sigma_t$ ) <sup>†</sup>
	$\frac{1}{4\pi} (r \sigma_s \phi, v_i)_{\mathbb{D}_k}$	$(r \varphi, v)_{\mathbb{D}_k}$	RZDomainLFIIntegrator( $\frac{1}{4\pi} \sigma_s \phi$ ) <sup>†</sup>
	$\frac{1}{4\pi} (r S_0, v_i)_{\mathbb{D}_k}$	$(r S_0, v)_{\mathbb{D}_k}$	RZDomainLFIIntegrator( $\frac{S_0}{4\pi}$ ) <sup>†</sup>
	$(r \Omega \cdot \hat{n} \psi_{inc}, v_i)_{\partial \mathbb{D}_k}$	$\frac{\alpha}{2} (r \psi_{inc} \Omega \cdot \hat{n}, v)_{\partial \mathbb{D}_k}$ $- \beta (r \psi_{inc}  \Omega \cdot \hat{n} , v)_{\partial \mathbb{D}_k}$	RZBoundaryFlowIntegrator( $\psi_{inc}, \Omega, \alpha, \beta$ ) <sup>†</sup>

<sup>†</sup> Modified MFEM operator

Table 4: MFEM PDE function calls where the arguments have been dropped (see Equations ?? and ?? for these details).

Table 5: MFEM default integration orders for transport operators. The notation for the finite element order is  $p$ , mesh order is  $m$ , and problem dimension is  $d$ .

MFEM Integrator	Default Integration Order
DGTraceIntegrator	$m \cdot d + 2 \cdot p - 1$
ConvectionIntegrator	$m \cdot d + 2 \cdot p - 1$
MassIntegrator	$m \cdot d + 2 \cdot p - 1$
DomainLFIntegrator	$2 \cdot m$
BoundaryFlowIntegrator	$m \cdot d + 2 \cdot p - 1$

### A.1.1 MIP DSA Operators

There are some specific function calls to MFEM for the diffusion equation that are listed in Table 6. Items 3 and 4 have a  $\sigma_D$  value that controls the DG method to be used, where  $\sigma_D = -1$  is for the symmetric interior penalty method. Item 6 is for the Robin boundary condition described by Equation 23. Item 4 is the function that will need to be modified to adapt Methods 1 and 2 (this function is not used for Method 3).

no.	FEM Equations 14 & 15	MFEM Equation	User Input
1	$(\sigma_a \phi, w)_{\nabla}$	$(\sigma_a \phi, w)_{\nabla}$	MassIntegrator( $\sigma_a$ )
2	$(D \nabla \phi, \nabla w)_{\nabla}$	$(D \nabla \phi, \nabla w)_{\nabla}$	DiffusionIntegrator( $D$ )
3	$(\{\{ D \partial_n \phi \}\}, [w])_{\partial \mathbb{V}^i}$ + $([\phi], \{\{ D \partial_n w \}\})_{\partial \mathbb{V}^i}$ + $(\kappa_e [\phi], [w])_{\partial \mathbb{V}^i}$	$(\{\{ D \nabla \phi \cdot \hat{n} \}\}, [w])_{\partial \mathbb{V}^i}$ + $\sigma_D ([\phi], \{\{ D \nabla w \cdot \hat{n} \}\})_{\partial \mathbb{V}^i}$ + $\kappa \left( \left\{ \frac{D}{h_{\perp}} \right\} [\phi], [w] \right)_{\partial \mathbb{V}^i}$	DGDiffusionIntegrator( $D, \sigma_D, \kappa$ )
4	$(\{\{ D \partial_n \phi \}\}, [w])_{\partial \mathbb{V}^d}$ - $\frac{1}{2} ([\phi], \{\{ D \partial_n w \}\})_{\partial \mathbb{V}^d}$ - $\frac{1}{2} (\kappa_e [\phi], [w])_{\partial \mathbb{V}^d}$	$(\{\{ D \nabla \phi \cdot \hat{n} \}\}, [w])_{\partial \mathbb{V}^d}$ + $\sigma_D ([\phi], \{\{ D \nabla w \cdot \hat{n} \}\})_{\partial \mathbb{V}^d}$ + $\kappa \left( \left\{ \frac{D}{h_{\perp}} \right\} [\phi], [w] \right)_{\partial \mathbb{V}^d}$	DGDiffusionIntegrator( $D, \sigma_D, \kappa$ )
5	$(Q_0, w)_{\nabla}$	$(Q_0, w)_{\nabla}$	DomainLFIntegrator( $Q_0$ )
6	$(\frac{1}{2} \phi, w)_{\partial \mathbb{V}^d}$	$(\frac{1}{2} \phi, w)_{\partial \mathbb{V}^d}$	BoundaryMassIntegrator( $\frac{1}{2}$ )

<sup>†</sup>  $\partial \mathbb{V}^i$  denotes an internal edge  
<sup>‡</sup>  $\partial \mathbb{V}^d$  denotes a boundary edge

Table 6: MFEM diffusion equation function calls.

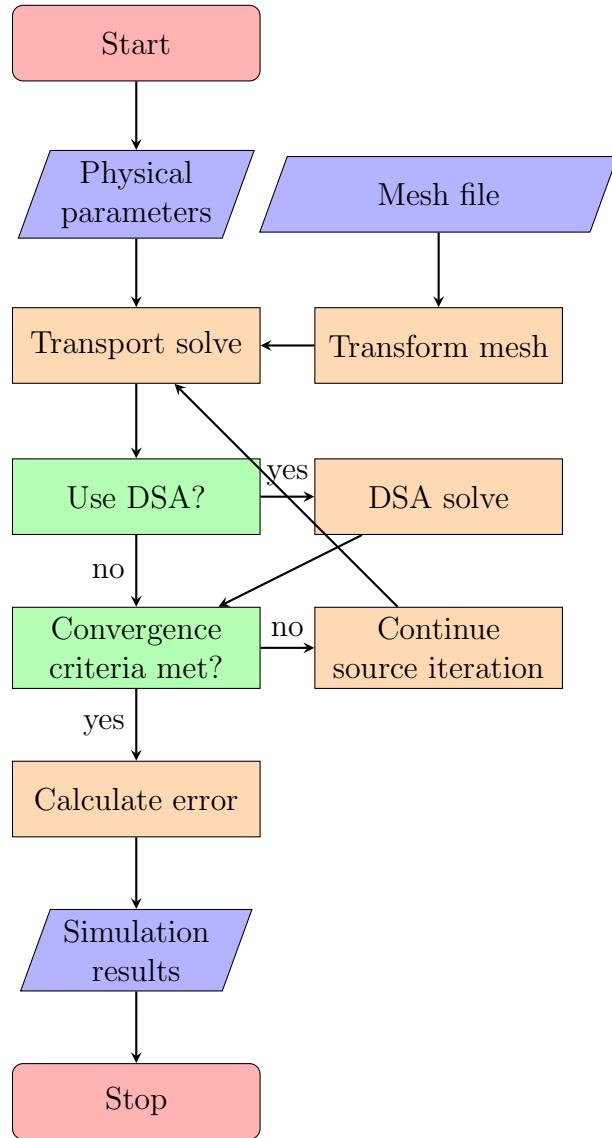


Figure 21: Flow diagram for solution process.

## B Mesh Examples

## C Mathematica Scripts

### C.1 Mass Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```
(*Mass matrix entry for RZ BLD Gauss-Legendre on quadrilaterals*)
\

Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, mr3, mz0, mz1, \
mz2, mz3]

(*Define physical integration point coordinates to compare to MFEM \
output:r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0) ;
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0) ;
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0) ;
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)

(*These make the mesh \
rectangular:*)
z2 := z3
z1 := z0
r2 := r1
r3 := r0

(*Quadrilateral transformation back to reference square-more general \
than rectangle transofrmation-but NOT curved mesh*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 - \
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 - \
z3) rho kappa + z0
```

```

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)

\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1

mr3 := mr0

mz1 := mz0

mz2 := mz3

mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0

mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0

a = {mr[rho, kappa], mz[rho, kappa]};

b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +

```

```

3/(3 + Sqrt[3])*kappa - 0.5

BLD3[rho_, kappa_] :=

3*rho*kappa - 3/(3 + Sqrt[3])*rho -

3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)

BLD4[rho_, kappa_] := -3*rho*kappa +

3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 0.5

(*Check for orthogonality.*)

BLD1[r0, z0]*BLD2[r0, z0];

(*The r variable gets transformed back to the reference element*)

M =

Integrate[

mr[rho, kappa]*BLD1[rho, kappa]*BLD1[rho, kappa]*j, {rho, 0,
1}, {kappa, 0, 1}]

mr0 = 0.25;

mz0 = 0.25;

mr1 = 0.5;

mz3 = 0.5;

```

M

## C.2 Angular Redistribution Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*Angular Redistribution matrix for RZ BLD Gauss-Legendre on \
quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, \

```

```

mr3, mz0, mz1, mz2, mz3]

(*Define physical integration point coordinates to compare to MFEM \
output:r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)

(*These make the mesh \
rectangular:*)
z2 := z3
z1 := z0
r2 := r1
r3 := r0

(*Quadrilateral transformation back to reference square-more general \
than rectangle transofrmation-but NOT curved mesh*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)
\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1

```

```

mr3 := mr0
mz1 := mz0
mz2 := mz3
mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0
mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0
a = {mr[rho, kappa], mz[rho, kappa]};
b = {rho, kappa};
(*Determinant of the Jacobian-for determining area of the physical \
element.*)
j = Det[D[a, {b}]];
(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)
BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)
BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2
BLD3[rho_, kappa_] :=
3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)
BLD4[rho_, kappa_] := -3*rho*kappa +
3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2
M = Integrate[

```

```

BLD1[rho, kappa]*BLD1[rho, kappa]*j, {rho, 0, 1}, {kappa, 0, 1}]

mr0 = 0;
mz0 = 0;
mr1 = 0.25;
mz3 = 0.25;

thisalpha = 0.350021175;
w = 0.523598775598298;
M*thisalpha

```

### C.3 r-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*r-Leakage matrix for RZ BLD Gauss-Legendre on \
quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, \
mr3, mz0, mz1, mz2, mz3, mu, rho, kappa]

(*Define physical integration point coordinates to compare to MFEM \
output: *)

r0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
r1 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z3 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);

(*Use to define rectangular mesh:*)

r2 = r1;
r3 = r0;

```

```

z1 = z0;
z2 = z3;

(*Transform arbitrary quadrilateral to reference square*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)

\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1
mr3 := mr0
mz1 := mz0
mz2 := mz3

mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0
mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0
a = {mr[rho, kappa], mz[rho, kappa]};
b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];

```

```

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=

3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2

BLD3[rho_, kappa_] :=

3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)

BLD4[rho_, kappa_] := -3*rho*kappa +
3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2

(*The r value gets transformed back to the reference element*)

Kr = \
-mu*Integrate[

BLD1[rho, kappa]*D[BLD1[rho, kappa], rho]*j, {rho, 0, 1}, {kappa,
0, 1}]

mr0 = 0.0;

mr1 = 0.25;

mz0 = 0.0;

mz3 = 0.25;

mu = -0.495004692;

BLD1[r0, z0]

rho = r1;

D[BLD1[rho, kappa], kappa]

```

Kr

#### C.4 z-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```
(*z-Leakage matrix for RZ BLD Gauss-Legendre on \
quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, \
mr3, mz0, mz1, mz2, mz3, xi, rho, kappa]

(*Define physical integration point coordinates to compare to MFEM \
output:*)

r0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
r1 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z3 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);

(*Use to define rectangular mesh:*)

r2 = r1;
r3 = r0;
z1 = z0;
z2 = z3;

(*Transform arbitrary quadrilateral to reference square*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 - 
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 - 
z3) rho kappa + z0
```

```

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)

\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1

mr3 := mr0

mz1 := mz0

mz2 := mz3

mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0

mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0

a = {mr[rho, kappa], mz[rho, kappa]};

b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2

```

```

BLD3[rho_, kappa_] :=  

  3*rho*kappa - 3/(3 + Sqrt[3])*rho -  

  3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)  

BLD4[rho_, kappa_] := -3*rho*kappa +  

  3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2  

(*The r value gets transformed back to the reference element*)  

Kz = \  

-xi*Integrate[  

  BLD1[rho, kappa]*D[BLD1[rho, kappa], kappa]*j, {rho, 0, 1}, {kappa,  

  0, 1}]  

mr0 = 0.0;  

mr1 = 0.25;  

mz0 = 0.0;  

mz3 = 0.25;  

xi = -0.868890301;  

D[BLD1[rho, kappa], kappa]  

xi*j  

Kz  

Kr + Kz

```

## C.5 z-Leakage Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*z-Leakage matrix for RZ BLD Gauss-Legendre on \  

quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, \  

mr3, mz0, mz1, mz2, mz3, xi, rho, kappa]

```

```

(*Define physical integration point coordinates to compare to MFEM \
output:*)

r0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
r1 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z0 = (1 + (-Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);
z3 = (1 + (Sqrt[1/3]))/(1 - (-1))*(0.25 - 0);

(*Use to define rectangular mesh:*)

r2 = r1;
r3 = r0;
z1 = z0;
z2 = z3;

(*Transform arbitrary quadrilateral to reference square*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)

\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1
mr3 := mr0
mz1 := mz0

```

```

mz2 := mz3

mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
    mr1 - mr3) rho kappa + mr0

mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
    mz1 - mz3) rho kappa + mz0

a = {mr[rho, kappa], mz[rho, kappa]};

b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2

BLD3[rho_, kappa_] :=
3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)

BLD4[rho_, kappa_] := -3*rho*kappa +
3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2

(*The r value gets transformed back to the reference element*)

Kz = \
-xi*Integrate[

```

```

BLD1[rho, kappa]*D[BLD1[rho, kappa], kappa]*j, {rho, 0, 1}, {kappa,
0, 1}]

mr0 = 0.0;

mr1 = 0.25;

mz0 = 0.0;

mz3 = 0.25;

xi = -0.868890301;

D[BLD1[rho, kappa], kappa]

xi*j

Kz

Kr + Kz

```

## C.6 r Surface-Integral Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*r surface-integral matrix for RZ BLD Gauss-Legendre on \
quadrilaterals*)

Clear[r0, r1, r2, r3, z0, z1, z2, z3]

(*Define physical integration point coordinates to compare to MFEM \
output:

r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)
(*Use to define rectangular \
mesh:*)

```

```

r2 = r1;
r3 = r0;
z1 = z0;
z2 = z3;

(*Transform arbitrary quadrilateral to reference square*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0
a = {r[rho, kappa], z[rho, kappa]};
b = {rho, kappa};
(*Determinant of the Jacobian-for determining area of the physical \
element.*)
j = Det[D[a, {b}]];
(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

```

```

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)
BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2
BLD3[rho_, kappa_] :=
3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)

```

```

BLD4[rho_, kappa_] := -3*rho*kappa +
3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2
(*The r value gets transformed back to the reference element*)

Lr =
Integrate[r1*BLD1[1, kappa]*BLD1[1, kappa]*j, {kappa, 0, 1}] -
Integrate[r0*BLD1[0, kappa]*BLD1[0, kappa]*j, {kappa, 0, 1}]
r0 = 0.0;
r1 = 0.25;
z0 = 0.0;
z3 = 0.25;

```

## C.7 z Surface-Integral Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*z surface-integral matrix for RZ BLD Gauss-Legendre on \
quadrilaterals*)

Clear[r0, r1, r2, r3, z0, z1, z2, z3]
(*Define physical integration point coordinates to compare to MFEM \
output:
r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)
(*Use to define rectangular \
mesh:*)
r2 = r1;

```

```

r3 = r0;
z1 = z0;
z2 = z3;

(*Transform arbitrary quadrilateral to reference square*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0
a = {r[rho, kappa], z[rho, kappa]};
b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];
(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)
BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2
BLD3[rho_, kappa_] :=
3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)
BLD4[rho_, kappa_] := -3*rho*kappa +

```

```

3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2
(*The r value gets transformed back to the reference element*)

Lz =
Integrate[r[rho, 1]*BLD1[rho, 1]*BLD1[rho, 1]*j, {rho, 0, 1}] -
Integrate[r[rho, 0]*BLD1[rho, 0]*BLD1[rho, 0]*j, {rho, 0, 1}]
r0 = 0.0;
r1 = 0.25;
z0 = 0.0;
z3 = 0.25;
Lz
Lr + Lz

```

## C.8 LinearForm Angular Redistribution Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```

(*LinearForm Angular Redistribution matrix for RZ BLD Gauss-Legendre \
on quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, \
mr2, mr3, mz0, mz1, mz2, mz3]
(*Define physical integration point coordinates to compare to MFEM \
output:r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)
(*These make the mesh \
rectangular:*)
z2 := z3

```

```

z1 := z0
r2 := r1
r3 := r0
(*Quadrilateral transformation back to reference square-more general \
than rectangle transofrmation-but NOT curved mesh*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 -
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 -
z3) rho kappa + z0

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)
\
(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1
mr3 := mr0
mz1 := mz0
mz2 := mz3
mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0
mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0
a = {mr[rho, kappa], mz[rho, kappa]};
b = {rho, kappa};

```

```

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]] ;

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +
3/(3 + Sqrt[3])*kappa - 1/2

BLD3[rho_, kappa_] :=
3*rho*kappa - 3/(3 + Sqrt[3])*rho -
3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)

BLD4[rho_, kappa_] := -3*rho*kappa +
3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2

M = Integrate[
BLD2[rho, kappa]*BLD2[rho, kappa]*j, {rho, 0, 1}, {kappa, 0, 1}]

mr0 = 0;
mz0 = 0;
mr1 = 0.25;
mz3 = 0.25;

alphatau = 0.845025867;
psiMinusHalf = 0.1591549;

```

M\*alphatau\*psiMinusHalf

## C.9 LinearForm Scattering/Fixed Source Matrix Entries for RZ BLD Gauss-Legendre on Quadrilaterals

```
(*LinearForm scattering/fixed sources for RZ BLD Gauss-Legendre on \
quadrilaterals*)Clear[r0, r1, r2, r3, z0, z1, z2, z3, mr0, mr1, mr2, \
mr3, mz0, mz1, mz2, mz3, Source]

(*Define physical integration point coordinates to compare to MFEM \
output:r0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
r1=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);
z0=(1+(-Sqrt[1/3]))/(1-(-1))*(0.25-0);
z3=(1+(Sqrt[1/3]))/(1-(-1))*(0.25-0);*)

(*These make the mesh \
rectangular:*)
z2 := z3
z1 := z0
r2 := r1
r3 := r0

(*Quadrilateral transformation back to reference square-more general \
than rectangle transofrmation-but NOT curved mesh*)

r[rho_, kappa_] := (r1 - r0) rho + (r3 - r0) kappa + (r2 + r0 - r1 - 
r3) rho kappa + r0
z[rho_, kappa_] := (z1 - z0) rho + (z3 - z0) kappa + (z2 + z0 - z1 - 
z3) rho kappa + z0
```

```

(*Mesh nodes are needed to conserve cell volume.Cannot define mr0=0 \
nor mz0=0 here.Perform integration and then set the values.*)

\

(*Comment out to get the general mesh element mr1=0.25;
mz3=0.25;*)

mr2 := mr1

mr3 := mr0

mz1 := mz0

mz2 := mz3

mr[rho_, kappa_] := (mr1 - mr0) rho + (mr3 - mr0) kappa + (mr2 + mr0 -
mr1 - mr3) rho kappa + mr0

mz[rho_, kappa_] := (mz1 - mz0) rho + (mz3 - mz0) kappa + (mz2 + mz0 -
mz1 - mz3) rho kappa + mz0

a = {mr[rho, kappa], mz[rho, kappa]};

b = {rho, kappa};

(*Determinant of the Jacobian-for determining area of the physical \
element.*)

j = Det[D[a, {b}]];

(*Basis functions.The coefficients are from MATLAB script \
"GaussLegendreQuadrilateral.m".*)

BLD1[rho_, kappa_] :=
3*rho*kappa - (3 + Sqrt[3])/2*rho - (3 + Sqrt[3])/2*
kappa + (1 + Sqrt[3]/2)

BLD2[rho_, kappa_] := -3*rho*kappa + (3 + Sqrt[3])/2*rho +

```

```
3/(3 + Sqrt[3])*kappa - 1/2
BLD3[rho_, kappa_] := 
  3*rho*kappa - 3/(3 + Sqrt[3])*rho -
  3/(3 + Sqrt[3])*kappa + (1 - Sqrt[3]/2)
BLD4[rho_, kappa_] := -3*rho*kappa +
  3/(3 + Sqrt[3])*rho + (3 + Sqrt[3])/2*kappa - 1/2
M = Integrate[
  Source*mr[rho, kappa]*BLD2[rho, kappa]*j, {rho, 0, 1}, {kappa, 0, 1}]
(*Source=0.7/(2*Pi);*)
Source = (0.3*1)/(2*Pi);
mr0 = 0;
mz0 = 0;
mr1 = 0.25;
mz3 = 0.25;
M
```