HIGHER-ORDER DGFEM TRANSPORT CALCULATIONS ON POLYTOPE MESHES FOR MASSIVELY-PARALLEL ARCHITECTURES

A Dissertation

by

MICHAEL WAYNE HACKEMACK

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jean Ragusa
Committee Members,	Marvin Adams
	Jim Morel
	Nancy Amato
	Troy Becker
Head of Department,	Yassin Hassan

August 2016

Major Subject: Nuclear Engineering

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

1.1 Introduction

In this chapter, we analyze the Modified Interior Penalty (MIP) form of the diffusion equation as a discretization scheme for use with Diffusion Synthetic Acceleration (DSA) of the DFEM S_N transport equation on unstructured grids. Specifically, we wish to analyze its efficacy on massively-parallel computer architectures where scalability of solution times and memory footprints can become burdensome. This chapter is laid out in the following manner. The remainder of this Section will provide an overview of synthetic acceleration techniques as well as a review of DSA schemes. Section 1.2 provides the DSA methodologies that we will employ for 1-group and thermal neutron upscattering acceleration. We then present the discontinuous Symmetric Interior Penalty (SIP) form of the diffuion equation in Section 1.3 as well as the MIP variant that we will use for our DSA analysis in Section 1.4. The numerical procedures that we will use to solve the diffusion system of equations is given in Section 1.5. The theoretical Fourier analysis tool is given in Section 1.6. Results are provided in Section 1.7 and we finish the chapter with some concluding remarks in Section 1.8

1.1.1 Review of Diffusion Synthetic Accleration Schemes

cleanup the beginning here later...

The ability to efficiently invert the transport (streaming and collision) operator does not necessarily mean that transport solutions can be easily obtained. In general, radiation transport solutions are obtained iteratively. The simplest and widely-used method is a fixed-point scheme (*i.e.* Richardson iteration) ubiquitously called source

iteration (SI) in the transport community. Unfortunately, the iteration process of SI can converge arbitrarily slowly if the problem is optically thick [1]. This corresponds to long mean free paths for neutronics problems. This also corresponds to time steps and material heat capacities tending to infinity and zero, respectively, for thermal radiative transport (TRT) problems.

For these problem regimes in which solution is prohibitively slow, additional steps should be taken to speed up, or accelerate, solution convergence [1]. The most used methods to assist in solution convergence are often called synthetic acceleration techniques. These techniques were first introduced by Kopp [2] and Lebedev [3, 4, 5, 6, 7, 8, 9] in the 1960's. From Kopp's and Lebedev's work, Gelbard and Hageman then introduced two synthetic acceleration options for the low-order operator: diffusion and S_2 [10]. Their diffusion preconditioning led to efficient convergence properties on fine spatial meshes. Reed then showed that Gelbard and Hageman's diffusion preconditioning would yield a diverging system for coarse meshes [11]. At this point in time, no one knew if an unconditionally efficient acceleration method could be derived.

Then in 1976, Alcouffe proposed a remedy to Gelbard and Reed that he called diffusion synthetic acceleration (DSA) [12, 13, 14]. He showed that if you derived the diffusion operator consistently with the discretized transport operator, then SI could be accelerated with DSA in an efficient and robust manner. Larsen and McCoy then demonstrated that unconditional stability required that consistency be maintained in both spatial and angular discretization in their four-step procedure [15, 16]. However, Adams and Martin then showed that partially-consistent diffusion discretizations could effectively accelerate DFEM discretizations of the neutron transport equation [17]. Their modified-four-step procedure (M4S), based on Larsen and McCoy's work, was shown to be unconditionally stable for regular geometries, but divergent for un-

structured multi-dimensional meshes [18]. In more recent years, alternate discretizations for the diffusion operator have been applied to unstructured multi-dimensional grids. These include the partially consistent Wareing-Larsen-Adams (WLA) DSA [19], the fully consistent DSA (FCDSA) [18], and the partially consistent MIP DSA [20, 21, 22].

Most recently, the partially consistent MIP DSA method has been shown to be an unconditionally stable acceleration method for the 2D DFEM transport equation on unstructured meshes. Wang showed that it acted as an effective preconditioner for higher-order DFEM discretizations on triangles [20, 21]. Turcksin and Ragusa then extended the work to arbitrary polygonal meshes [22]. The MIP diffusion operator is symmetric positive definite (SPD) and was shown to be efficiently invertible with preconditioned conjugate gradient (PCG) and advanced preconditioners such as algebraic multi-grid (AMG) [22].

1.1.2 Synthetic Acceleration Overview

Synthetic acceleration techniques have been widely used in the nuclear engineering community to improve solution convergence for prohibitively slow problems. We now provide a general framework for how synthetic acceleration methods are derived. We begin by expressing our neutron transport equation in the following form,

$$(\mathbf{A} - \mathbf{B})\,\Psi = \mathbf{Q}, \tag{1.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are both linear operators, Ψ is the full angular flux solution in space, angle, and energy, and $\mathbf{Q}$ is the source or driving function. If we had the ability to efficiently invert $(\mathbf{A} - \mathbf{B})$ directly, then Ψ could be directly computed:

$$\Psi = (\mathbf{A} - \mathbf{B})^{-1}\,\mathbf{Q}. \tag{1.2}$$

However, since in practice the discretized version of $(\mathbf{A} - \mathbf{B})$ is much more costly to directly invert than the discretized version of $\mathbf{A}$, we instead choose to iteratively solve for Ψ.

To compute Ψ, the following iterative system of equations is usually employed,

$$\mathbf{A}\Psi^{(k+1)} = \mathbf{B}\Psi^{(k)} + \mathbf{Q}, \tag{1.3}$$

where directly solving for $\Psi^{(k+1)}$ yields the following:

$$\Psi^{(k+1)} = \mathbf{A}^{-1}\mathbf{B}\Psi^{(k)} + \mathbf{A}^{-1}\mathbf{Q}. \tag{1.4}$$

For brevity, we define a new operator $\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}$ which is known as the iteration operator. The spectral radius, ρ, of this operator is simply the supremum of the absolute values of its eigenvalues. For this work, we assume that ρ is less than unity to guarantee convergence. We next define the residual, $r^{(k)}$, as the difference between two successive solution iterates,

$$r^{(k)} = \Psi^{(k)} - \Psi^{(k-1)}, \tag{1.5}$$

which can also be written as the following:

$$r^{(k)} = \mathbf{C}r^{(k-1)}. \tag{1.6}$$

With the iteration operator, $\mathbf{C}$, and the residual for iterate k , $r^{(k)}$, defined, we can then write the true, converged solution in terms of the solution at iteration k and an infinite series of residuals:

$$\Psi = \Psi^{(k)} + \sum_{n=1}^{\infty} r^{(k+n)}. \tag{1.7}$$

Using both Eqs. (1.6) and (1.7), we can rewrite Eq. (1.7) using the iteration operator and the last residual,

$$\Psi = \Psi^{(k)} + \left(\mathbf{I} + \mathbf{C} + \mathbf{C}^2 + ...\right) \mathbf{C} r^{(k)}. \tag{1.8}$$

Since we have assumed that the spectral radius of $\mathbf{C}$ is less than unity, the infinite operator series of Eq. (1.8) converges to $(\mathbf{I} - \mathbf{C})^{-1} \mathbf{C}$. This means that we can succinctly write Eq. (1.8) as the following:

$$\Psi = \Psi^{(k)} + (\mathbf{I} - \mathbf{C})^{-1} \mathbf{C} r^{(k)}. \tag{1.9}$$

By using the definition of $\mathbf{C}$ along with some linear algebra, Eq. (1.9) becomes

$$\Psi = \Psi^{(k)} + (\mathbf{A} - \mathbf{B})^{-1} \mathbf{B} r^{(k)}. \tag{1.10}$$

We would like to use the results of Eq. (1.10) to immediately compute our exact transport solution, Ψ. However, this would require the inversion of $(\mathbf{A} - \mathbf{B})$ which we did not employ originally in Eq. (1.1) because of the difficulty. This means that, in its current form, Eq. (1.10) is no more useful to us than Eq. (1.1). This would then be an exercise in futility if we were restricted to only working with the $(\mathbf{A} - \mathbf{B})$ operator. Instead, suppose that we could define an operator, $\mathbf{W}$, that closely approximates $(\mathbf{A} - \mathbf{B})$ but it is easily invertible. If $\mathbf{W}$ efficiently approximates the slowest converging error modes of $(\mathbf{A} - \mathbf{B})$, then Eq. (1.10) can be modified to form a new iterative procedure.

The new iterative procedure begins by simply taking the half-iterate of Eq. (1.4)

instead of its full version: $(k+1/2)$ instead of (k+1). This half-iterate has the form

$$\Psi^{(k+1/2)} = \mathbf{C}\Psi^{(k)} + \mathbf{A}^{-1}\mathbf{Q}. \tag{1.11}$$

We can then express the full-iterate by the suggestion of Eq. (1.10). Using the low-order operator, we express the full-iterate as the following,

$$\Psi^{(k+1)} = \Psi^{(k+1/2)} + \mathbf{W}^{-1}\mathbf{B}r^{(k+1/2)}, \tag{1.12}$$

where $r^{(k+1/2)} = \Psi^{(k+1/2)} - \Psi^{(k)}$. We can also express Eq. (1.12) in terms of just the previous iterate, $\Psi^{(k)}$, and a new operator:

$$\Psi^{(k+1)} = \left[\mathbf{I} - \mathbf{W}^{-1}\left(\mathbf{A} - \mathbf{B}\right)\right]\mathbf{C}\Psi^{(k)} + \left(\mathbf{I} + \mathbf{W}^{-1}\mathbf{B}\right)\mathbf{A}^{-1}\mathbf{Q}. \tag{1.13}$$

Observe in Eq. (1.13) that as $\mathbf{W}$ more closely approximates $(\mathbf{A} - \mathbf{B})$, the operator $\mathbf{W}^{-1}\left(\mathbf{A} - \mathbf{B}\right)$ converges to the identity matrix, $\mathbf{I}$. This means that the spectral radius of this new iteration matrix will approach zero as $\mathbf{W}$ gets closer to $(\mathbf{A} - \mathbf{B})$ and therefore more quickly and efficiently converge to the true solution.

1.2 Diffusion Synthetic Acceleration Methodologies

The procedures outlined in Section 1.1.2 define a general methodology to perform synthetic acceleration on the transport equation. We could utilize any of the acceleration strategies that have been developed over the years including DSA, TSA, BPA, etc. The only difference arises in what form the low-order operator, $\mathbf{W}$, will take. We obviously are focusing on DSA for this dissertation work, and we do so by first describing in Section 1.2.1 a simple 1-group specification of the synthetic acceleration methodology just presented. We then present a pair of strategies that can be employed to accelerate thermal neutron upscattering in Section 1.2.2.

1.2.1 Simple 1-group DSA Strategy

$$\vec{\Omega}_m \cdot \vec{\nabla}\psi_m + \sigma_t \psi_m = c\frac{\sigma_t}{4\pi}\phi + \frac{Q}{4\pi} \tag{1.14}$$

Recall the operator form of the fully discretized transport equation as defined in Section **??**,

$$\begin{aligned} \mathbf{L\Psi} &= \mathbf{M\Sigma\Phi} + \mathbf{Q} \\ \mathbf{\Phi} &= \mathbf{D\Psi} \end{aligned}, \tag{1.15}$$

where $\mathbf{L}$ is the total interaction and streaming operator, $\mathbf{M}$ is the moment-to-discrete operator, $\mathbf{D}$ is the discrete-to-moment operator, $\mathbf{\Sigma}$ is the scattering operator, and $\mathbf{Q}$ is the forcing function. In this case, we simply treat this discretized problem as only having 1 energy group. The functional form of the discretized moment-to-discrete and discrete-to-moment operators are

$$M_m \equiv \sum_{p=0}^{N_P} \frac{2p+1}{4\pi} \sum_{n=-p}^{p} Y_{p,n}(\vec{\Omega}_m), \tag{1.16}$$

and

$$D_{p,n} \equiv \sum_{m=1}^{M} w_m Y_{p,n}(\vec{\Omega}_m)\Psi(\vec{\Omega_m}), \tag{1.17}$$

respectively. This means that the operation $\mathbf{DL}^{-1}$ corresponds to a full-domain transport sweep followed by the computation of the flux moments from the angular flux. We next apply our half-iterate and previous iterate indices on Eq. (1.15) to form the iteration procedure for our transport equation

$$\mathbf{L\Psi}^{(k+1/2)} = \mathbf{M\Sigma\Phi}^{(k)} + \mathbf{Q} \tag{1.18}$$

We can then use the $\mathbf{DL}^{-1}$ operator to present our transport equation of Eq. (1.18) in terms of just the half-iterate flux moments,

$$\mathbf{\Phi}^{(k+1/2)} = \mathbf{DL}^{-1}\mathbf{M\Sigma\Phi}^{(k)} + \mathbf{DL}^{-1}\mathbf{Q}. \tag{1.19}$$

1.2.2 DSA Acceleration Strategies for Thermal Neutron Upscattering

1.2.2.1 Standard Two-Grid (TG) Acceleration

The first thermal neutron upscatter acceleration methodology that we will investigate is the standard two-grid acceleration scheme devised by Adams and Morel [23]. They originally derived the scheme to accelerate 1D multigroup transport problems that are dominated by thermal neutron upscattering. The method can be quickly summarized as the following:

1. Perform a Gauss-Seidel procedure in energy for the thermal groups and converge the inner iterations;

2. Perform a spectral collapse of the transport iteration error into a 1-group diffusion equation;

3. Solve the 1-group diffusion equation for the spatial variation of the transport iteration error;

4. Interpolate the diffusive multigroup error back onto the transport solution.

We now provide fulls details for each of these steps of the TG method.

The first step in the TG method is to perform a Gauss-Seidel procedure in energy across the thermal groups. This means that for every outer iteration (we can also think of these as thermal iterations), we sequentially proceed through the thermal energy groups. Based on the notation of a group set introduced in Section **??**, we can

achieve this Gauss-Seidel iteration scheme by having each thermal group be in its own group set. The TG method then requires full convergence of the within-group inner iterations for each of the group sets. If we have G number of thermal groups, this Gauss-Seidel iteration scheme (with convergence of the inner iterations) leads to the following iteration equation,

$$\mathbf{L_{gg}}\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{g}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k+1/2)} + \mathbf{M}\sum_{g'=g+1}^{G}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k)} + \mathbf{Q}_g, \tag{1.20}$$

where the scattering terms still contain some arbitrary number of moments. In operator form, the exact solution and half-iterate equations are given by

$$\mathbf{L}\Psi = \mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D} + \mathbf{S_U}\right)\Phi + \mathbf{Q}, \tag{1.21}$$

and

$$\mathbf{L}\Psi^{(k+1/2)} = \mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\Phi^{(k+1/2)} + \mathbf{M}\mathbf{S_U}\Phi^{(k)} + \mathbf{Q}, \tag{1.22}$$

respectively. $\mathbf{S_L}$, $\mathbf{S_D}$, and $\mathbf{S_U}$ are the strictly-downscattering, diagonal within-group scattering, and strictly upscattering portions of the scattering matrix, respectively. By moving the downscattering and diagonal portions of the scattering operator to the left side of the equation, inverting the $\mathbf{L}$ operator, and applying the discrete-to-moment operator, $\mathbf{D}$, we can rewrite the iteration equation of Eq. (1.22) in terms of only the flux moments:

$$\left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]\Phi^{(k+1/2)} = \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S_U}\Phi^{(k)} + \mathbf{D}\mathbf{L}^{-1}\mathbf{Q}. \tag{1.23}$$

By inverting the left-side operator, we directly solve for the half-iterate flux moments,

$$\Phi^{(k+1/2)} = \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]^{-1} \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S_U}\Phi^{(k)} + \mathbf{b}, \tag{1.24}$$

where the simplified distributed source term, $\mathbf{b}$, is now defined for further clarity:

$$\mathbf{b} = \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]^{-1} \mathbf{D}\mathbf{L}^{-1}\mathbf{Q}. \tag{1.25}$$

At this point, Eq. (1.24) provides a formulation for the transport solution at iteration $(k+1/2)$ based on the solution at iteration (k). We now require a formulation for the accompanying error at this iteration step. We subtract Eq. (1.20) from the exact transport solution to form an equation specifying the exact error at iteration $(k+1/2)$,

$$\mathbf{L_{gg}}\delta\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{g} \mathbf{\Sigma}_{gg'}\delta\phi_{g'}^{(k+1/2)} + \mathbf{M}\sum_{g'=g+1}^{G} \mathbf{\Sigma}_{gg'}\delta\phi_{g'}^{(k)}, \tag{1.26}$$

where the angular flux and flux moment error terms have an analogous multigroup form,

$$\begin{aligned} \delta\psi_g^{(k+1/2)} &= \psi_g - \psi_g^{(k+1/2)} \\ \delta\phi_g^{(k+1/2)} &= \mathbf{D}\delta\psi_g^{(k+1/2)} \end{aligned}. \tag{1.27}$$

Next, we add and subtract $\mathbf{M}\sum_{g'=g+1}^{G} \mathbf{\Sigma}_{gg'}\phi_{g'}^{(k+1/2)}$ to Eq. (1.26) to form

$$\mathbf{L_{gg}}\delta\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{G} \mathbf{\Sigma}_{gg'}\delta\phi_{g'}^{(k+1/2)} + \mathbf{M}\sum_{g'=g+1}^{G} \mathbf{\Sigma}_{gg'}\left(\phi_{g'}^{(k+1/2)} - \phi_{g'}^{(k)}\right). \tag{1.28}$$

Equation (1.28) can be recast into its appropriate operator notation,

$$\mathbf{L}\delta\Psi^{(k+1/2)} - \mathbf{MS}\delta\Phi^{(k+1/2)} = \mathbf{MS_U}\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right), \tag{1.29}$$

where $\mathbf{S} = \mathbf{S_L} + \mathbf{S_D} + \mathbf{S_U}$ is the full scattering operator. Equation (1.28) and the corresponding operator form in Eq. (1.29) provide the complete formulation of the multrigroup iteration error. Just like it was previously mentioned for the 1-group scenario, these error equations are just as difficult to solve as the full transport equation.

We again choose to utilize the diffusion operator as the low-order operator for the iteration error. Taking the 0th and 1st angular moments of Eq. (1.28) and applying Fick's Law, we arrive at the standard multigroup diffusion equation for the error,

$$\vec{\nabla}\cdot D_g\vec{\nabla}\delta\phi_g^{(k+1/2)} + \sigma_{t,g}\delta\phi_g^{(k+1/2)} - \sum_{g'=1}^{G}\sigma_{s,0}^{gg'}\delta\phi_{g'}^{(k+1/2)} = R_g^{(k+1/2)}, \tag{1.30}$$

where the residual, R_g, is only in terms of the 0th-order scattering moment and has the following form:

$$R_g^{(k+1/2)} = \sum_{g'=g+1}^{G}\sigma_{s,0}^{gg'}\left(\phi_{g'}^{(k+1/2)} - \phi_{g'}^{(k)}\right). \tag{1.31}$$

We could solve these G coupled multigroup diffusion equations of Eq. (1.30) for the iteration error. However, if the number of thermal groups becomes large, then these coupled equations could be become burdensome to solve. Instead, the TG method opts to perform a spectral collapse of the multigroup diffusion error. First, we factorize the multigroup error,

$$\delta\phi_g^{(k+1/2)} = \xi_g \epsilon^{(k+1/2)}(\vec{r}), \qquad \sum_{g=1}^{G} \xi_g = 1, \tag{1.32}$$

into a spatial component, $\epsilon^{(k+1/2)}(\vec{r})$, and an energy component, ξ_g. The spectral shape, ξ_g, is the eigenfunction corresponding to the largest eigenvalue of the infinite medium iteration matrix of Eq. (1.22) with no driving source term. This eigenvalue problem can be succinctly written as,

$$\left(\mathbf{T} - \mathbf{S_{L,0}} - \mathbf{S_{D,0}}\right)^{-1} \mathbf{S_{U,0}} \xi = \rho \xi, \tag{1.33}$$

where $\mathbf{T}$ is the diagonal matrix of total cross sections and $\mathbf{S_{L,0}}$, $\mathbf{S_{D,0}}$, and $\mathbf{S_{U,0}}$ are restricted to the 0th-order moments of the scattering cross sections. Inserting the factorized error of Eq. (1.30) into Eq. (1.30) and summing over energy groups gives

$$\vec{\nabla} \cdot \left\langle D \right\rangle \vec{\nabla} \epsilon^{(k+1/2)} + \vec{\nabla} \cdot \left\langle \vec{D} \right\rangle \epsilon^{(k+1/2)} + \left\langle \sigma \right\rangle \epsilon^{(k+1/2)} = \left\langle R^{(k+1/2)} \right\rangle, \tag{1.34}$$

where the energy-averaged terms are

$$\begin{aligned}
\left\langle D \right\rangle &= \sum_{g=1}^{G} D_g \xi_g, \\
\left\langle \vec{D} \right\rangle &= \sum_{g=1}^{G} D_g \vec{\nabla} \xi_g, \\
\left\langle \sigma \right\rangle &= \sum_{g=1}^{G} \left(\sigma_{t,g} \xi_g - \sum_{g'=1}^{G} \sigma_{s,0}^{gg'} \xi_{g'} \right), \\
\left\langle R^{(k+1/2)} \right\rangle &= \sum_{g=1}^{G} R_g^{(k+1/2)}.
\end{aligned} \tag{1.35}$$

Equation (1.34) is not the standard diffusion equation because of the drift term containing the gradient of the spectral shape. This term is an artifact of the factor-

ization of the error and it is identically zero in homogeneous regions but undefined at material interfaces. The TG method simply neglects this term, which leads to the final form for our coarse-grid error equation,

$$\vec{\nabla} \cdot \big\langle D \big\rangle \vec{\nabla} \epsilon^{(k+1/2)} + \big\langle \sigma \big\rangle \epsilon^{(k+1/2)} = \big\langle R^{(k+1/2)} \big\rangle. \tag{1.36}$$

If we define the left-hand-side matrix operator of Eq. (1.36) as $\mathbf{A}$, then the operator form of this coarse-grid error equation is

$$\mathbf{A}\epsilon^{(k+1/2)} = \mathbf{X}\mathbf{S}_{\mathbf{U}} \left(\Phi^{(k+1/2)} - \Phi^{(k)} \right), \tag{1.37}$$

where $\mathbf{X}$ is the restriction operator that confines the diffusion problem to the 0th-order moment and performs the sum from Eq. (1.35). Solving Eq. (1.36) gives the spatial distribution of the iteration error. We can then update the error for the 0th moment flux with the following:

$$\phi_{g,0}^{(k+1)} = \phi_{g,0}^{(k+1/2)} + \xi_g \epsilon^{(k+1/2)}, \qquad g = 1, ..., G. \tag{1.38}$$

Up to this point, we have provided the full details of the TG methodology including the Gauss-Seidel iteration equations, the process to spectrally-collapse the diffusive error, and the addiditve interpolation of the diffusive error. We now go through these steps using compact operator notation to arive at a single matrix form for the TG accelerated transport iterations. First, we express the full phase-space update equation as

$$\Phi^{(k+1)} = \Phi^{(k+1/2)} + \delta\Phi^{(k+1/2)}. \tag{1.39}$$

The half-iterate solution, $\Phi^{(k+1/2)}$, is given by Eqs. (1.24) and (1.25), and the iteration error, $\delta\Phi^{(k+1/2)}$, is

$$\delta\Phi^{(k+1/2)} = \mathbf{PA}^{-1}\mathbf{XS_U}\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right), \tag{1.40}$$

where $\mathbf{P}$ is the prolongation operator which interpolates the error back into the full phase-space of the transport equation. For the TG method, this prolongation operator acts on only the 0th-order flux moments (the higher moments are characteristically zero) and appropriately adds the error correction for thermal group g with the appropriate spectral weight, ξ_g. Inserting these definitions into Eq. (1.39) gives

$$\begin{aligned}\Phi^{(k+1)} = &\left[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]^{-1}\mathbf{DL}^{-1}\mathbf{MS_U}\Phi^{(k)} \\ &+ \mathbf{PA}^{-1}\mathbf{XS_U}\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right) + \mathbf{b}\end{aligned}. \tag{1.41}$$

We further define the term $\mathbf{F} \equiv \left[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]^{-1}\mathbf{DL}^{-1}$, and use it to re-express Eq. (1.41) into a singular equation for the full-iterate solution,

$$\begin{aligned}\Phi^{(k+1)} &= \mathbf{FMS_U}\Phi^{(k)} + \mathbf{PA}^{-1}\mathbf{XS_U}\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right) + \mathbf{b} \\ &= \mathbf{FMS_U}\Phi^{(k)} + \mathbf{PA}^{-1}\mathbf{XS_U}\left(\ \mathbf{FMS_U}\Phi^{(k)} + \mathbf{b} - \Phi^{(k)}\right) + \mathbf{b} \\ &= \mathbf{FMS_U}\Phi^{(k)} + \mathbf{PA}^{-1}\mathbf{XS_U}\left(\ \mathbf{FMS_U} - \mathbf{I}\right)\Phi^{(k)} + \left(\mathbf{PA}^{-1}\mathbf{XS_U} + \mathbf{I}\right)\mathbf{b} \\ &= \left[\mathbf{FMS_U} + \mathbf{PA}^{-1}\mathbf{XS_U}\left(\ \mathbf{FMS_U} - \mathbf{I}\right)\right]\Phi^{(k)} + \left(\mathbf{PA}^{-1}\mathbf{XS_U} + \mathbf{I}\right)\mathbf{b}\end{aligned}, \tag{1.42}$$

where we note that $\mathbf{b} = \mathbf{FDL}^{-1}\mathbf{Q}$. From Eq. (1.42), the iteration matrix for the TG scheme is given by $\left[\mathbf{FMS_U} + \mathbf{PA}^{-1}\mathbf{XS_U}\left(\ \mathbf{FMS_U} - \mathbf{I}\right)\right]$. The eigenvalues of this iteration matrix will give insight into the convergence properties of the TG method in the asymptotic region.

1.2.2.2 Modified Two-Grid (MTG) Acceleration

The second thermal upscattering acceleration method that we will investigate is a simple modification to the standard TG method of Section 1.2.2.1. At the end of their work involving a TSA variant of the TG method [24], Evans, Clarno, and Morel proposed a modified form for the TG method, which they labeled the Modified Transport Two-Grid (MTTG) method. We wish to adopt their iterative strategy, but use the diffusion equation as our low-order operator. We choose to call this method the Modified Two-Grid (MTG) method. In their work, they proposed to not fully converge the inner iterations for each group in the Gauss-Seidel process. Just like the TG method, we again sequentially proceed through the thermal groups in a Gauss-Seidel manner but only perform 1 transport sweep for each group. This process yields the following iteration scheme,

$$\mathbf{L_{gg}}\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{g-1}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k+1/2)} + \mathbf{M}\sum_{g'=g}^{G}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k)} + \mathbf{Q}_g, \tag{1.43}$$

where it differs with Eq. (1.20) in the ending and beginning energy indices for the $(k+1/2)$ and (k) iterations, respectively. In operator notation, the iteration equation of the MTG method is the following:

$$\mathbf{L}\Psi^{(k+1/2)} = \mathbf{M}\mathbf{S_L}\Phi^{(k+1/2)} + \mathbf{M}\left(\mathbf{S_D} + \mathbf{S_U}\right)\Phi^{(k)} + \mathbf{Q}. \tag{1.44}$$

This operator equation for the MTG differs from Eq. (1.22) of the TG method by the locations of the different scattering operators. Solving for the flux moments, we can give the half-iterate and external source equations as,

$$\Phi^{(k+1/2)} = \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S_L}\right]^{-1}\mathbf{D}\mathbf{L}^{-1}\mathbf{M}\left(\mathbf{S_D} + \mathbf{S_U}\right)\Phi^{(k)} + \mathbf{b}, \tag{1.45}$$

and

$$\mathbf{b} = \left[\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}_{\mathbf{L}}\right]^{-1}\mathbf{D}\mathbf{L}^{-1}\mathbf{Q}. \tag{1.46}$$

respectively.

The generation of the spectrally-collapsed diffusion equation for the MTG iteration error is almost identical to the TG method. The only differences lie with generating the spectral shape functions and the residual. Like the TG method, the spectral shape function for each material is the eigenfunction corresponding to the largest eigenvalue of the infinite medium iteration matrix of Eq. (1.43). This eigenproblem is given by,

$$\left(\mathbf{T} - \mathbf{S}_{\mathbf{L},\mathbf{0}}\right)^{-1}\left(\mathbf{S}_{\mathbf{D},\mathbf{0}} + \mathbf{S}_{\mathbf{U},\mathbf{0}}\right)\xi = \rho\xi, \tag{1.47}$$

where the matrix operators remain from before. With this spectral shape, the average diffusion coefficient and average absorption cross section can again be calculated by Eq. (1.35), while the error residual is given by

$$R_g^{(k+1/2)} = \sum_{g'=g}^{G} \sigma_{s,0}^{gg'}\left(\phi_{g'}^{(k+1/2)} - \phi_{g'}^{(k)}\right). \tag{1.48}$$

Again, the coarse-grid error equation is given by Eq. (1.36), with a corresponding operator form of

$$\mathbf{A}\epsilon^{(k+1/2)} = \mathbf{X}\left(\mathbf{S}_{\mathbf{D}} + \mathbf{S}_{\mathbf{U}}\right)\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right). \tag{1.49}$$

We now provide the full phase-space update equation in a like manner to TG. Again, the update equation can be expressed as

$$\Phi^{(k+1)} = \Phi^{(k+1/2)} + \delta\Phi^{(k+1/2)}, \tag{1.50}$$

where we insert the half-iterate and coarse-grid error terms from Eqs. (1.45) and (1.49), respectively. Defining $\mathbf{F} \equiv \left[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS_L}\right]^{-1}\mathbf{DL}^{-1}$ for brevity, we can give the singular equation for the MTG method,

$$\begin{aligned}
\Phi^{(k+1)} =& \mathbf{FM}\left(\mathbf{S_D} + \mathbf{S_U}\right)\Phi^{(k)} + \mathbf{PA}^{-1}\mathbf{X}\left(\mathbf{S_D} + \mathbf{S_U}\right)\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right) + \mathbf{b} \\
=& \mathbf{FM}\left(\mathbf{S_D} + \mathbf{S_U}\right)\Phi^{(k)} + \mathbf{PA}^{-1}\mathbf{X}\left(\mathbf{S_D} + \mathbf{S_U}\right)\left(\ \mathbf{FM}\left(\mathbf{S_D} + \mathbf{S_U}\right) - \mathbf{I}\right)\Phi^{(k)} \\
&+ \left(\mathbf{PA}^{-1}\mathbf{X}\left(\mathbf{S_D} + \mathbf{S_U}\right) + \mathbf{I}\right)\mathbf{b} \\
=& \left[\mathbf{FM}\left(\mathbf{S_D} + \mathbf{S_U}\right) + \mathbf{PA}^{-1}\mathbf{X}\left(\mathbf{S_D} + \mathbf{S_U}\right)\left(\ \mathbf{FM}\left(\mathbf{S_D} + \mathbf{S_U}\right) - \mathbf{I}\right)\right]\Phi^{(k)} \\
&+ \left(\mathbf{PA}^{-1}\mathbf{X}\left(\mathbf{S_D} + \mathbf{S_U}\right) + \mathbf{I}\right)\mathbf{b}
\end{aligned} \tag{1.51}$$

where we note that $\mathbf{b} = \mathbf{FDL}^{-1}\mathbf{Q}$ and is unchanged from the TG method ($\mathbf{F}$ jsut simply has a different definition). Equation (1.51) gives the iteration matrix for the MTG scheme, which can be used to understand the convergence properties of the method in the asymptotic region.

1.2.2.3 Multigroup Jacobi Acceleration

The third and final thermal neutron upscattering acceleration method that we will investigate is markedly different than the TG and MTG methods. With the increasing expansion of parallel computing resources, the sequential Gauss-Seidel approach for the thermal energy groups necessarily becomes a limiting bottleneck. Therefore, we have natural recourse to develop an alternate thermal upscattering acceleration method that can more effectively make use of parallel algorithms and architectures. Therefore, instead of sequentially marching through the thermal energy groups in a

Gauss-Seidel, we solve them simultaneously at each outer iteration. This is achieved by placing all the thermal energy groups into a single group set. Then, each outer iteration performs one transport sweep where the thermal scattering source was generated from the sweep of the previous outer iteration. This procedure is identical to a block Jacobi iteration where a single inner iteration is performed for each thermal group.

$$\mathbf{L_{gg}}\psi_g^{(k+1/2)} = \mathbf{M}\sum_{g'=1}^{G}\mathbf{\Sigma}_{gg'}\phi_{g'}^{(k)} + \mathbf{Q}_g \tag{1.52}$$

$$\Phi^{(k+1/2)} = \mathbf{DL}^{-1}\mathbf{MS}\Phi^{(k)} + \mathbf{b}, \tag{1.53}$$

and

$$\mathbf{b} = \mathbf{DL}^{-1}\mathbf{Q}. \tag{1.54}$$

respectively.

$$\mathbf{T}^{-1}\left(\mathbf{S_{L,0}} + \mathbf{S_{D,0}} + \mathbf{S_{U,0}}\right)\xi = \rho\xi, \tag{1.55}$$

1.2.2.4 Summary of the Thermal Neutron Upscattering Acceleration Methods

In this work, we have defined three different methodologies to accelerate thermal neutron upscattering: the Two-Grid method, the Modified Two-Grid method, and the Multigroup Jacobi method. These methods are similar in that we perform a single coarsening step for each iteration where a spectrally-collapsed, 1-group diffusion equation is the low-order error operator. However, there are key differences in the iterative procedures of the three methods. Both the TG and MTG methods perform a Gauss-Seidel procedure in energy where the thermal energy groups are solved

sequentially. However, the inner iterations are not converged for any of the thermal groups with the MTG method. The Multigroup Jacobi method is different from the other two in that the thermal groups are iterated upon simultaneously.

For all three methods, we can define a general expression for the full phase-space solution at each iteration by the following,

$$\begin{aligned}\Phi^{(k+1)} = &\left[\mathbf{FM\Sigma} + \mathbf{PA}^{-1}\mathbf{X\Sigma}\left(\mathbf{FM\Sigma} - \mathbf{I}\right)\right]\Phi^{(k)} \\ &+ \left(\mathbf{PA}^{-1}\mathbf{X\Sigma} + \mathbf{I}\right)\mathbf{FDL}^{-1}\mathbf{Q}\end{aligned}, \tag{1.56}$$

where the differences lie in the $\mathbf{F}$ and $\mathbf{\Sigma}$ terms. These terms are given in Table 1.1, and we can clearly see that the differences in the schemes arise from the ordering of the scattering operators.

Table 1.1: Iteration terms for the different thermal upscatter acceleration methods.

Method	$\mathbf{F}$	$\mathbf{\Sigma}$
Two-Grid	$\left[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{M}\left(\mathbf{S_L} + \mathbf{S_D}\right)\right]^{-1}\mathbf{DL}^{-1}$	$\mathbf{S_U}$
Modified Two-Grid	$\left[\mathbf{I} - \mathbf{DL}^{-1}\mathbf{MS_L}\right]^{-1}\mathbf{DL}^{-1}$	$\mathbf{S_D} + \mathbf{S_U}$
Multigroup Jacobi	$\mathbf{DL}^{-1}$	$\mathbf{S_L} + \mathbf{S_D} + \mathbf{S_U}$

1.3 Symmetric Interior Penalty Form of the Diffusion Equation

So far, we have presented several strategies in Section 1.2 in which DSA can be used to accelerate both within-group scattering and thermal neutron upscattering. We have also simply stated that our low-order operator will be the diffusion equation. However, we have not presented the exact form of the diffusion equation that we will utilize. In Section 1.4, we present the full form of the modified interior penalty (MIP) form of the diffusion equation that we will use as our low-order operator for DSA

calculations. However, we first present in this Section a more generalized version of the interior penalty form that we could use as a stand-alone solver for the standard diffusion equation: the symmetric interior penalty (SIP) form [25, 26, 27].

We begin our discussion of the SIP form by analyzing the standard form of the diffusion equation,

$$-\vec{\nabla} \cdot D(\vec{r})\vec{\nabla}\Phi(\vec{r}) + \sigma\Phi(\vec{r}) = Q(\vec{r}), \qquad \vec{r} \in \mathcal{D}, \tag{1.57}$$

with Dirichlet boundary conditions

$$\Phi(\vec{r}) = \Phi_0(\vec{r}), \qquad \vec{r} \in \partial\mathcal{D}^d, \tag{1.58}$$

Neumann boundary conditions

$$-D\partial_n\Phi(\vec{r}) = J_0(\vec{r}), \qquad \vec{r} \in \partial\mathcal{D}^n, \tag{1.59}$$

and Robin boundary conditions

$$\frac{1}{4}\Phi(\vec{r}) + \frac{D}{2}\partial_n\Phi(\vec{r}) = J^{inc}(\vec{r}), \qquad \vec{r} \in \partial\mathcal{D}^r. \tag{1.60}$$

We then convert Eq. (1.57) into its weak formulation by left-multiplying it with the test function, b, and apply Gauss' theorem to the Laplacian term,

$$\left(D\vec{\nabla}b, \vec{\nabla}\Phi\right)_{\mathcal{D}} - \left\langle b, D\partial_n\Phi \right\rangle_{\partial\mathcal{D}} + \left(\sigma b, \Phi\right)_{\mathcal{D}} = \left(b, Q\right)_{\mathcal{D}} \tag{1.61}$$

If we were to use the CFEM form of Eq. (1.61), then there would be no further formulations required except on how to properly apply the boundary term: $\left\langle b, D\partial_n\Phi \right\rangle_{\partial\mathcal{D}}$. For the neumann and robin boundary conditions, this is straightforward since we

simply need to insert the definitions of the outgoing currents, $D\partial_n\Phi$, of Eqs. (1.59) and (1.60) into Eq. (1.61). However, this still leaves the question of how to handle the dirichlet boundary conditions. Again, if we were to use CFEM, we could simply strongly enforce these boundary conditions [28].

Instead, we are choosing to utilize a discontinuous form of the diffusion equation. This means that we can employ the same DG finite elements used in the discretization of the transport operator in Section **??**. With this in mind, we recast Eq. (1.61) to only contain the appropriate inner products for element K,

$$\Big(D\vec{\nabla}b, \vec{\nabla}\Phi\Big)_K - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial K} + \Big(\sigma b, \Phi\Big)_K = \Big(b, Q\Big)_K, \tag{1.62}$$

where we use the same notation for the volumetric and surface inner products as Section **??**. We further decompose the boundary terms for element K into its respective interior ($\partial K \backslash \partial\mathcal{D}$), dirichlet ($\partial K \cap \partial\mathcal{D}^d$), neumann ($\partial K \cap \partial\mathcal{D}^n$), and robin ($\partial K \cap \partial\mathcal{D}^r$) components:

$$\begin{aligned}
&\Big(D\vec{\nabla}b, \vec{\nabla}\Phi\Big)_K + \Big(\sigma b, \Phi\Big)_K - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial K\backslash\partial\mathcal{D}} \\
&- \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial K\cap\partial\mathcal{D}^d} - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial K\cap\partial\mathcal{D}^n} - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial K\cap\partial\mathcal{D}^r} \\
&= \Big(b, Q\Big)_K
\end{aligned} \tag{1.63}$$

We can then immediately utilize the definitions of the outgoing currents from Eqs. (1.59) and (1.60), and add the neumann and robin boundary contributions from element K:

$$\begin{aligned}\Big(D\vec{\nabla}b,\vec{\nabla}\Phi\Big)_K+\Big(\sigma b,\Phi\Big)_K\\ -\Big\langle b,D\partial_n\Phi\Big\rangle_{\partial K\setminus\partial\mathcal{D}}-\Big\langle b,D\partial_n\Phi\Big\rangle_{\partial K\cap\partial\mathcal{D}^d}+\frac{1}{2}\Big\langle b,\Phi\Big\rangle_{\partial K\cap\partial\mathcal{D}^r}\\ =\Big(b,Q\Big)_K-\Big\langle b,J_0\Big\rangle_{\partial K\cap\partial\mathcal{D}^n}+2\Big\langle b,J^{inc}\Big\rangle_{\partial K\cap\partial\mathcal{D}^r}.\end{aligned}\tag{1.64}$$

Unfortunately, we are now left with the burden of deciding what to do with element K's interior and dirichlet boundary surface terms. In conforming CFEM analysis, we would enforce at least C^0 continuity across the elements, thus not allowing any interelement jumps in the solution. Instead, with the choice of a DG formulation, we have a wide variability in how we wish to weakly express the discontinuous solution between two elements. This choice is extended to the dirichlet boundary conditions as well. Instead of simply strongly-enforcing the dirichlet conditions, we choose to weakly enforce them via a penalty method. The idea of penalty methods can be traced back to [29], where the weakly-enfoced dirichlet conditions now have the form,

$$\Phi(\vec{r})+\frac{1}{\kappa}D\partial_n\Phi(\vec{r})=\Phi_0(\vec{r}),\qquad \vec{r}\in\partial\mathcal{D}^d,\tag{1.65}$$

where κ is known as the penalty coefficient and $\kappa\gg 1$. It is clear that as κ becomes large, the solution, Φ, converges to the dirichlet value, Φ_0. In his work [30], Nitsche further proposed a consistent formulation with this penalty method via symmetrization. This led to the following weak formulation of the Laplacian term with dirichlet boundary conditions,

$$\begin{aligned}\Big(D\vec{\nabla}b,\vec{\nabla}\Phi\Big)_{\mathcal{D}}-\Big\langle D\partial_n b,\Phi\Big\rangle_{\partial\mathcal{D}^d}-\Big\langle b,D\partial_n\Phi\Big\rangle_{\partial\mathcal{D}^d}\\ +\Big\langle\kappa b,(\Phi-\Phi_0)\Big\rangle_{\partial\mathcal{D}^d}=-\Big\langle D\partial_n b,\Phi_0\Big\rangle_{\partial\mathcal{D}^d}\end{aligned}\tag{1.66}$$

where we dropped the reaction and forcing terms. Here the penalty coefficient, κ, has

the form $\kappa = \alpha/h$ and $\alpha > 1$ to ensure stability. Later, Arnold proposed extending the weak enforcement of the dirichlet boundaries by Nitsche onto all interior surfaces [31]. If the same symmetric consistency of Nitsche is utilized and we integrate over all mesh elements, then our weak formulation for the solution across all interior faces becomes,

$$\left\langle \kappa[\![b]\!], [\![\Phi]\!] \right\rangle_{E_h^i} + \left\langle [\![b]\!], \{\{D\partial_n\Phi\}\} \right\rangle_{E_h^i} + \left\langle \{\{D\partial_n b\}\}, [\![\Phi]\!] \right\rangle_{E_h^i} = 0, \tag{1.67}$$

where κ is again a penalizing coefficient to ensure stability. The mean value and the jump of the terms on a face from Eq. (1.67) are defined as

$$\{\{\Phi\}\} \equiv \frac{\Phi^+ + \Phi^-}{2} \qquad \text{and} \qquad [\![\Phi]\!] \equiv \Phi^+ - \Phi^-, \tag{1.68}$$

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

$$\Phi^\pm \equiv \lim_{s \to 0^\pm} \Phi(\vec{r} + s\vec{n}), \tag{1.69}$$

where the face's unit normal direction, $\vec{n}$, has been arbitrarily chosen.

These weak formulations for the dirichlet boundary conditions and the interior faces can now inserted into Eq. (1.64). From there, we sum the remaining inner products besides the interior face terms across all elements. With this completed, the SIP form of the diffusion equation can be succinctly written as

$$a^{SIP}(b, \Phi) = b^{SIP}(b), \tag{1.70}$$

with the following bilinear matrix:

$$
\begin{aligned}
a^{SIP}(b,\Phi) = \Big(D\vec{\nabla}b, \vec{\nabla}\Phi\Big)_{\mathcal{D}} + \Big(\sigma b, \Phi\Big)_{\mathcal{D}} + \frac{1}{2}\Big\langle b, \Phi\Big\rangle_{\partial\mathcal{D}^r} \\
+\Big\langle \kappa_e^{SIP}[\![b]\!], [\![\Phi]\!]\Big\rangle_{E_h^i} + \Big\langle [\![b]\!], \{\{D\partial_n\Phi\}\}\Big\rangle_{E_h^i} + \Big\langle \{\{D\partial_n b\}\}, [\![\Phi]\!]\Big\rangle_{E_h^i}, \\
+\Big\langle \kappa_e^{SIP} b, \Phi\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle b, D\partial_n\Phi\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle D\partial_n b, \Phi\Big\rangle_{\partial\mathcal{D}^d}
\end{aligned}
\tag{1.71}
$$

and with the following linear right-hand-side:

$$
\begin{aligned}
b^{SIP}(b) = \Big(b, Q\Big)_{\mathcal{D}} - \Big\langle b, J_0\Big\rangle_{\partial\mathcal{D}^n} + 2\Big\langle b, J^{inc}\Big\rangle_{\partial\mathcal{D}^r}. \\
+\Big\langle \kappa_e^{SIP} b, \Phi_0\Big\rangle_{\partial\mathcal{D}^d} - \Big\langle D\partial_n b, \Phi_0\Big\rangle_{\partial\mathcal{D}^d}
\end{aligned}
\tag{1.72}
$$

As previously stated, the general penalty term, κ needs to have sufficient positive measure to ensure stability. From previous investigations [20, 21, 22], we choose the SIP penalty coefficient to be face dependent with the following form,

$$
\kappa_e^{SIP} = \begin{cases} \frac{c}{2}\left(\frac{D^+}{h^+} + \frac{D^-}{h^-}\right) & e \in E_h^i \\ c\frac{D^-}{h^-} & e \in \partial\mathcal{D} \end{cases}, \tag{1.73}
$$

for interior, E_h^i, and boundary, $\partial\mathcal{D}$, faces respectively. In Eq. (1.73), $h^{\pm}$ is the orthogonal projection of the face, e, into the cells defined by the trace in Eq. (1.69). Turcksin and Ragusa, [22], defined $h^{\pm}$ for 2D polygons, whose definitions can be seen in Table 1.2. The orthogonal projection for both triangles and quadrilaterals can be explicitly defined from simple geometric relationships. However, for polygons with > 4 faces, there is no explicit geometric relationship to define the orthogonal projection. Instead, the polygon is approximated as regular, and the orthogonal projection is no longer face-dependent. For polygons with an even number of faces greater than 4, the orthogonal projection is twice the apothem, which is the line

segment between the polygon's center and the midpoint of each polygon's side. For odd number of faces greater than 4, the polygon's orthogonal projection becomes the sum of the apothem and the circumradius.

In a similar manner to the 2D orthogonal projections defined in Table 1.2, we define our choice for the extension of the orthogonal projections to 3D in Table 1.3. Like triangles and quadrilaterals in 2D, the orthogonal projections for tetrahedra and hexahedra can be explicitly defined from the volume equations for pyramids and parallelepipeds, respectively. For cells that are not tetrahedra or hexahedra, we introduce an approximation similar to 2D where we treat the cell as a regular polyhedron. In 3D there is no compact formula that can be given, unlike the definitions of the apothem and circumradius for 2D. Instead, we take the geometric limit of a polyhedra as the number of faces tends to infinity (a sphere). In this limiting case, the orthogonal projection simply becomes the sphere's diameter. We can then define the sphere's diameter with geometric information that would also be available to polyhedra by dividing a sphere's volume (the polyhedral volume) by its surface area (the sum of the areas of the polyhedral faces). While this leads to a gross approximation of the orthogonal projection for polyhedra that are not tetrahedra or hexahedra, it will provide appropriate geometric measure, especially for strictly convex polyhedra.

Table 1.2: Orthogonal projection, h, for different polygonal types: A_K is the area of cell K, L_e is the length of face e, and P_K is the perimeter of cell K.

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

Table 1.3: Orthogonal projection, h, for different polyhedral types: V_K is the volume of cell K, A_e is the area of face e, and SA_K is the surface area of cell K.

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

1.3.1 Elementary Stiffness Matrices

In Eqs. (1.71) and (1.72), there are two additional elementary matrix types required other than those presented in Chapter **??**. The first has the form of $\left(D\vec{\nabla}b, \vec{\nabla}\Phi\right)_K$ and is referred to as the stiffness matrix [28]. For a cell K, we define the stiffness matrix $\mathbf{S}$ as

$$\mathbf{S}_K = \int_K \vec{\nabla}\mathbf{b}_K \cdot \vec{\nabla}\mathbf{b}_K^T \, dr, \tag{1.74}$$

which has dimensionality (N_KxN_K). Just like the cell-wise elementary matrices presented in Chapter **??**, it is possible that these integrals can be computed analytically, depending on the FEM basis functions used. However, for most of the 2D basis functions presented in Chapter **??**, this cannot be done. Instead, we can again employ a numerical quadrature set, $\left\{\vec{x}_q, w_q^K\right\}_{q=1}^{N_q}$, for cell K, consisting of N_q points, $\vec{x}_q$, and weights, w_q^K. Using this quadrature set, the stiffness matrix can be calculated by the following

$$\mathbf{S}_K = \sum_{q=1}^{N_q} w_q^K \vec{\nabla}\mathbf{b}_K(\vec{x}_q) \cdot \vec{\nabla}\mathbf{b}_K^T(\vec{x}_q). \tag{1.75}$$

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

$$\mathbf{S}_K = \begin{bmatrix} \int_K \vec{\nabla} b_1 \cdot \vec{\nabla} b_1 & \dots & \int_K \vec{\nabla} b_1 \cdot \vec{\nabla} b_j & \dots & \int_K \vec{\nabla} b_1 \cdot \vec{\nabla} b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_i \cdot \vec{\nabla} b_1 & \dots & \int_K \vec{\nabla} b_i \cdot \vec{\nabla} b_j & \dots & \int_K \vec{\nabla} b_i \cdot \vec{\nabla} b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_K \vec{\nabla} b_{N_K} \cdot \vec{\nabla} b_1 & \dots & \int_K \vec{\nabla} b_{N_K} \cdot \vec{\nabla} b_j & \dots & \int_K \vec{\nabla} b_{N_K} \cdot \vec{\nabla} b_{N_K} \end{bmatrix}, \tag{1.76}$$

where an individual matrix entry is of the form:

$$S_{i,j,K} = \int_K \vec{\nabla} b_i \cdot \vec{\nabla} b_j. \tag{1.77}$$

1.3.2 Elementary Surface Gradient Matrices

The second new elementary matrix corresponds to the integrals of the product of the basis functions with their gradients on a given surface. For a given face f, these matrices are of the general form: $\left\langle D\partial_n b, \Phi \right\rangle_f$. These terms are analagous to the cell streaming matrix but are computed on the cell boundary with dimensionality $(d-1)$. Analyzing a single face, f, in cell K, the analytical surface gradient matrix is of the form,

$$\mathbf{N}_{f,K} = \int_f \vec{n}(s) \cdot \vec{\nabla} \mathbf{b}_K \, \mathbf{b}_K^T \, ds, \tag{1.78}$$

where the surface normal, $\vec{n}$, is directed outwards from cell K and is allowed to vary along the cell face. From the analytical integral, we can see that these matrices have dimensionality, (N_KxN_K). We include the operation of the dot product between the outward normal and the basis function gradient for two reasons. First, it reduces the dimensionality of the matrices. Second, because the interior face terms in the SIP

bilinear form are independent of the orientation of the normal unit vector along the face, we do not need to perform any additional handling of the face normals. We can see that the bilinear form is independent of the face normal orientation by observing the following relations:

$$\begin{aligned} \Big\langle [\![u]\!], \{\{\partial_n v\}\} \Big\rangle_f &= -\Big\langle \{\{\vec{n}u\}\}, \{\{\vec{n}\partial_n v\}\} \Big\rangle_f \\ \Big\langle \{\{\partial_n u\}\}, [\![v]\!] \Big\rangle_f &= -\Big\langle \{\{\vec{n}\partial_n u\}\}, \{\{\vec{n}v\}\} \Big\rangle_f \end{aligned}. \tag{1.79}$$

If the direction of the normal is changed from $\vec{n}$ to $-\vec{n}$ in Eq. (1.79), we can see that these terms are not modified.

Similar to the surface matrix defined in Chapter **??**, it is possible that the gradients of the basis functions cannot be integrated analytically along a cell face. Using the same face-wise quadrature notation as before, $\left\{\vec{x}_q, w_q^f\right\}_{q=1}^{N_q^f}$, we can numerically calculate the surface gradient matrix for face f along cell K:

$$\mathbf{N}_{f,K} = \sum_{q=1}^{N_q^f} w_q^f \vec{n}(\vec{x}_q)\, \vec{\nabla}\mathbf{b}_K(\vec{x}_q)\, \mathbf{b}_K^T(\vec{x}_q). \tag{1.80}$$

In this case, the local face-wise surface gradient matrix for face f has the full matrix form,

$$\mathbf{N}_{f,K} = \begin{bmatrix} \int_f \vec{n}\cdot\vec{\nabla}b_1\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_1\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_1\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\cdot\vec{\nabla}b_i\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_i\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_i\, b_{N_K} \\ \vdots & & \vdots & & \vdots \\ \int_f \vec{n}\cdot\vec{\nabla}b_{N_K}\, b_1 & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_{N_K}\, b_j & \dots & \int_f \vec{n}\cdot\vec{\nabla}b_{N_K}\, b_{N_K} \end{bmatrix}, \tag{1.81}$$

where an individual matrix entry is of the form:

$$N_{i,j,f,K} = \int_f \vec{n} \cdot \vec{\nabla} b_i \, b_j. \tag{1.82}$$

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

In Section 1.3, we presented the SIP form of the diffusion equation that uses discontinuous Galerkin finite elements. This form can be used as a general solver for the diffusion equation that contains boundary conditions of the first three kinds: dirichlet, neumann, and robin. From the SIP form, we simply need to make some modifications to account for the boundary conditions that arise from the transport solution error as detailed in Section 1.2. The two types of transport conditions that we have considered in this work are dirichlet type conditions (incoming incident and vacuum) and neumann type conditions (reflecting). Since there is no iteration error associated with the incident boundary conditions, we can express the corresponding diffusion boundary condition as a zero dirichlet condition ($\delta\Phi_0 = 0$). Conversely, reflecting transport boundary conditions yield neumann diffusion boundary conditions. However, depending on the mesh and sweep ordering employed, we are not guaranteed to have this reflecting boundary condition error, δJ^{inc}, be strictly zero. If we seek to accelerate the k iterate, then the error in the incoming current, δJ^{inc}, is given by

$$\delta J^{inc} = \sum_{\vec{\Omega}_m \cdot \vec{n} > 0} w_m \left(\vec{\Omega}_m \cdot \vec{n} \right) \delta \Psi_m^{(k)} \tag{1.83}$$

Using these modifications to the SIP diffusion form with the appropriate boundary conditions required to express the transport solution error, we write the MIP diffusion

form as

$$a^{MIP}(b,\delta\Phi) = b^{MIP}(b), \tag{1.84}$$

with the following bilinear matrix,

$$\begin{aligned} a^{MIP}(b,\delta\Phi) &= \left(D\vec{\nabla}b,\vec{\nabla}\delta\Phi\right)_{\mathcal{D}} + \left(\sigma b,\delta\Phi\right)_{\mathcal{D}} \\ +\left\langle\kappa_e^{MIP}[\![b]\!],[\![\delta\Phi]\!]\right\rangle_{E_h^i} &+ \left\langle[\![b]\!],\{\{D\partial_n\delta\Phi\}\}\right\rangle_{E_h^i} + \left\langle\{\{D\partial_n b\}\},[\![\delta\Phi]\!]\right\rangle_{E_h^i}, \\ +\left\langle\kappa_e^{MIP}b,\delta\Phi\right\rangle_{\partial\mathcal{D}^d} &- \frac{1}{2}\left\langle b,D\partial_n\delta\Phi\right\rangle_{\partial\mathcal{D}^d} - \frac{1}{2}\left\langle D\partial_n b,\delta\Phi\right\rangle_{\partial\mathcal{D}^d} \end{aligned} \tag{1.85}$$

and with the following linear right-hand-side,

$$b^{MIP}(b) = \left(b,Q\right)_{\mathcal{D}} + \left\langle b,\delta J^{inc}\right\rangle_{\partial\mathcal{D}^{ref}}. \tag{1.86}$$

The MIP penalty coefficient also needs to be of a different form than the one used for SIP. From Eq. (1.73), we can see that as the orthogonal projection, h, grows large compared to the diffusion coefficient, D, the SIP penalty coefficient can become arbitrarily small. Wang and Ragusa demonstrated in [32] that if the penalty coefficient becomes too small, then MIP used as a DSA diffusion form becomes unstable. Instead, they limited κ^{MIP} to the maximum of either κ^{SIP} or $1/4$. The value of $1/4$ arises as the constant in the terms $\left\langle[\![b]\!],[\![\delta\Phi]\!]\right\rangle_{E_h^i}$ and $\left\langle b,\delta\Phi\right\rangle_{\partial\mathcal{D}^d}$ when the *diffusion conforming form* (DCF) of the diffusion equation is consistently derived from the DGFEM transport equation [20]. This new definition of κ_e^{MIP} can be succinctly written as

$$\kappa_e^{MIP} = \max\left(\kappa_e^{SIP},\frac{1}{4}\right). \tag{1.87}$$

Just like the SIP penalty coefficient, this definition of κ_e^{MIP} ensures that the MIP bilinear form of Eq. (1.85) is SPD. We next describe the procedures used to efficiently invert the MIP system matrix for our work.

1.5 Solving the MIP Diffusion Problem

Equations (1.85) and (1.86) form the system matrix and right-hand-side, respectively, that we need to solve for a given DSA step. Just like the system of equations for the transport problem is too large to solve in a direct manner, we again employ an iterative scheme. Because the MIP bilinear form is SPD, we have natural recourse to use the simple Preconditioned Conjugate Gradient (PCG) method [33]. If the system of equations you are trying to solve for is SPD, then Conjugate Gradient (CG) is the most light-weight iterative method possible. It has a low memory footprint and is guaranteed to converge in $(N_{dof}/2)$ iterations, even if it is ill-conditioned. With a good preconditioner, then the iteration counts with PCG can be reduced substantially further. If we define $\mathbf{A}$ as the MIP system matrix to be inverted and $\mathbf{b}$ as the right-hand-side vector, then the CG algorithm acts by minimizing the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$. This is accomplished by taking successive operations of matrix-vector multiplications on conjugate krylov vectors, $\mathbf{p}_k$. The PCG algorithm performs one additional step by solving the equation $\mathbf{M}\mathbf{z}_\mathbf{k} = \mathbf{r}_\mathbf{k}$ at each CG iteration, where $\mathbf{M}$ is some preconditioner. We provide the simple pseudocode for PCG in algorithm 1.

There are several choices of preconditioners that can be employed with PCG [33]. Depending on the structure and conditioning of the matrix to be inverted, some simple preconditioners can be effective. We can decompose the MIP system matrix, $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$, into its strictly lower-triangular portion, $\mathbf{L}$, its strictly diagonal portion, $\mathbf{D}$, and its strictly lower-triangular portion, $\mathbf{L}^T$. The simplest preconditioner we could employ is Jacobi preconditioning, which is just the strictly diagonal portion

of the system matrix: $\mathbf{M} = \mathbf{D}$. This preconditioner is effective for diagonally-dominant matrices. The next preconditioner would be a simple Symmetric Successive Over-Relaxation (SSOR) method: $\mathbf{M}(\omega) = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}\mathbf{D}+\mathbf{L}\right)\mathbf{D}^{-1}\left(\frac{1}{\omega}\mathbf{D}+\mathbf{L}\right)^T$. The PCG algorithm can be simplified with this preconditioner choice by the Eisenstat trick [34]. The final simple preconditioner that we will consider is Incomplete LU Factorization (ILU). Instead of simply decomposing the system matrix, we factorize it into a unit-lower triangular portion, $\tilde{\mathbf{L}}$, and an upper triangular portion, $\tilde{\mathbf{U}}$. These factorized matrices then form our preconditioner: $\mathbf{M} = \tilde{\mathbf{L}}\tilde{\mathbf{U}}$. From this factorization, the preconditioning step to solve $\mathbf{Mz} = \mathbf{r}$ is accomplished by first solving $\tilde{\mathbf{L}}\mathbf{y} = \mathbf{r}$, followed by $\tilde{\mathbf{U}}\mathbf{z} = \mathbf{y}$. While this leads to a second preconditioning step, the factorized matrices are easy to invert since they are triangular.

Besides Jacobi, SSOR and ILU preconditioning, we also seek to investigate multi-grid methods to invert the MIP system matrix.

Algorithm 1 PCG Algorithm

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$
for k=1,2,... **do**
  Solve: $\mathbf{Mz}_{i-1} = \mathbf{r}_{i-1}$
  $\rho_{i-1} = \mathbf{r}_{i-1}^T\mathbf{z}_{i-1}$
  if $k = 1$ **then**
    $\mathbf{p}_i = \mathbf{z}_{i-1}$
  else
    $\mathbf{p}_i = \mathbf{z}_{i-1} + \frac{\rho_{i-1}}{\rho_{i-2}}\mathbf{p}_{i-1}$
  end if
  $\mathbf{q}_i = \mathbf{Ap}_i$
  $\alpha_i = \frac{\rho_{i-1}}{\mathbf{p}_i^T\mathbf{q}_i}$
  $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\mathbf{p}_i$
  $\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i\mathbf{q}_i$
  if $\frac{||\mathbf{r}_i||}{||\mathbf{b}||} <$ tol **then**
    Exit
  end if
end for

1.6 Fourier Analysis

With the acceleration methodologies and diffusion discretization scheme described in detail, we now present the Fourier Analysis tool. Fourier Analysis is commonly used to analyze the performance characteristics of acceleration schemes for the transport equation. It is a powerful tool because it allows us to express the iteration error in terms of a fourier series. Then, because of the orthogonality of the terms in the series, each fourier mode can be analyzed independently. If these modes were not independent, then we would have no ability to simultaneously solve the infinite spectrum of fourier modes.

$$\Psi^{(k)}(\vec{r},\vec{\Omega}) = \hat{\Psi}^{(k)}(\vec{\Omega})\, e^{i\vec{\lambda}\cdot\vec{r}} \tag{1.88}$$

or in a discretized case for angular direction m,

$$\Psi_m^{(k)}(\vec{r}) = \hat{\Psi}_m^{(k)}\, e^{i\vec{\lambda}\cdot\vec{r}}, \tag{1.89}$$

where $i = \sqrt{-1}$.

$$\Phi^{(k)}(\vec{r}) = \hat{\Phi}^{(k)} e^{i\vec{\lambda}\cdot\vec{r}} \tag{1.90}$$

iteration matrices here...

spectral radius = largest eigenvalue here...

For this work, all fourier analysis was performed in MATLAB. All the eigenmodes corresponding to a fourier wave number for a given iteration matrix can be easily computed with MATLAB's built-in *eig* function. The maximum eigenvalue is found over the wave number space by use of the Nelder-Mead simplex algorithm [35]. We stress that some sort of minimization algorithm must be employed because some

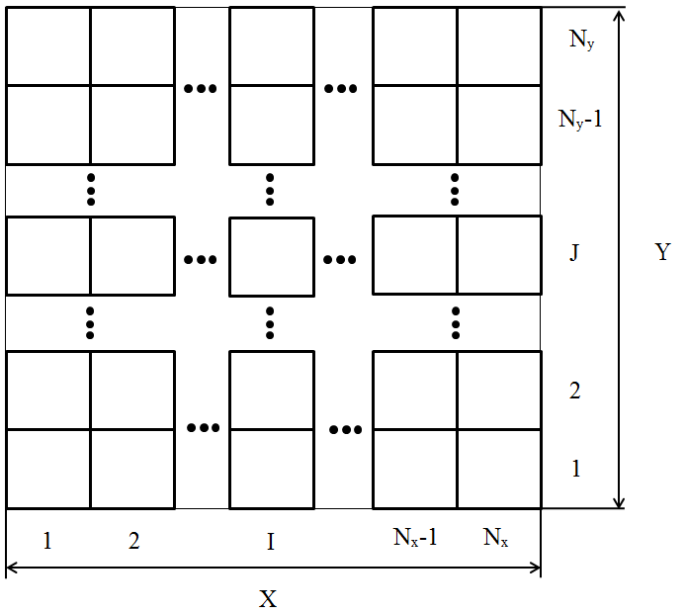
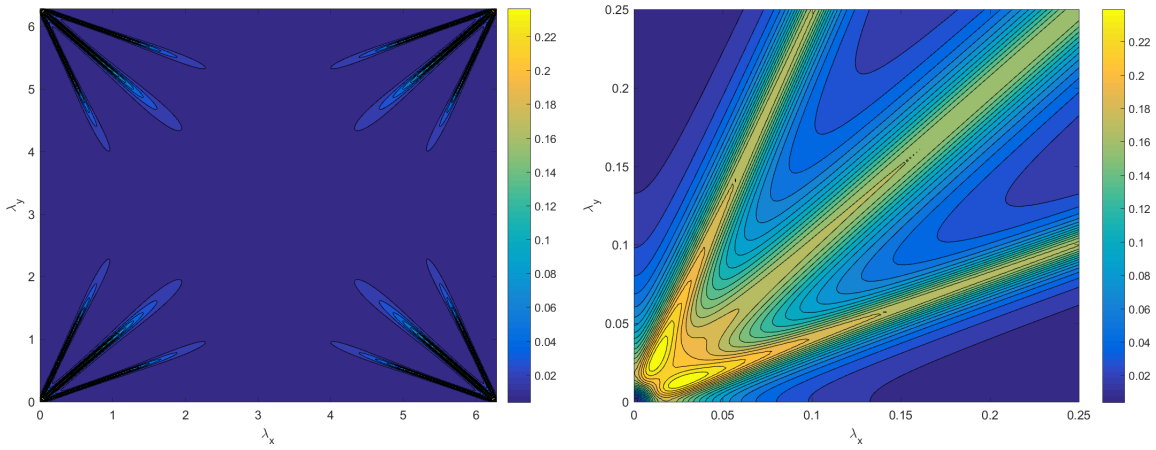


Figure 1.1: Fourier domain for 2D quadrilateral cells or an axial slice of 3D hexahedral cells in a regular grid.

problem configurations can have extremely narrow local maxima. These difficult to find local maxima can correspond to the global maximum which is our desired spectral radius that we wish to compute.

We illustrate the necessity for a minimization algorithm in Figure 1.2. We have modeled a single 2D square mesh cell with dimensions $X = 1$ and $Y = 1$. The total cross section, σ_t, is set to 10^{-2} and the scattering ratio, c, set to 0.9999. We use the $S4$ level-symmetric quadrature set. The left image of Figure 1.2 has the 2D fourier wave number span the full domain space of $[\lambda_x, \lambda_y] = [0, 2\pi]^2$. The right image zooms in on the wave number ranging: $[\lambda_x, \lambda_y] = [0, 1/4]^2$. From the right image, we can see two extremely narrow local maxima. We can qualitatively ascertain that these local maxima would be difficult to find if we had simply laid a grid of wave number points over $[0, 2\pi]^2$. Chang and Adams presented an even more extreme example of a difficult to find global maximum using transport synthetic acceleration [36].



(a) (b)

Figure 1.2: 2D fourier wave form for MIP, with 1 square cell with 1e-2 mfp, with the PWL coordinates, with the LS4 quadrature, and where the wave numbers range from: (a) $[\lambda_x, \lambda_y] = [0, 2\pi]^2$ and (b) $[\lambda_x, \lambda_y] = [0, 1/4]^2$.

1.7 Numerical Results

We now present all results

1.7.1 Transport Solutions in the Thick Diffusive Limit

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

$$\vec{\Omega} \cdot \vec{\nabla}\Psi + \sigma_t \Psi = \frac{\sigma_s}{4\pi}\sigma_t + \frac{Q_0}{4\pi} \tag{1.91}$$

As the transport problem becomes more optically thick, the total mean free paths of the neutroncs increases. In the thick diffusion limit, the domain mean free path approaches infinity. If we fix the physical dimensions of the problem to some finite value, then we can scale the cross sections and the source term to reflect the properties of the thick diffusion limit. In the limit the total and scattering cross sections tend to infinity whilc the absorption cross section and the source term tend to zero. If we

introduce a scaling parameter, ϵ,

$$\begin{aligned} \sigma_t &\rightarrow \frac{\sigma_t}{\epsilon} \\ \sigma_a &\rightarrow \epsilon\sigma_t \\ \sigma_s &\rightarrow \left(\frac{1}{\epsilon} - \epsilon\right)\sigma_t \\ \frac{Q_0}{4\pi} &\rightarrow \epsilon\frac{Q_0}{4\pi} \end{aligned} \tag{1.92}$$

Inserting these scaled cross sections and source term into Eq. (1.91) leads to following scaled transport equation:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi + \frac{\sigma_t}{\epsilon}\Psi = \sigma_t\left(\frac{1}{\epsilon} - \epsilon\right)\frac{\Phi}{4\pi} + \epsilon\frac{Q_0}{4\pi}. \tag{1.93}$$

We can also use the scaled terms of Eq. (1.92) to give the corresponding scaled diffusion equation. If we take the 0th and 1st moments of Eq. (1.93) in the usual way and assume that the P1 terms obey Fick's Law, then the scaled diffusion equation is

$$\epsilon\vec{\nabla}\cdot\frac{1}{3\sigma_t}\vec{\nabla}\Phi + \epsilon\sigma_t\Phi = \epsilon Q_0. \tag{1.94}$$

One can immediately see that Eq. (1.94) does not truly scale because there is an ϵ for each term. This is the desired behavior we want to see from the diffusion equation because, as $\epsilon \rightarrow 0$, the transport equation will converge to its diffusive limit and satisfy a diffusion equation.

For the sake of analysis, we seek to

$$\vec{\Omega}\cdot\vec{\nabla}\Psi + \frac{1}{\epsilon}\Psi = \left(\frac{1}{\epsilon} - \epsilon\right)\frac{\Phi}{4\pi} + \frac{\epsilon}{4\pi}, \tag{1.95}$$

and

$$\frac{\epsilon}{3}\nabla^2\Phi + \epsilon\Phi = \epsilon, \tag{1.96}$$

respectively.

1.7.2 SIP used as a Diffusion Solver

We first wish to know how an interior penalty form of the diffusion equation will perform on unstructured polyhedral grids. The SIP diffusion formulation has previously been analyzed for use as a DFEM diffusion solver for unstructured 2D polygonal grids [26]. The MIP DSA form has also been successfully utilized for unstructured 2D polygonal grids [22]. Here, we first seek to extend the efficacy of the SIP form as a diffusion solver on polyhedral mesh cellls. We will do this by analyzing the following two problem types:

1. An exactly-linear solution to determine if linear basis functions will span the solution space;

2. The Method of Manufactured Solutions (MMS) to test basis function convergence rates.

The polyhedral mesh types that we employ for this analysis are presented in Section 1.7.2.1. Next, the exactly-linear solution analysis is performed in Section 1.7.2.2. Finally, the MMS analysis to confirm the second order convergence rates of the 3D PWL basis functions is presented in Section 1.7.2.3.

1.7.2.1 Geometry Specification for the SIP Problems

To analyze the SIP diffusion form on 3D polyhedral grids, we will utilize many of the 2D polygonal grids that were used in Section **??**. For this analysis we will resuse the cartesian, ordered-triangular, polygonal sinusoidal, and polygonal-z meshes. We

will also use purely-randomized polygonal grids formed from Voronoi mesh generation as outlined in Section **??**. To form the needed 3D polyhedral grids, we will take these 2D grids and simply extrude the meshes in the z-dimension.

Figure 1.3 provides the 2D mesh types that will be utilized in this analysis. Figure 1.4 then provides the same meshes after they have been extruded in the z-dimension. We note that it will not simply be these exact grids that are employed. For the MMS analysis in Section 1.7.2.3, various refinements of these meshes will be utilized.

1.7.2.2 Exactly-Linear Solution

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

$$\Phi(x, y, z) = \frac{4J^{inc}}{L + 4D} \left(L + 2D - y\right), \tag{1.97}$$

with the following boundary conditions in the y-direction:

$$\begin{aligned} \Phi - 2D\partial_y\Phi =& 4J^{inc}, && \forall (x, z), y = 0 \\ \Phi + 2D\partial_y\Phi =& 0, && \forall (x, z), y = L \end{aligned} . \tag{1.98}$$

In Eqs. (1.97) and (1.98), D is once again the standard diffusion coefficient and J^{inc} is the incoming current on the $y = 0$ boundary. For this analysis, we choose D to be 2, J^{inc} to be 9, and L to be 1. Using Eq. (1.97), our solution has a value of 20 at $y = 0$ and linearly-decreases to 16 at $y = 1$.

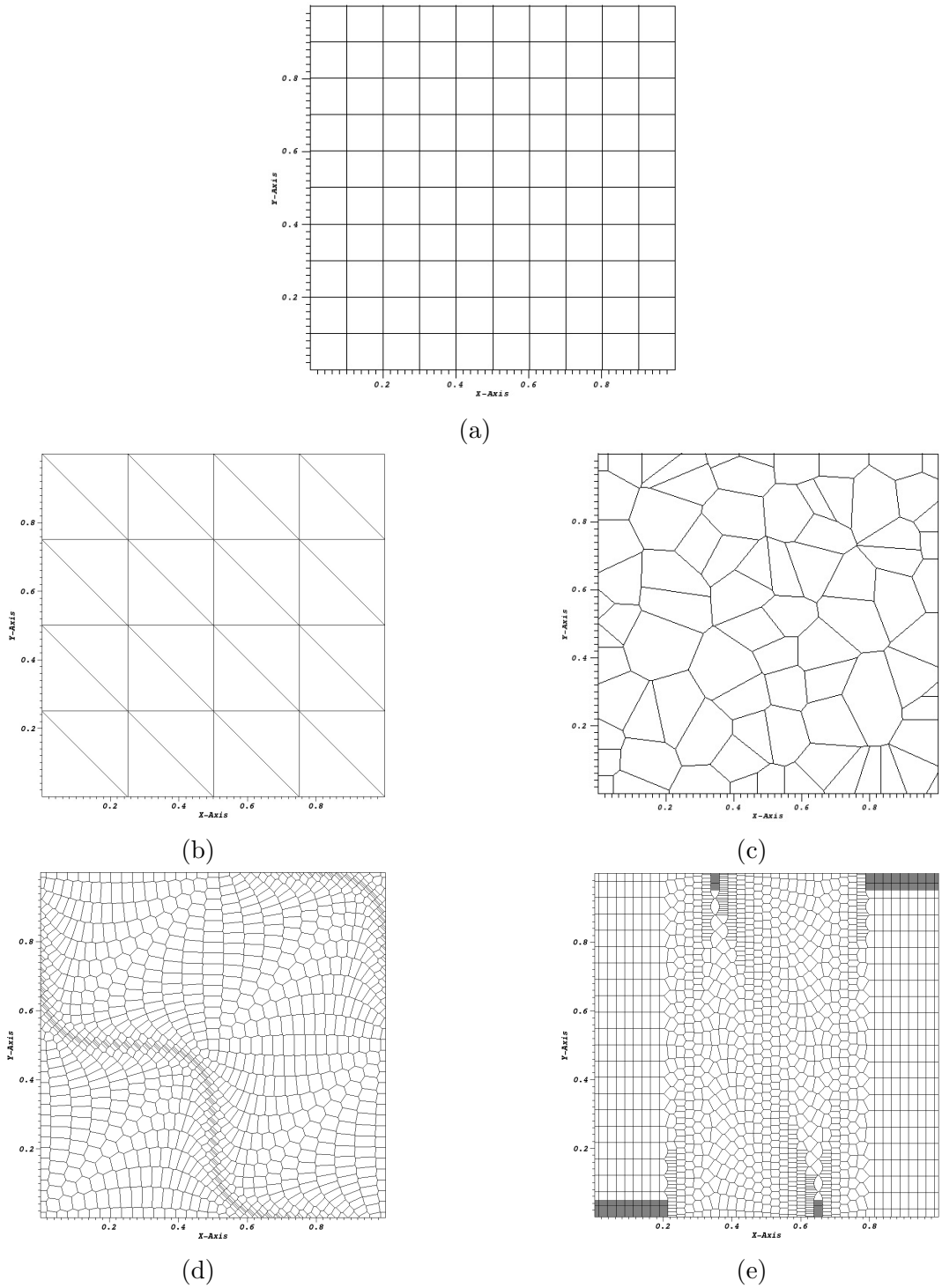
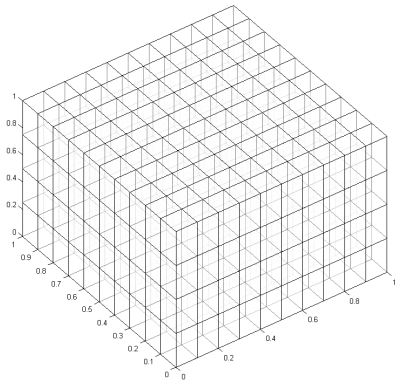
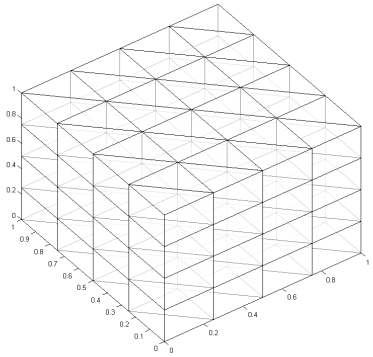


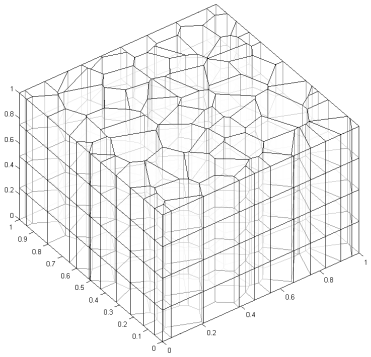
Figure 1.3: 2D grids to be extruded of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.



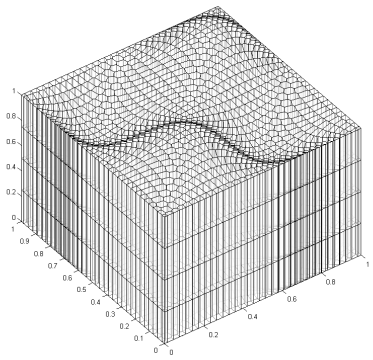
(a)



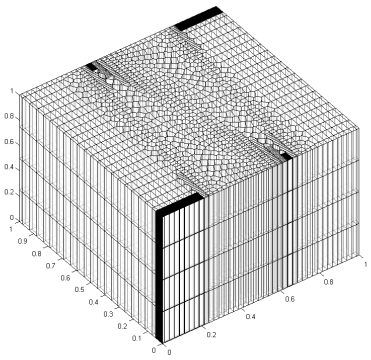
(b)



(c)



(d)



(e)

Figure 1.4: Extrusion of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

Using the 3D PWL basis functions, which were previously used for SIP on 2D polygons [26], the linear solutions are presented in Figure 1.5 at the midplane axial slice of $z = L/2$. We can see from the exact contour lines in the plots that SIP can capture an exactly-linear solution even on some highly distorted polyhedral grids.

1.7.2.3 Method of Manufactured Solutions

We next test to see if the SIP diffusion form can capture the appropriate second order convergencerates with the 3D PWL basis functions. These basis functions were previously shown to capture the appropriate convergence rates for the DGFEM transport equation on 3D hexahedral grids [37].

$$\Phi^{quad}(x, y, z) = x(1 - x)y(1 - y)z(1 - z) \tag{1.99}$$

$$\Phi^{gauss}(x, y, z) = \Phi^{quad}(x, y, z) \exp(-(\vec{r} - \vec{r}_0) \cdot (\vec{r} - \vec{r}_0)^T) \tag{1.100}$$

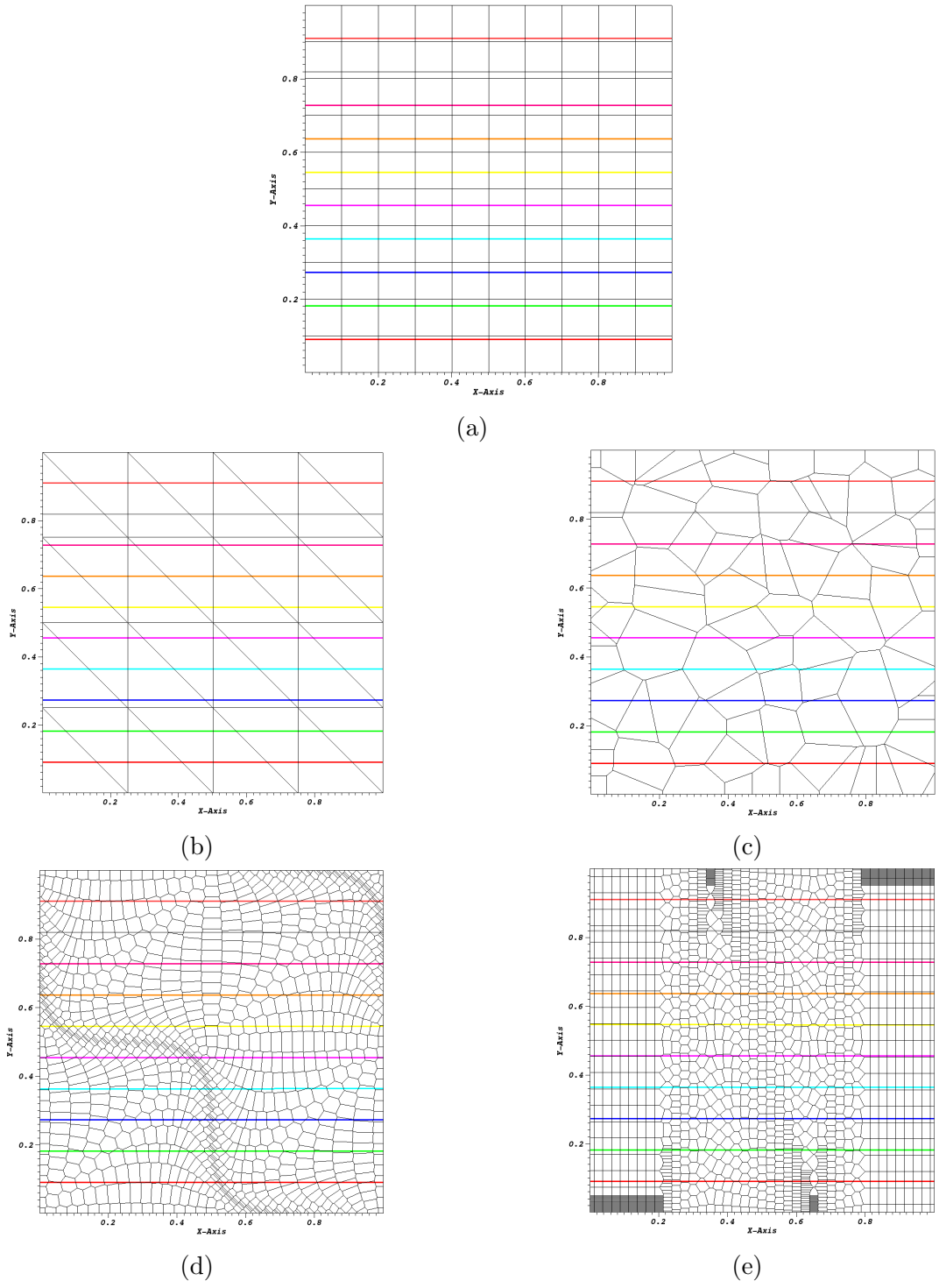
1.7.3 1 Group DSA Analysis

1.7.3.1 2D Homogeneous Medium Case

1.7.3.2 3D Homogeneous Medium Case

1.7.3.3 Periodic Horizontal Interface Problem

When DSA is applied to multidimensional problems (2D and 3D), the preconditioning of the transport operators can degrade in the presence of heterogeneous configurations with large material discontinuities [38]. The Periodic Horizontal Interface (PHI) problem is considered a litmus test for heterogeneous DSA techniques. This problem consists of horizontal strips of alternating optically thick and optically thin materials that are 1 cell in depth. We define σ_1 as the optically thick total cross



(a)

(b)

(c)

(d)

(e)

Figure 1.5: Axial slice showing the contours for the linear solution of the different mesh types: (a) cartesian, (b) ordered triangles, (c) random polygons, (d) sinusoidal polygons, and (e) polygonal z-mesh.

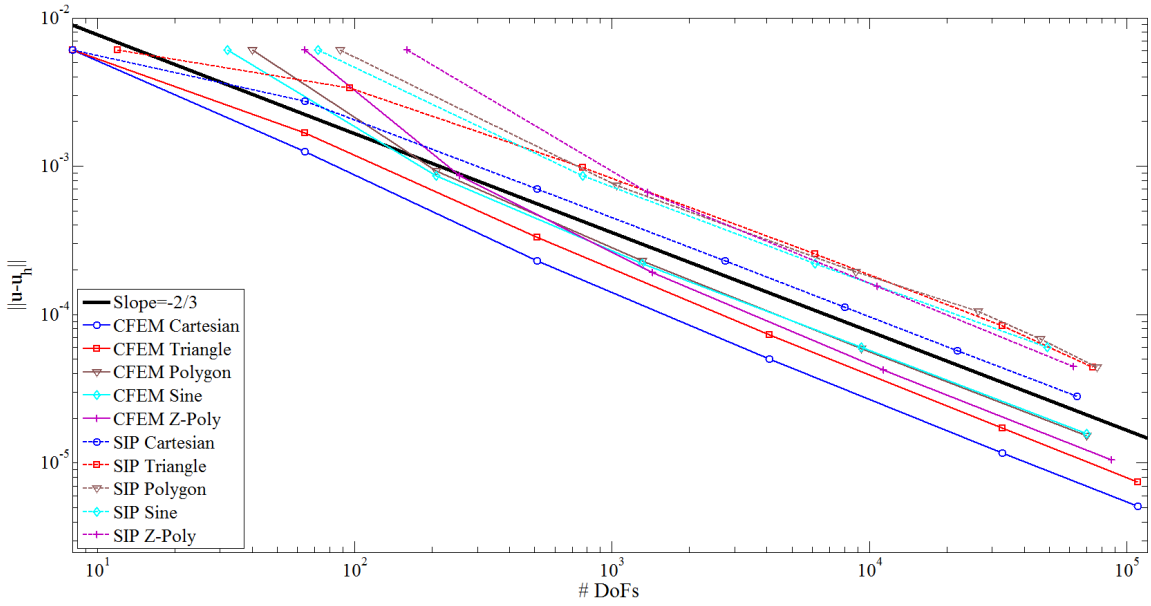


Figure 1.6: blah

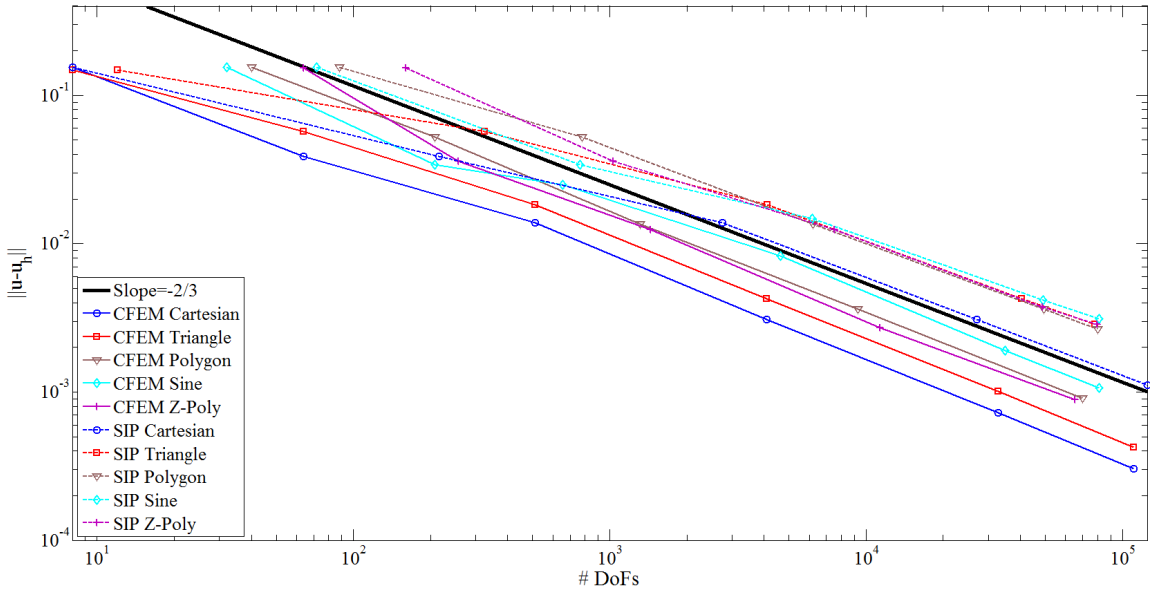


Figure 1.7: blah

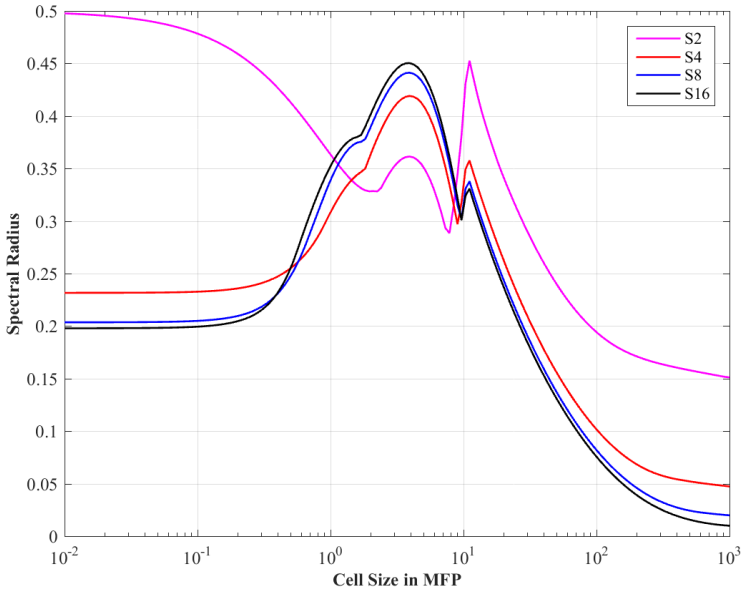


Figure 1.8: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear Wachspress coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

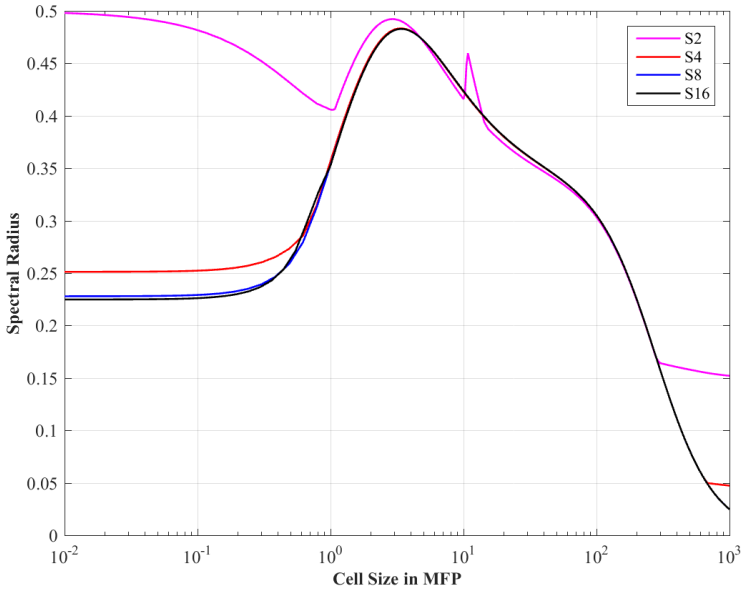


Figure 1.9: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear PWL coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

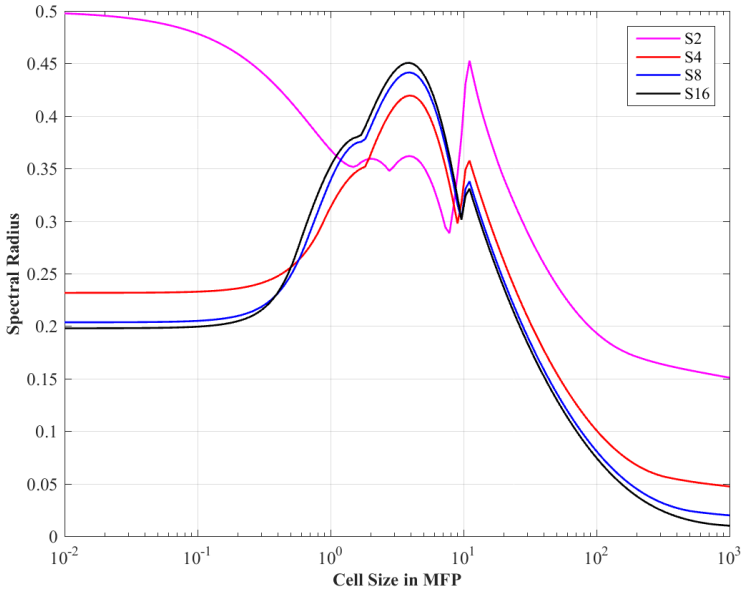


Figure 1.10: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear mean value coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.

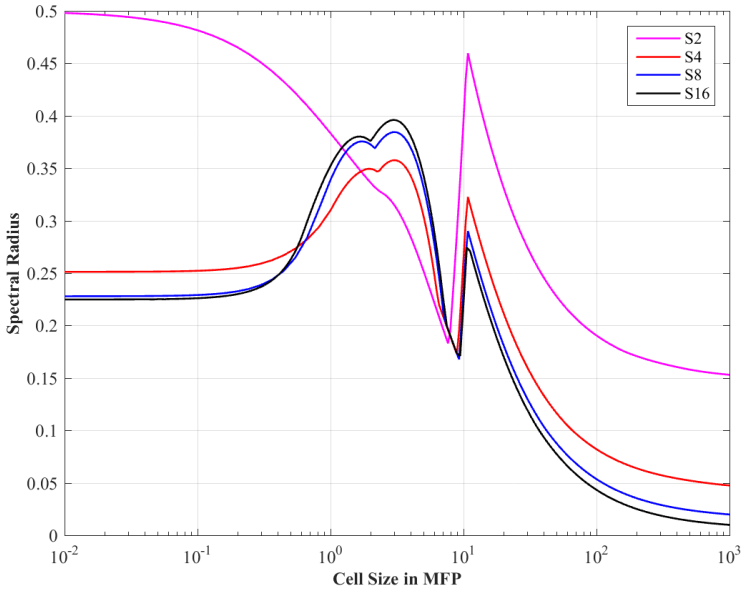
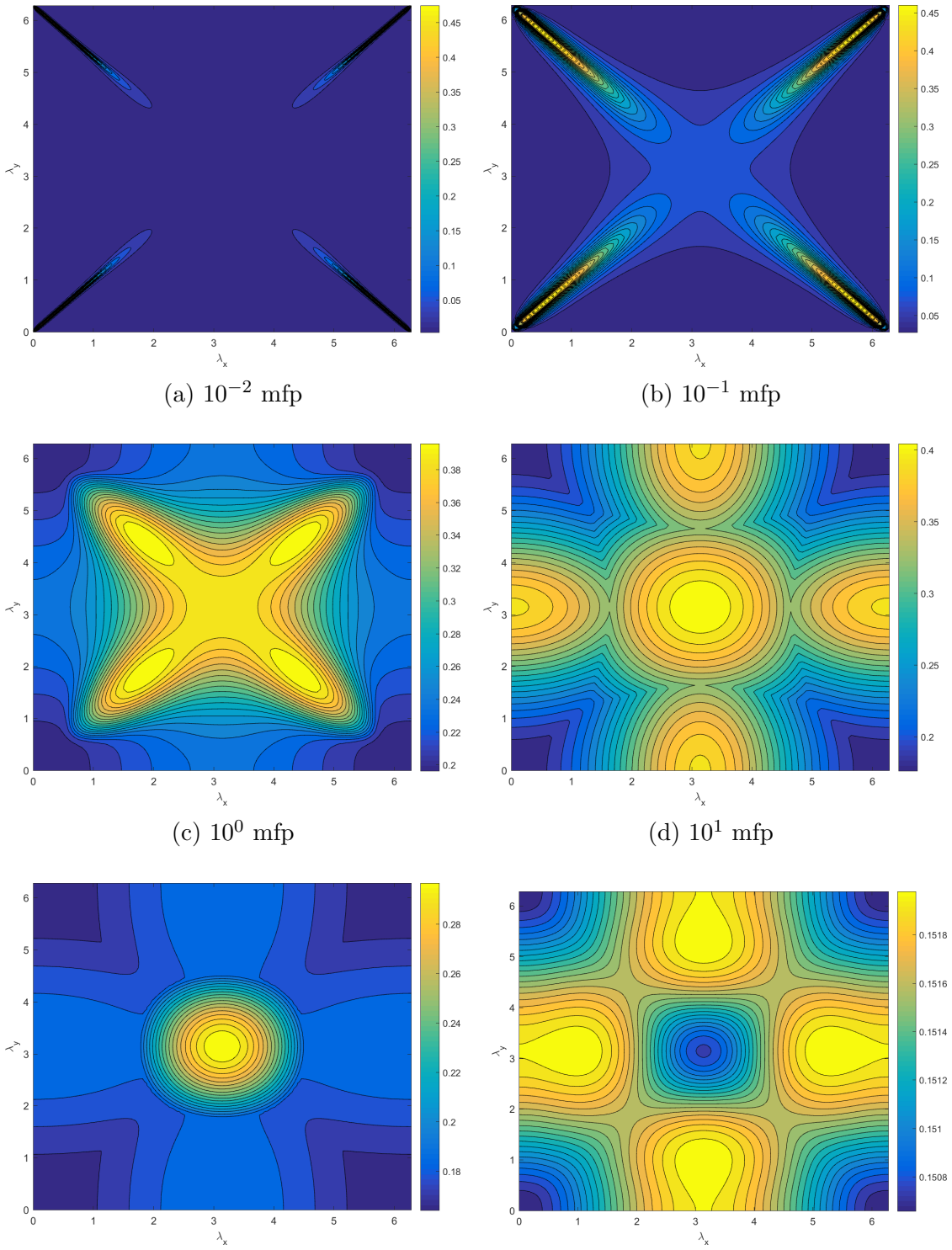


Figure 1.11: Fourier analysis of the 2D MIP form with $c = 4$ and using the linear maximum entropy coordinates for the homogeneous infinite medium case as a function of the mesh optical thickness.



(a) 10^{-2} mfp

(b) 10^{-1} mfp

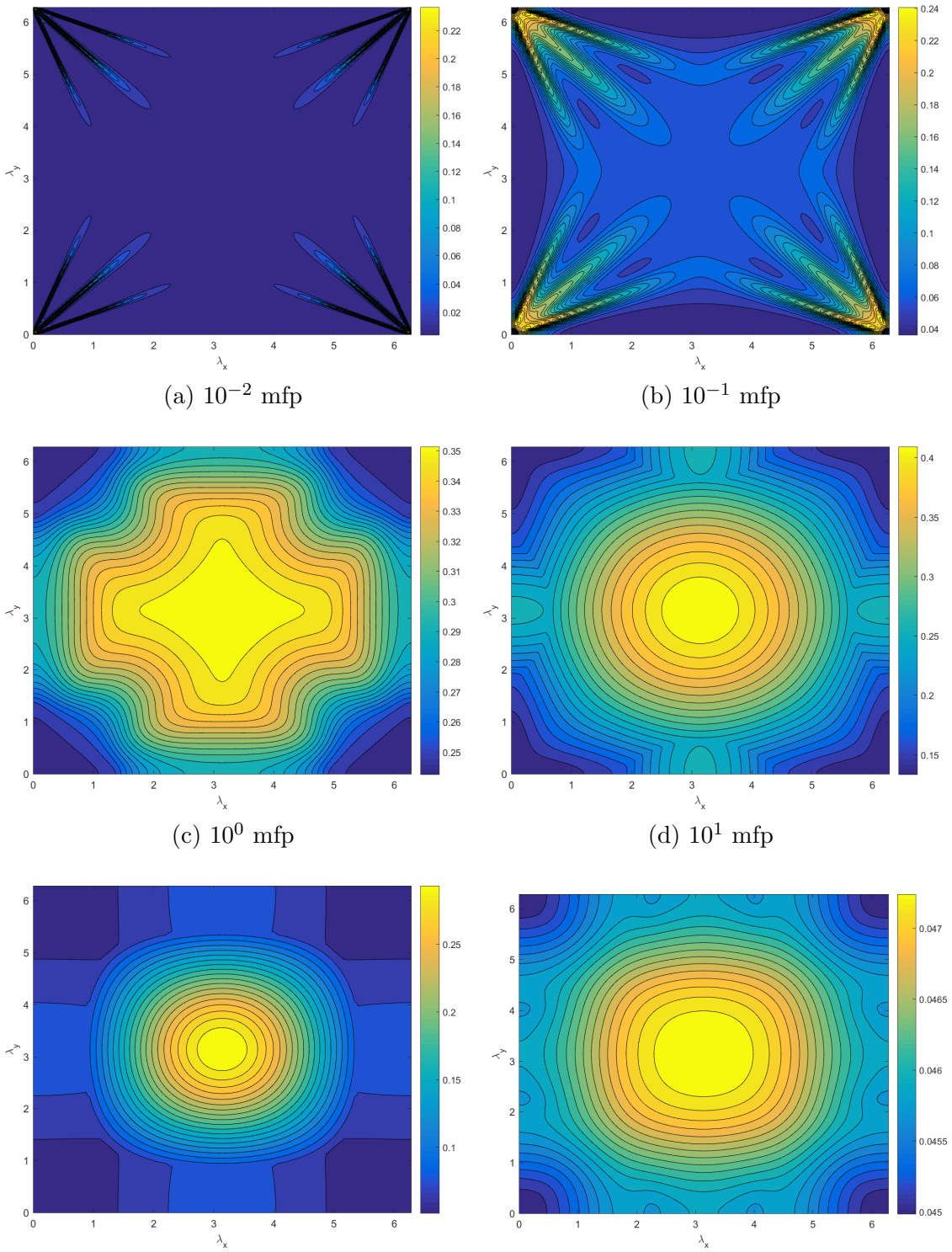
(c) 10^{0} mfp

(d) 10^{1} mfp

(e) 10^{2} mfp

(f) 10^{3} mfp

Figure 1.12: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS2 quadrature.



(a) 10^{-2} mfp

(b) 10^{-1} mfp

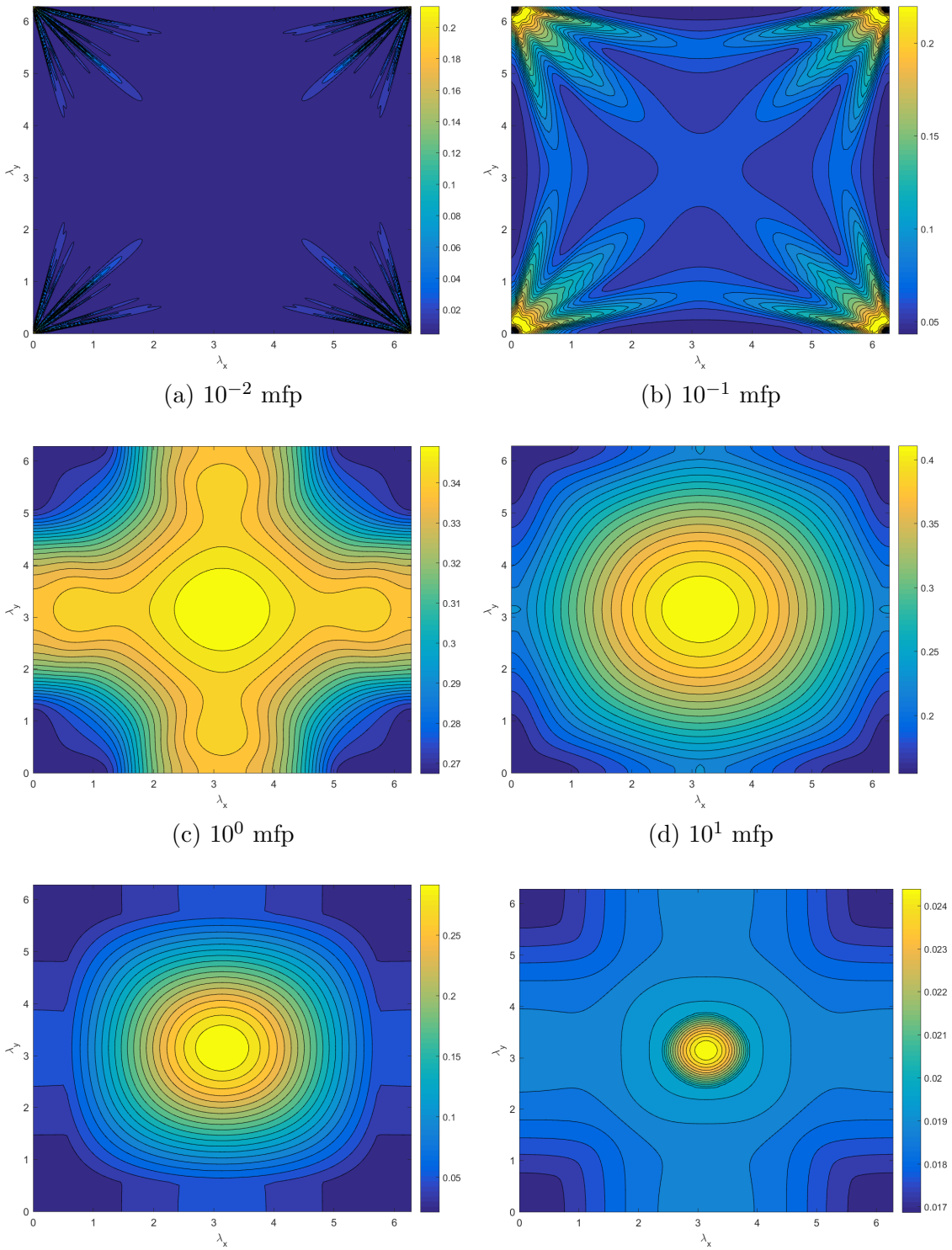
(c) 10^{0} mfp

(d) 10^{1} mfp

(e) 10^{2} mfp

(f) 10^{3} mfp

Figure 1.13: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS4 quadrature.



(a) 10^{-2} mfp

(b) 10^{-1} mfp

(c) 10^{0} mfp

(d) 10^{1} mfp

(e) 10^{2} mfp

(f) 10^{3} mfp

Figure 1.14: Fourier wave number distribution for different mesh optical thicknesses of a single 2D square cell with PWL coordinates and LS8 quadrature.

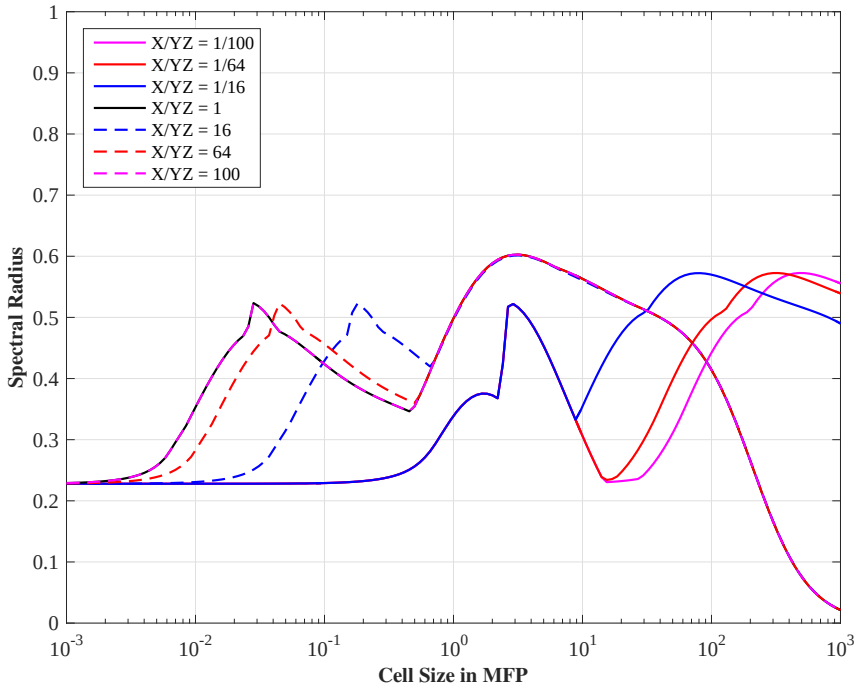


Figure 1.15: Fourier spectral radii for MIP with 3D PWL coordinates with different aspect ratios and $c = 1$.

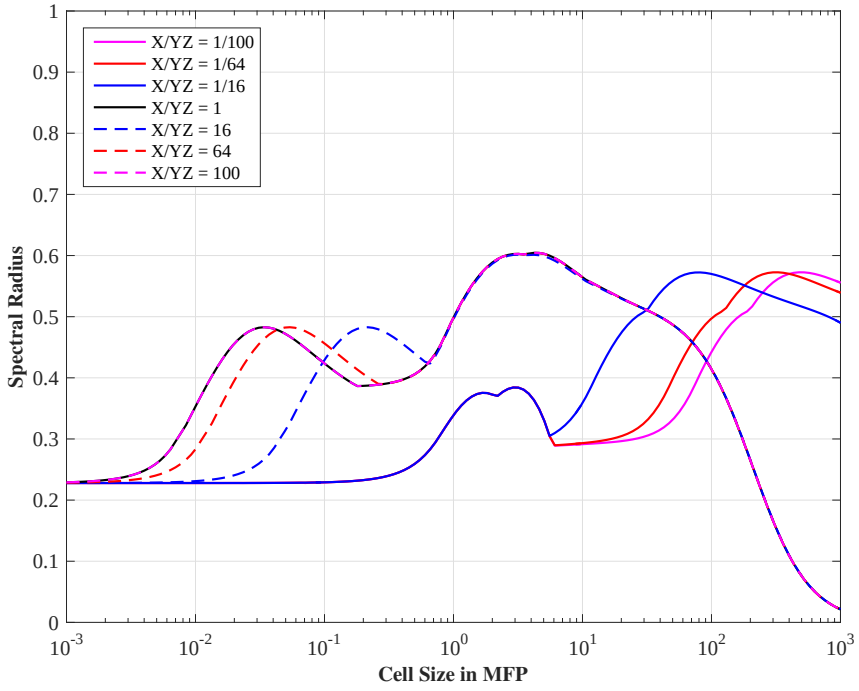


Figure 1.16: Fourier spectral radii for MIP with 3D PWL coordinates with different aspect ratios and $c = 4$.

section and σ_2 as the optically thin total cross section. We then define a tuning parameter, σ, and allow the region cross sections to have the following form:

$$\begin{aligned} \sigma_1 &= \sigma \\ \sigma_2 &= \frac{1}{\sigma}. \end{aligned} \tag{1.101}$$

We can see that increasing the value of σ will increase the magnitude of difference between the two region cross sections. Therefore, as σ grows large, the material discontinuities will grow which could potentially reduce the performance of our DSA scheme.

From the analysis presented in Section 1.7.3.1, we showed that our different 2D linear and quadratic basis functions were robust and stable, even for mesh cells with large aspect ratios. For this PHI analysis, we concentrate on analyzing just the linear PWL coordinates as our basis functions. Just like before, we will also examine different level-symmetric quadrature sets and their effects problems with varying optical thickness. The optical thickness and diffusivity of the problem is increased by varying both σ and the scattering ratio, c. This study was conducted with the sequence, $\sigma = [10, 20, 40, 80, 160, 320, 640]$, and with following scattering ratios: $c = [0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999]$.

The full results of this PHI analysis are presented in Tables 1.4 - 1.7 for the LS2, LS4, LS8, and LS16 quadratures, respectively. From these tables, we can see that MIP DSA loses its effectiveness as the heterogeneity and overall diffusivity of the problem increases. The theoretical spectral radii are greater than any of the values presented for the homogeneous case from Figure 1.9. This result is true even for the example with the smallest heterogeneity and smallest diffusivity ($\sigma = 10$ and $c = 0.9$). We can gain some more knowledge of the DSA degradation by observing the eigenvalue dependency on the fourier wave numbers for our problems. Figure

Table 1.4: Spectral radius for the 2D PHI problem with the PWL basis functions and LS2 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.77091	0.89908	0.91279	0.91417	0.91431	0.91432
20	0.82038	0.94425	0.95859	0.96003	0.96018	0.96019
40	0.84803	0.96525	0.97982	0.98129	0.98144	0.98145
80	0.86188	0.97491	0.98955	0.99102	0.99117	0.99119
160	0.87134	0.98003	0.99409	0.99557	0.99571	0.99573
320	0.87833	0.98353	0.99626	0.99774	0.99789	0.99790
640	0.88187	0.98533	0.99732	0.99880	0.99894	0.99896

1.17 provides the eigenvalue distribution based off the fourier wave numbers for the different quadrature sets for $\sigma = 10$ and $c = 0.9$. Figure 1.18 then provides the same information for $\sigma = 640$ and $c = 0.9999$.

1.7.3.4 Performance of MIP DSA with Adaptive Mesh Refinement

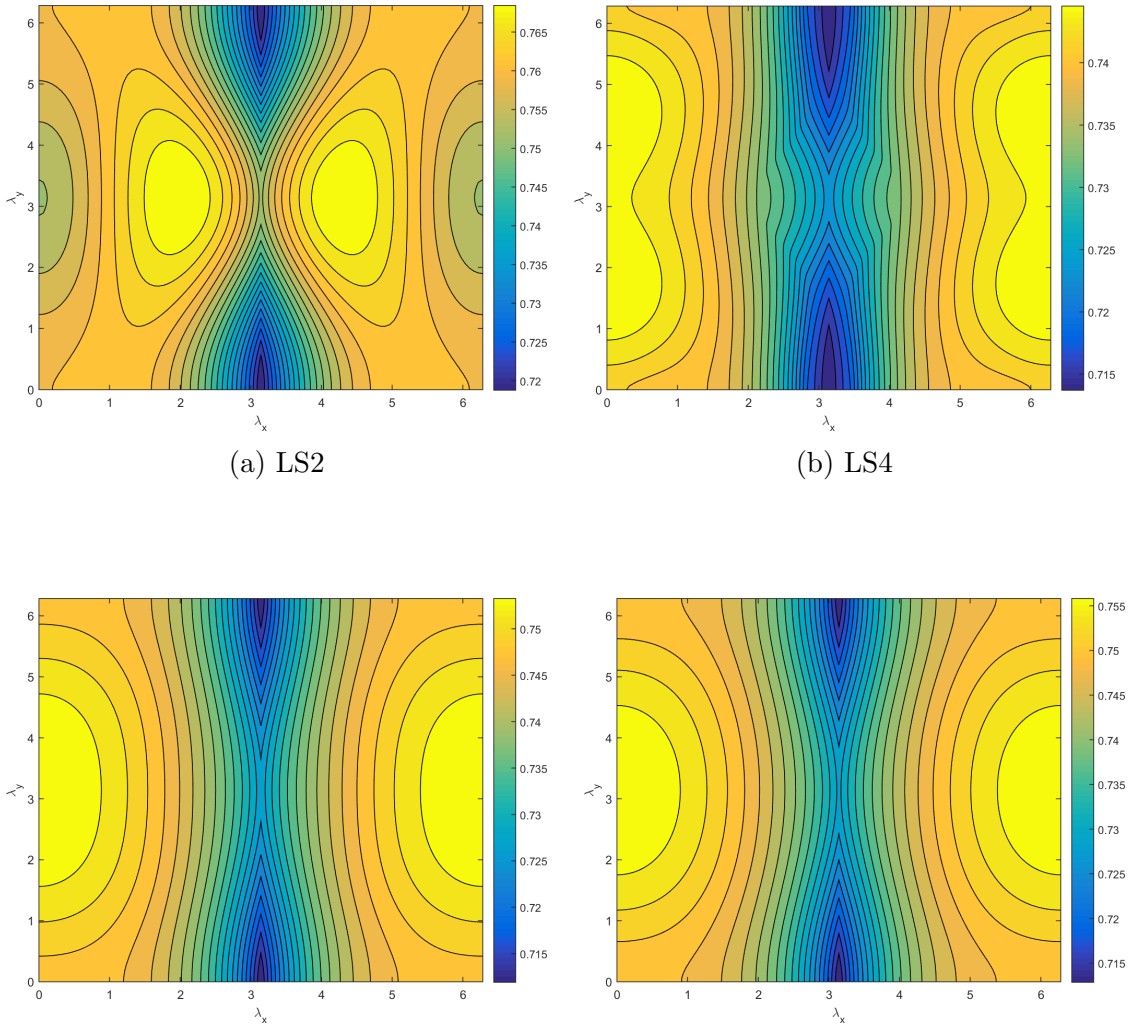
For the final theoretical analysis of DSA with the MIP form, we analyze the acceleration performance when AMR is utilized on a sufficiently optically thick transport with material discontinuities. The 2D transport problem that will be examined is similar to the Iron-Water problem [39]. It was modified by Wang and Ragusa for use with higher-order basis functions on triangular meshes with hanging nodes [20]. We will reexamine their work on degenarate polygonal grids and not use hanging nodes. The complete geometric desciption of our problem including boundary conditions and material distributions is given in Figure 1.19. The material properties for each

Table 1.5: Spectral radius for the 2D PHI problem with the PWL basis functions and LS4 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.74609	0.87284	0.88970	0.89148	0.89166	0.89167
20	0.81135	0.93358	0.95001	0.95179	0.95197	0.95199
40	0.84353	0.96104	0.97612	0.97781	0.97799	0.97800
80	0.85865	0.97342	0.98785	0.98943	0.98960	0.98961
160	0.86998	0.97913	0.99335	0.99482	0.99497	0.99499
320	0.87767	0.98307	0.99593	0.99739	0.99753	0.99755
640	0.88166	0.98500	0.99717	0.99863	0.99878	0.99879

Table 1.6: Spectral radius for the 2D PHI problem with the PWL basis functions and LS8 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.75532	0.87917	0.89780	0.89989	0.90010	0.90012
20	0.81754	0.93694	0.95380	0.95581	0.95602	0.95604
40	0.84681	0.96337	0.97793	0.97969	0.97988	0.97990
80	0.86032	0.97459	0.98898	0.99042	0.99057	0.99058
160	0.86980	0.98007	0.99394	0.99540	0.99554	0.99556
320	0.87795	0.98354	0.99623	0.99768	0.99783	0.99784
640	0.88204	0.98524	0.99732	0.99878	0.99892	0.99894



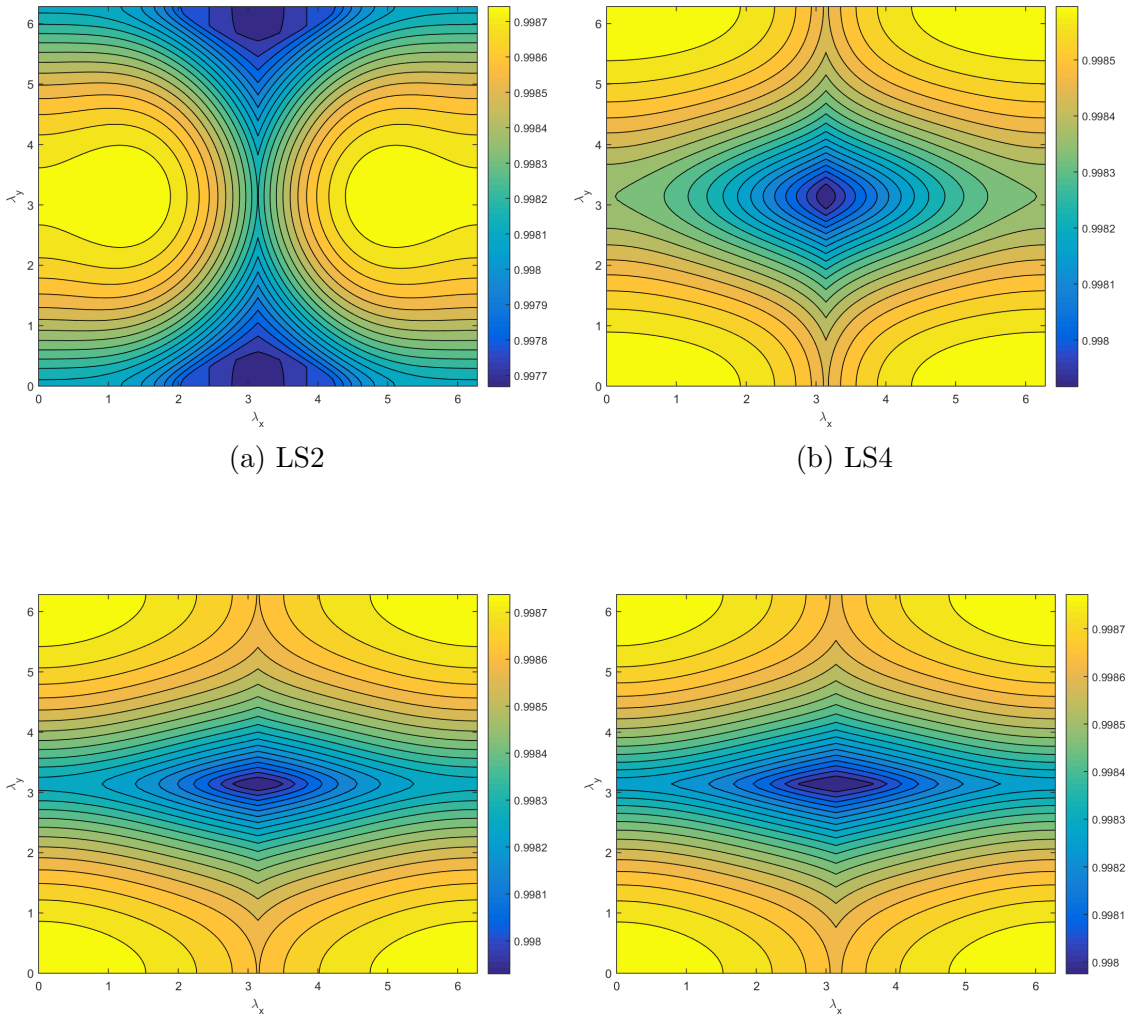
(a) LS2

(b) LS4

(c) LS8

(d) LS16

Figure 1.17: Fourier wave number distribution for the 2D PHI problem with $\sigma = 10$ and $c = 0.9$ and different level-symmetric quadratures.



(a) LS2

(b) LS4

(c) LS8

(d) LS16

Figure 1.18: Fourier wave number distribution for the 2D PHI problem with $\sigma = 640$ and $c = 0.9999$ and different level-symmetric quadratures.

Table 1.7: Spectral radius for the 2D PHI problem with the PWL basis functions and LS16 quadrature.

	Scattering ratios					
σ	0.9	0.99	0.999	0.9999	0.99999	0.99999
10	0.75796	0.88052	0.89888	0.90097	0.90119	0.90121
20	0.81903	0.93790	0.95455	0.95651	0.95671	0.95673
40	0.84758	0.96387	0.97842	0.98015	0.98033	0.98035
80	0.86070	0.97485	0.98924	0.99069	0.99083	0.99085
160	0.87025	0.98029	0.99407	0.99553	0.99567	0.99569
320	0.87817	0.98365	0.99629	0.99775	0.99790	0.99791
640	0.88216	0.98530	0.99735	0.99881	0.99896	0.99897

Table 1.8: Material definitions and physical properties for the Iron-Water problem.

Region	σ_t (cm^{-1})	c	S ($cm^{-3}sec^{-1}$)
I	1.0	0.90	1.0
II	1.5	0.96	0.0
III	1.0	0.30	0.0

region which include the total cross section, scattering ratio, and source strength are given in Table 1.8. Scattering is isotropic.

1.7.4 Scalability of the MIP DSA Preconditioner

So far, we have presented a detailed theoretical analysis of DSA preconditioning with the MIP diffusion form. We have shown that the different 2D and 3D basis functions provided in Chapter **??** are robust and stable, even on mesh cells with

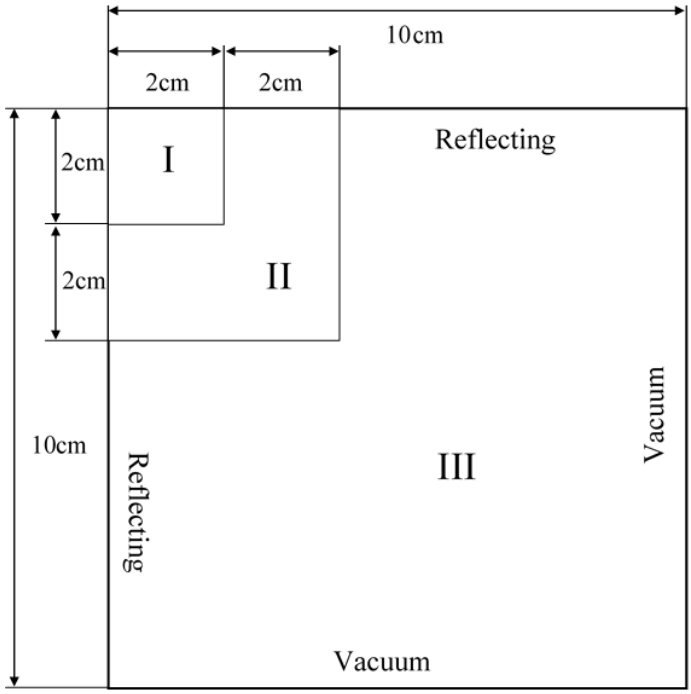


Figure 1.19: Geometry description for the Iron-Water problem.

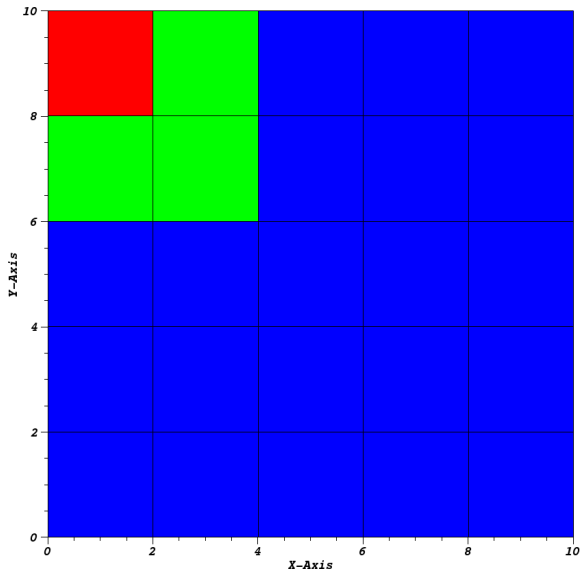
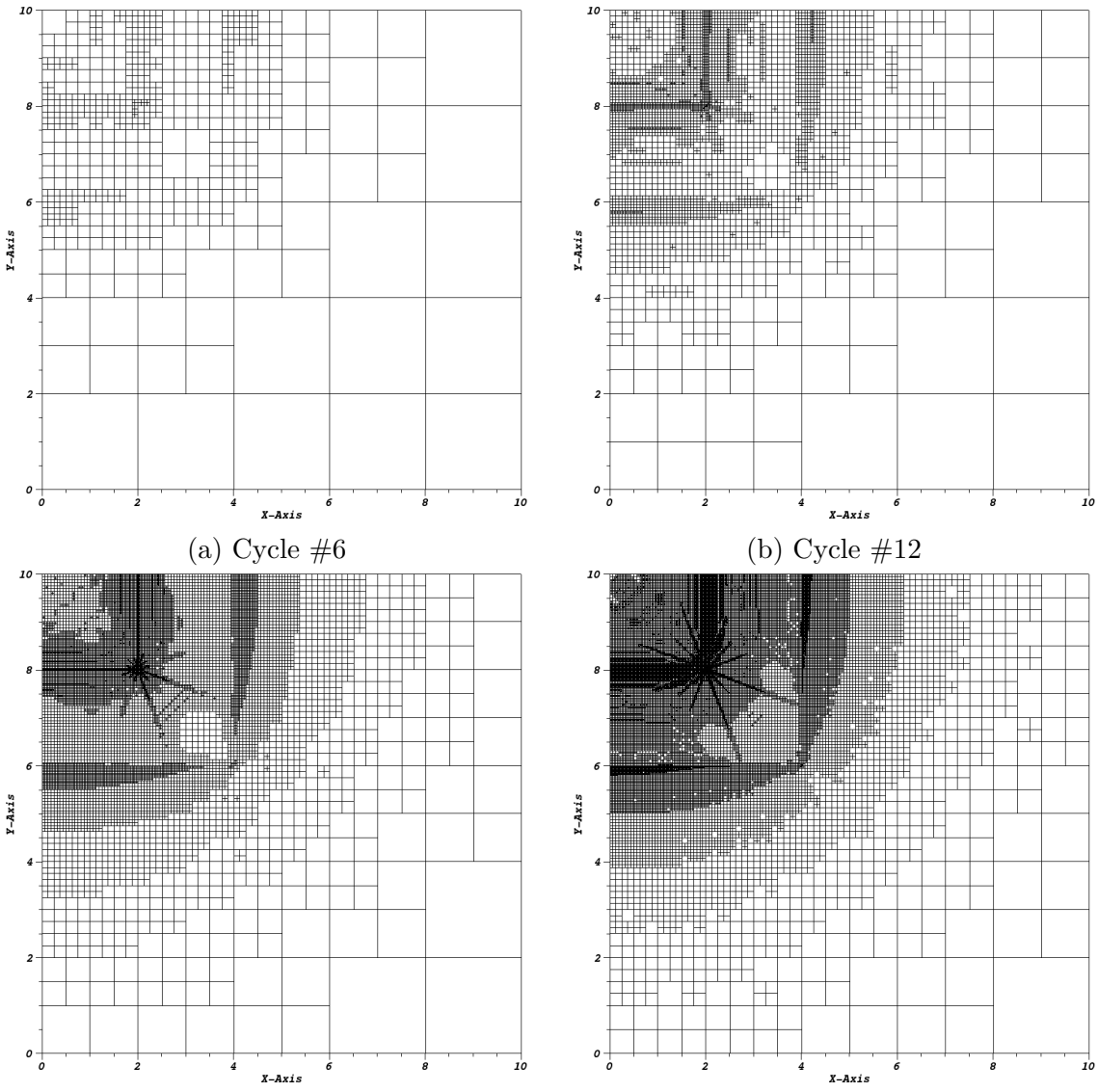


Figure 1.20: Initial mesh for the Iron-Water problem.



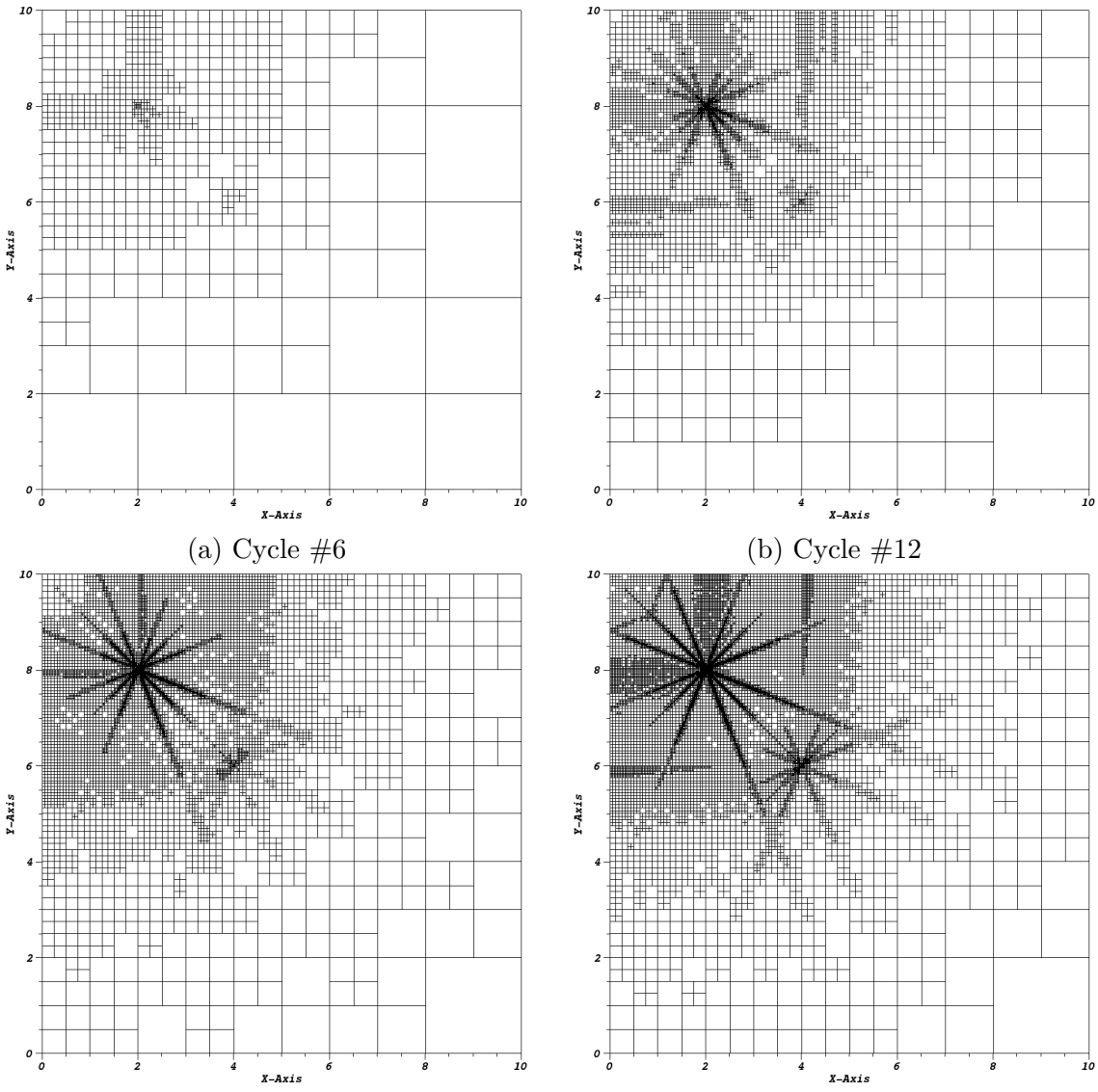
(a) Cycle #6

(b) Cycle #12

(c) Cycle #18

(d) Cycle #24

Figure 1.21: Meshes for the Iron-Water problem using the linear PWL coordinates and LS4 quadrature.



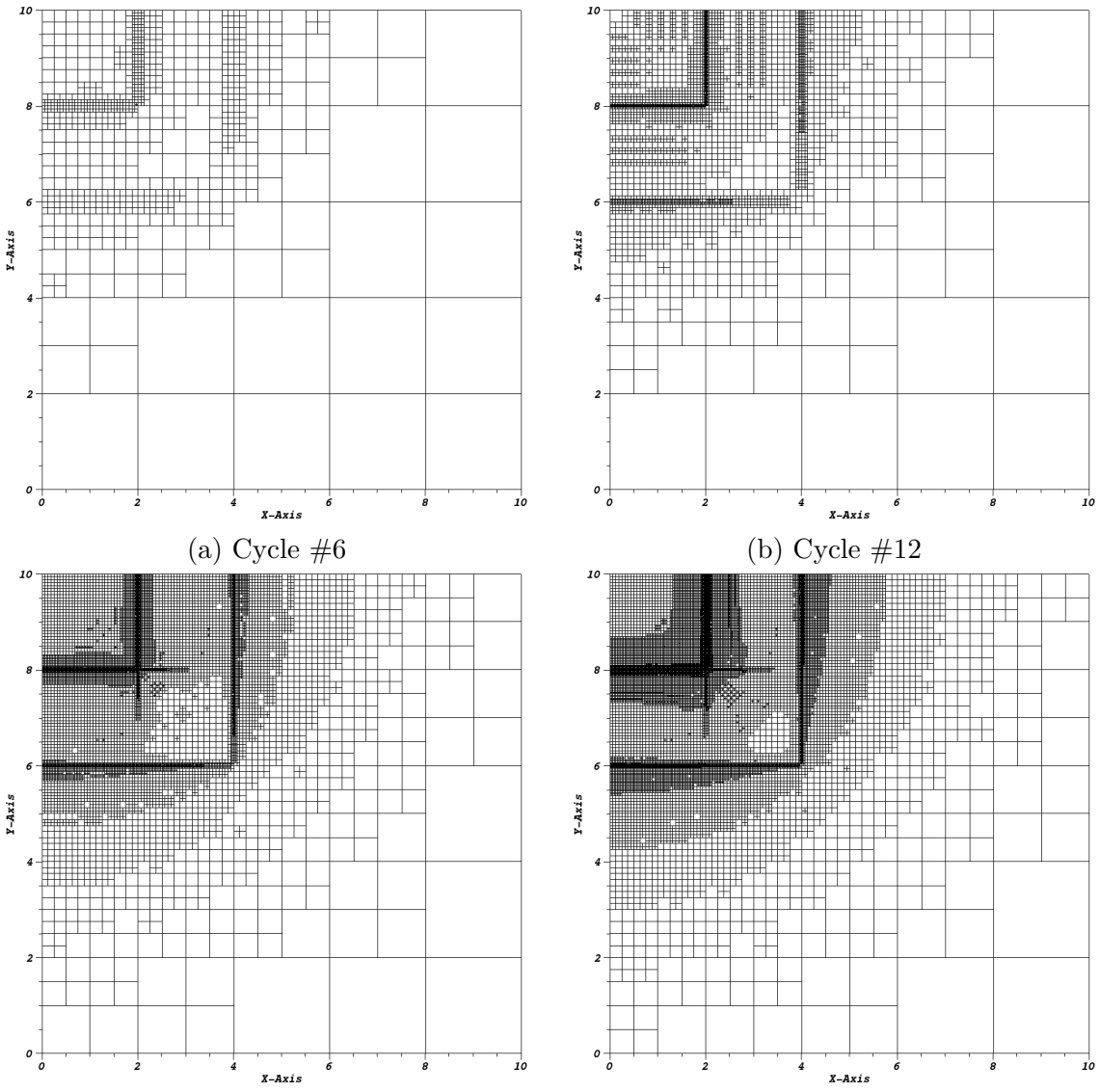
(a) Cycle #6

(b) Cycle #12

(c) Cycle #18

(d) Cycle #24

Figure 1.22: Meshes for the Iron-Water problem using the quadratic PWL coordinates and LS4 quadrature.



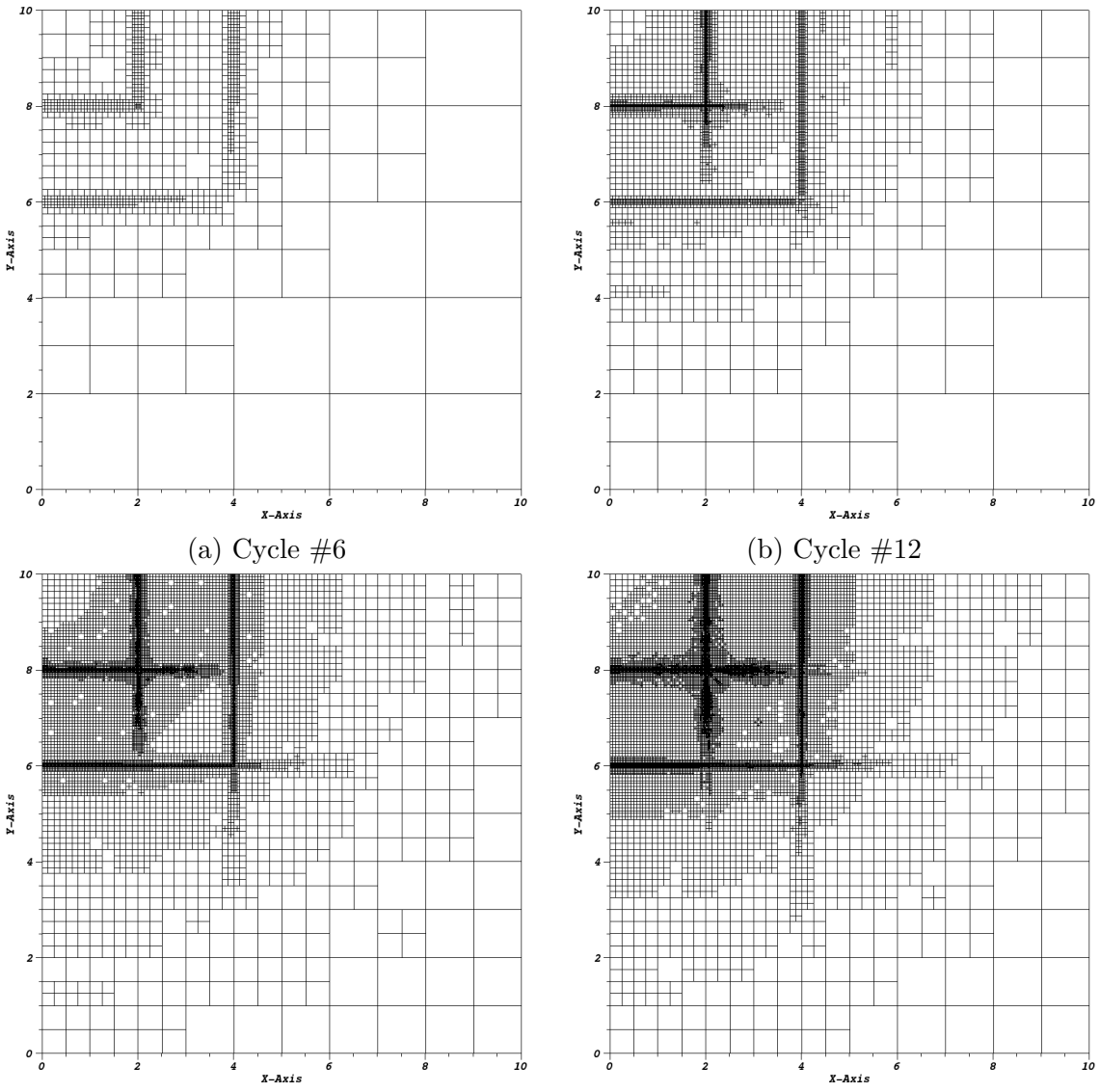
(a) Cycle #6

(b) Cycle #12

(c) Cycle #18

(d) Cycle #24

Figure 1.23: Meshes for the Iron-Water problem using the linear PWL coordinates and S_{24}^2 PGLC quadrature.



(a) Cycle #6

(b) Cycle #12

(c) Cycle #18

(d) Cycle #24

Figure 1.24: Meshes for the Iron-Water problem using the quadratic PWL coordinates and S_{24}^2 PGLC quadrature.

high aspect ratios. We also demonstrated that problems with large heterogeneous configurations can diminish the effectiveness of MIP DSA.

Now, we need to demonstrate the scalability of MIP DSA preconditioning onto large massively-parallel computer architectures. Our ability to utilize this transport acceleration form would be greatly minimized if the DSA solve times scaled at a much worse rate compared to the transport sweep times. We specifically use the low-order diffusion operator because it is supposed to be easier to invert than the full transport operator.

From the results of the Iron-Water problem in Section 1.7.3.4, we observed that the PCG iteration counts did not appreciably grow when utilizing AMG as the preconditioner for the diffusion solve. This was in direct contrast to the simpler preconditioners (Jacobi, GS, and ILU) that had their iteration counts grow rapidly as the number of unknowns increased. Motivated by the efficiency of AMG methods with the MIP diffusion form for the simple Iron-Water AMR problem, we will continue to use AMG as the diffusion preconditioner for our massively-parallel calculations.

We have implemented the 1-group and thermal upscattering DSA methodologies of Section 1.2 with the MIP form into Texas A&M University's PDT code. It is a massively-parallel DGFEM S_N transport code that has had good sweep scaling efficiency out to $O(10^5) - O(10^6)$ processes [40, 41]. BoomerAMG of the HYPRE library is used as the AMG diffusion preconditioner [42, 43]. We next analyze the scalability of HYPRE's PCG solver with BoomerAMG preconditioning and how this performs in comparison to PDT's transport sweeping.

1.7.4.1 Weak Scaling with a Homogeneous Zerr Problem

We first test MIP's scaling with HYPRE in PDT by analyzing a simple homogenized version of the Zerr scaling problem [44]. The problem configuration is a 3D

cube that spans $[0, 16]^3$ in dimension and uses strictly orthogonal hexahedral mesh cells. The total cross section is set to $\sigma_t = 10$ with a scattering ratio of $c = 0.9999$ and a distributed source of 1 $\frac{n}{cm^3\, s}$. LS8 quadrature is employed and vacuum boundary conditions are used for all boundaries. A residual tolerance of 10^{-8} is used for the Richardson transport iterations, and a residual tolerance of 10^{-3} is used for the HYPRE PCG iterations.

Our weak scaling study is performed with a fixed 4096 spatial cells per processor as we increase the number of processors. We do not change the physical dimensions of the problem. Instead, we simply increase the number of cells in a given coordinate direction whenever we allocate more processors in that dimension. The partitioning of the processor layout (P_x, P_y, and P_z), the task aggregation (A_x, A_y, and A_z), and the number of mesh cells per dimension (N_x, N_y, and N_z) are selected to minimize the time per sweep in PDT. These parallel parameters are given for the full scaling suite in Table 1.9.

Table 1.9: Partitioning factors, aggregation factors, and cell counts per dimension for the 3D homogeneous Zerr problem run on Vulcan.

P_{tot}	P_x	P_y	P_z	A_x	A_y	A_z	N_x	N_y	N_z
1	1	1	1	16	16	1	16	16	16
8	2	2	2	16	16	1	32	32	32
64	8	4	2	8	16	1	64	64	64
512	16	16	2	8	8	1	128	128	128
1024	32	16	2	4	8	1	128	128	256
2048	32	32	2	8	4	1	256	128	256
4096	64	32	2	4	8	1	256	256	256
8192	64	64	2	4	4	1	256	256	512
16384	128	64	2	4	4	1	512	256	512
32768	128	128	2	4	4	1	512	512	512
65536	256	128	2	2	4	1	512	512	1024
131072	256	256	2	4	2	1	1024	512	1024

The timing data for this weak scaling study is given in Figure 1.25. As a function of processors used, we plot the overall solve time, the overall sweep time, the overall MIP DSA time, the time to build the MIP system matrix, the time to perform the HYPRE BoomerAMG setup call, and the time per HYPRE PCG iteration. The sweep time only constitutes the time taken to perform the actual sweep. This discounts the time needed to build the transport sources and prepare for each sweep. There are a lot of results that can be observed from this plot. We first note that all the solve times increase at the low processor counts because the number of sweeps increases from about 20 to 35 as the mesh optical thicknesses move into the intermediate range. The second note is that while the sweep times remain about the same, the various HYPRE times begin to scale more poorly as we get to $O(10^4)$ processors and higher. We can see that the HYPRE setup time especially begins to increase at a rapid pace for the highest core counts. Finally we observe that at certain processor values, the HYPRE times seem to 'spike'. This occurs at 1024, 8192, and 65536 processor counts. By looking at the problem's parallel characteristics from Table 1.9, we can see that these processor counts occur when the number of mesh cells in the z-dimension is doubled. This means that the distribution of the cell sets on a processor look like an even longer and skinnier column. This behavior is analyzed in greater detail shortly in Section 1.7.4.2.

From the timing results of Figure 1.25, one must wonder about the scalability of solving the MIP equation at even higher processor counts out to $O(10^6)$ or $O(10^7)$. For this simple problem, our diffusion solver was taking about 50%-60% of the solve time. However, our transport problem was extremely coarse in angle and only 1 energy group. This leads to low parallel concurrency for the phase-space that we solve for in one transport sweep. This means that for a much larger 3D transport problem with many energy groups ($O(10^2)$) and many quadrature angles ($O(10^3)$),

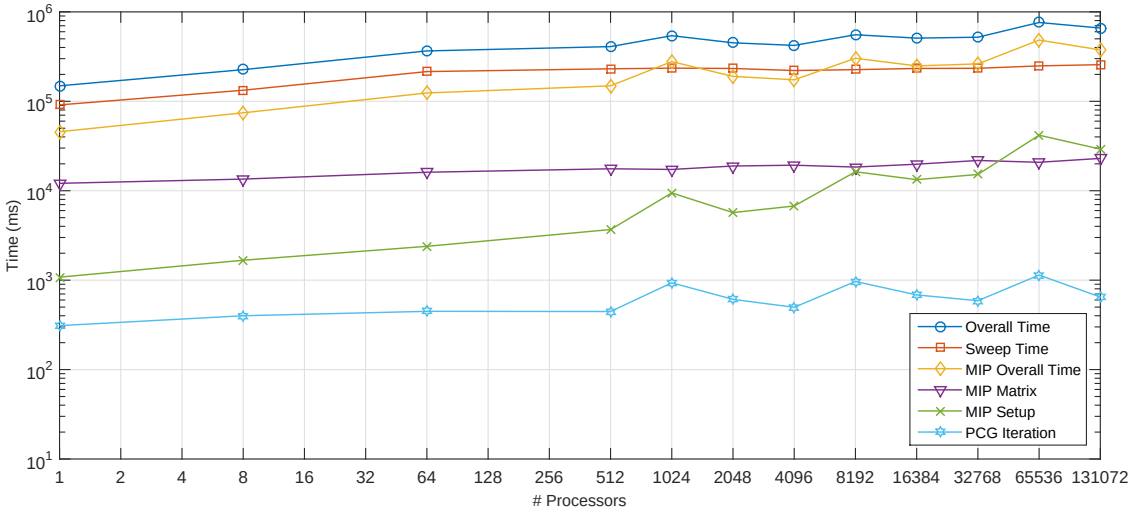


Figure 1.25: blah

the fraction of time that is spent performing DSA calculations will become negligibly small. In fact, we will show a little later that our energy-collapsed thermal neutron upscattering methodologies yield DSA solve times that are not noticible.

1.7.4.2 Aggregation and Partitioning Effects on the HYPRE PCG Algorithm

1.7.5 Thermal Neutron Upscattering Acceleration

We conclude the results of this chapter by presenting analysis on the ability to use DSA to accelerate the convergence of multigroup transport problems that are dominated by thermal neutron upscattering.

1.8 Conclusions

In this chapter, we analyzed the Modified Interior Penalty form of the diffusion equation for use as the diffusion solver for DSA preconditioning of the DGFEM S_N transport equation.

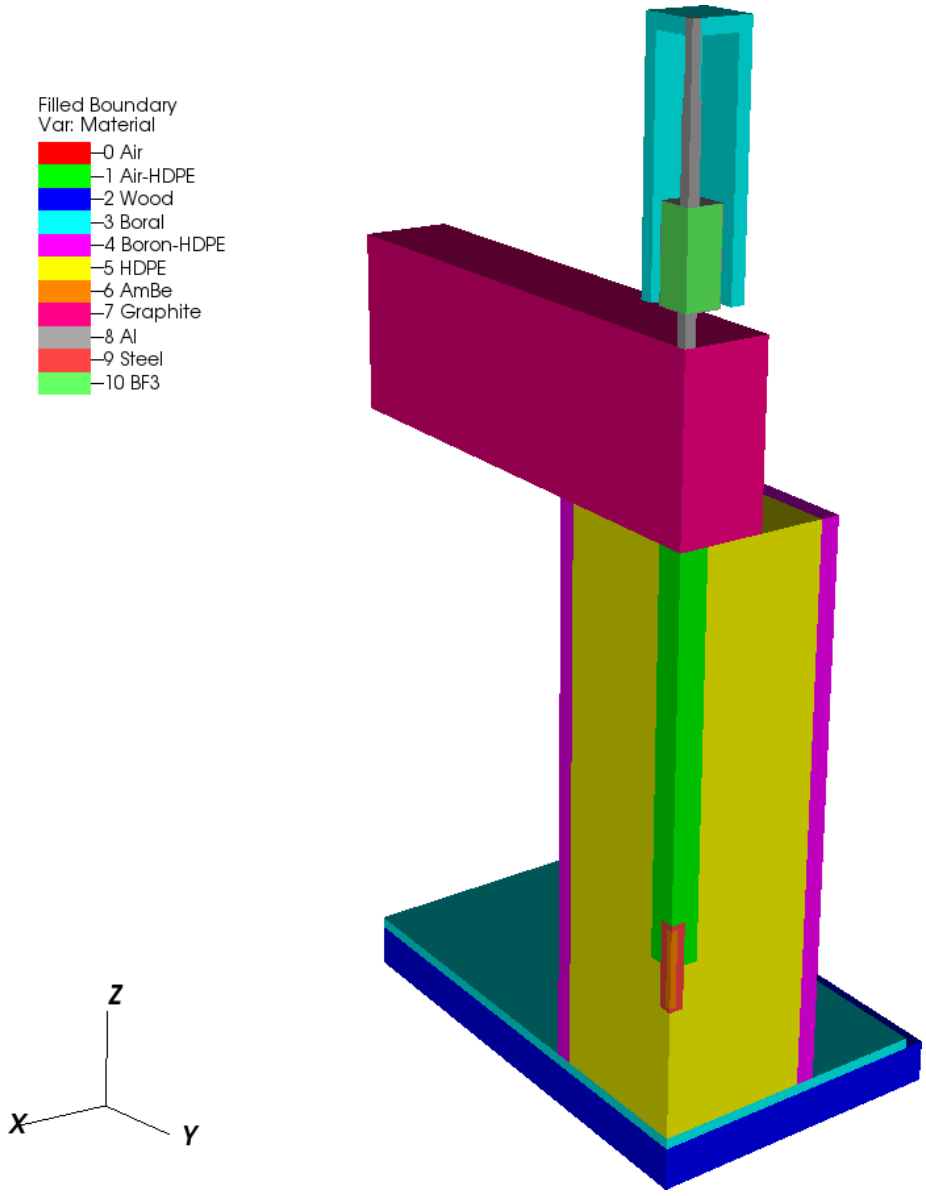
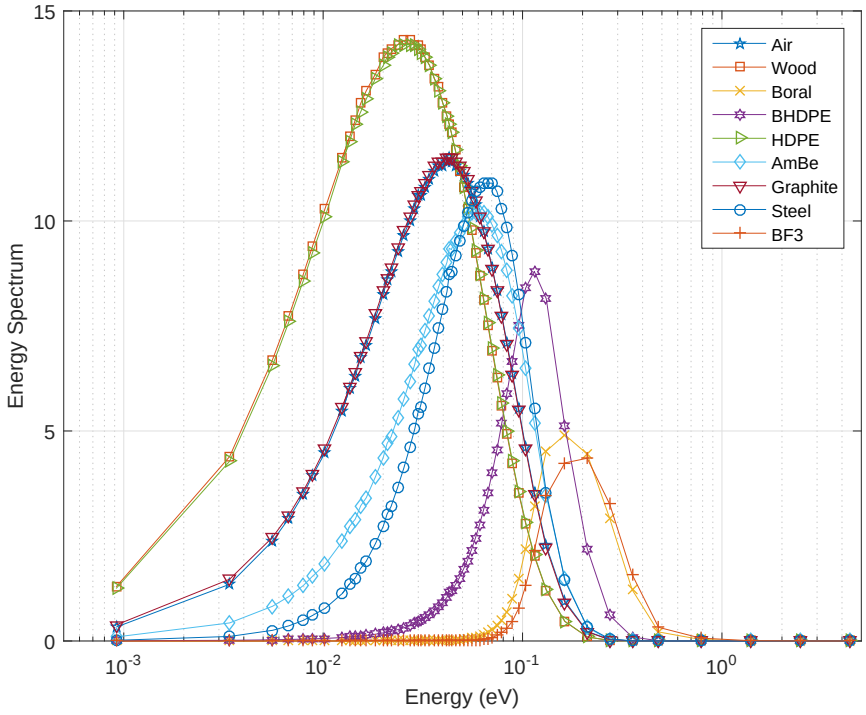
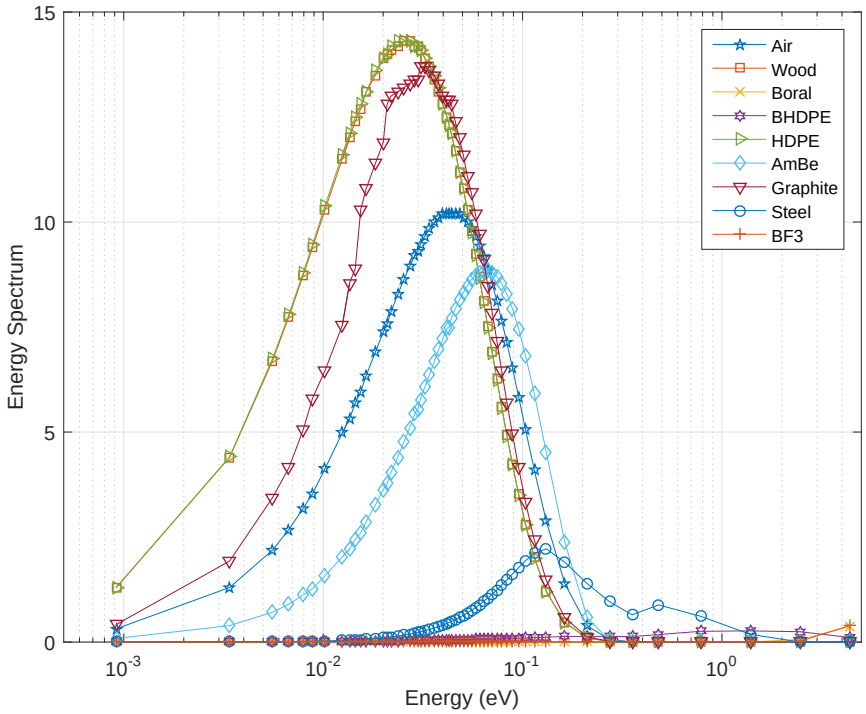


Figure 1.26: Configuration of the IM1 problem with the outer layers of air removed.



(a) Two-Grid



(b) Modified Two-Grid

Figure 1.27: Spectral shape of the infinite medium iteration matrices for the IM1 problem materials.

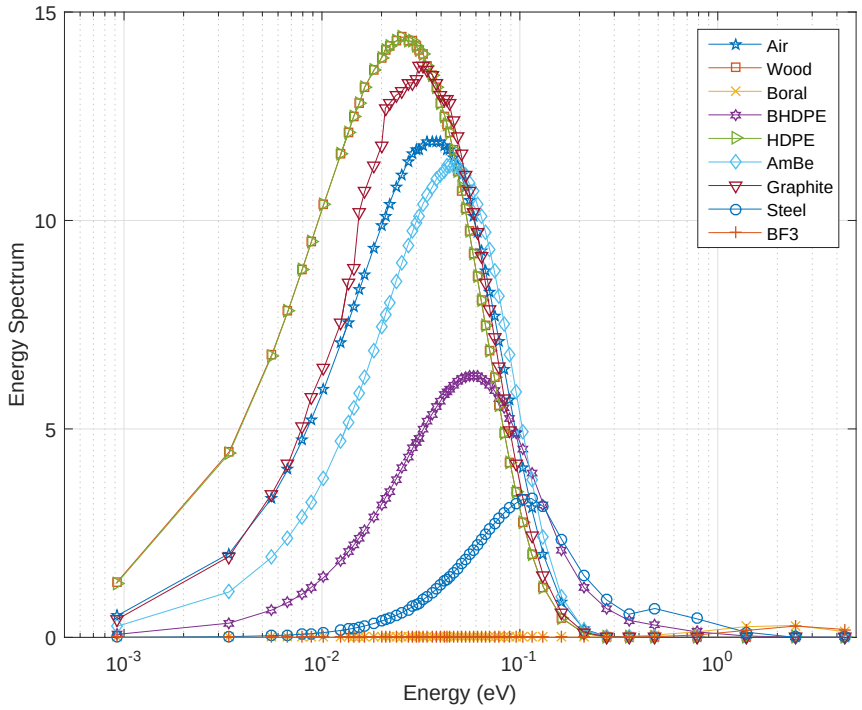
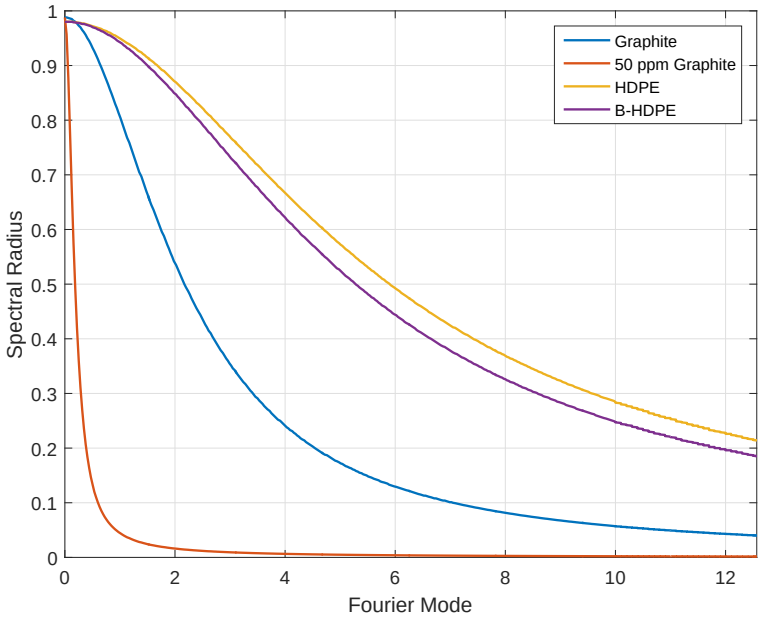
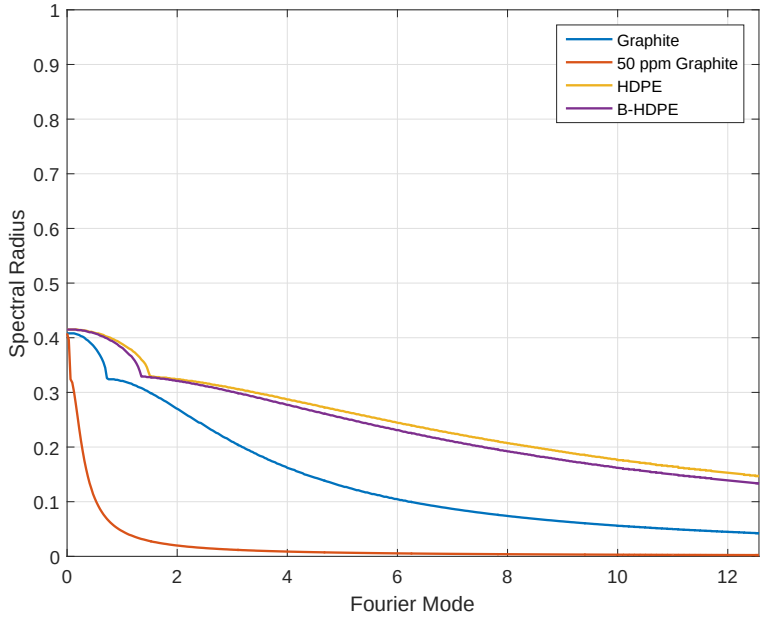


Figure 1.28: Spectral shape of the infinite medium iteration matrices for the Multigroup Jacobi Acceleration Method for the IM1 problem materials.



(a) Gauss-Seidel



(b)

Figure 1.29: 1D fourier analysis for some IM1 materials of interest. (a)

REFERENCES

[1] M. ADAMS and E. LARSEN, "Fast iterative methods for discrete-ordinates particle transport calculations," *Progress in nuclear energy*, **40**, *1*, 3–159 (2002).

[2] H. KOPP, "Synthetic method solution of the transport equation," *Nuclear Science and Engineering*, **17**, 65 (1963).

[3] V. LEBEDEV, "The Iterative *KP* Method for the Kinetic Equation," in "Proc. Conf. on Mathematical Methods for Solution of Nuclear Physics Problems," (1964).

[4] V. LEBEDEV, "The KP-method of accelerating the convergence of iterations in the solution of the kinetic equation," *Numerical methods of solving problems of mathematical physics, Nauka, Moscow*, pp. 154–176 (1966).

[5] V. LEBEDEV, "On Finding Solutions of Kinetic Problems," *USSR Comp. Math. and Math. Phys.*, **6**, 895 (1966).

[6] V. LEBEDEV, "An Iterative *KP* Method," *USSR Comp. Math. and Math. Phys.*, **7**, 1250 (1967).

[7] V. LEBEDEV, "Problem of the Convergence of a Method of Estimating Iteration Deviations," *USSR Comp. Math. and Math. Phys.*, **8**, 1377 (1968).

[8] V. LEBEDEV, "Convergence of the *KP* Method for Some Neutron Transfer Problems," *USSR Comp. Math. and Math. Phys.*, **9**, 226 (1969).

[9] V. LEBEDEV, "Construction of the *P* Operation in the *KP* Method," *USSR Comp. Math. and Math. Phys.*, **9**, 762 (1969).

[10] E. GELBARD and L. HAGEMAN, "The synthetic method as applied to the SN equations," *Nucl. Sci. Eng*, **37**, *2*, 288 (1969).

[11] W. H. REED, "The effectiveness of acceleration techniques for iterative methods in transport theory," *Nucl. Sci. Eng*, **45**, *3*, 245 (1971).

[12] R. ALCOUFFE, "Stable diffusion synthetic acceleration method for neutron transport iterations," Los Alamos Scientific Lab., NM (1976), vol. 23.

[13] R. ALCOUFFE, "The Diffusion Synthetic Acceleration Method Applied to Two-Dimenaionsl Neutron Transport Problems," Los Alamos Scientific Lab., NM (1977), vol. 27.

[14] R. E. ALCOUFFE, "Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations," *Nuclear Science and Engineering*, **64**, *2*, 344–355 (1977).

[15] E. W. LARSEN, "Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part I: Theory," *Nucl. Sci. Eng*, **82**, 47 (1982).

[16] D. R. MCCOY and E. W. LARSEN, "Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part II: Numerical results," *Nucl. Sci. Eng*, **82**, 64 (1982).

[17] M. L. ADAMS and W. R. MARTIN, "Diffusion synthetic acceleration of discontinuous finite element transport iterations," *Nucl. Sci. Eng*, **111**, 145–167 (1992).

[18] J. S. WARSA, T. A. WAREING, and J. E. MOREL, "Fully consistent diffusion synthetic acceleration of linear discontinuous S n transport discretizations on unstructured tetrahedral meshes," *Nuclear Science and Engineering*, **141**, *3*, 236–251 (2002).

[19] E. L. T.A. WAREING and M. ADAMS, "Diffusion Accelerated Discontinuous Finite Element Schemes for the S_N Equations in Slab and X-Y Geometries," in "Advances in Mathematics, Computations, and Reactor Physics," (1991), p. 245.

[20] Y. WANG and J. RAGUSA, "Diffusion Synthetic Acceleration for High-Order Discontinuous Finite Element S_n Transport Schemes and Application to Locally Refined Unstructured Meshes," *Nuclear Science and Engineering*, **166**, 145–166 (2010).

[21] Y. WANG, *Adaptive mesh refinement solution techniques for the multigroup SN transport equation using a higher-order discontinuous finite element method*, Ph.D. thesis, Texas A&M University (2009).

[22] B. TURCKSIN and J. C. RAGUSA, "Discontinuous diffusion synthetic acceleration for Sn transport on 2D arbitrary polygonal meshes," *Journal of Computational Physics*, **274**, 356–369 (2014).

[23] B. ADAMS and J. MOREL, "A two-grid acceleration scheme for the multigroup Sn equations with neutron upscattering," *Nuclear science and engineering*, **115**, *3*, 253–264 (1993).

[24] T. M. EVANS, K. T. CLARNO, and J. E. MOREL, "A transport acceleration scheme for multigroup discrete ordinates with upscattering," *Nuclear Science and Engineering*, **165**, *3*, 292–304 (2010).

[25] D. N. ARNOLD, F. BREZZI, B. COCKBURN, and L. D. MARINI, "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM journal on numerical analysis*, **39**, *5*, 1749–1779 (2002).

[26] J. C. RAGUSA, "Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids," *Journal of Computational Physics*, **280**, 195–213 (2015).

[27] M. HACKEMACK and J. RAGUSA, "A DFEM Formulation of the Diffusion Equation on Arbitrary Polyhedral Grids," in "Transactions of the American Nuclear Society," (2014).

[28] J. AKIN, *Application and implementation of finite element methods*, Academic Press, Inc. (1982).

[29] J. LIONS, "Problemes aux Limites non Homogenes a Donnees Irregulieres," *Numerical Analysis of Partial Differential Equations*, pp. 283–292 (2011).

[30] J. NITSCHE, "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind," in "Abhandlungen aus dem mathematischen Seminar der Universität Hamburg," Springer (1971), vol. 36, pp. 9–15.

[31] D. N. ARNOLD, "An interior penalty finite element method with discontinuous elements," *SIAM journal on numerical analysis*, **19**, *4*, 742–760 (1982).

[32] Y. WANG and J. C. RAGUSA, "A two-mesh adaptive mesh refinement technique for SN neutral-particle transport using a higher-order DGFEM," *Journal of computational and applied mathematics*, **233**, *12*, 3178–3188 (2010).

[33] Y. SAAD, *Iterative methods for sparse linear systems*, Siam (2003).

[34] S. C. EISENSTAT, "Efficient implementation of a class of preconditioned conjugate gradient methods," *SIAM Journal on Scientific and Statistical Computing*, **2**, *1*, 1–4 (1981).

[35] J. A. NELDER and R. MEAD, "A simplex method for function minimization," *The computer journal*, **7**, *4*, 308–313 (1965).

[36] J. CHANG and M. ADAMS, "Analysis of transport synthetic acceleration for highly heterogeneous problems," in "Proc. of M&C Topical Meeting," (2003).

[37] T. S. BAILEY, *The piecewise linear discontinuous finite element method applied to the RZ and XYZ transport equations*, Ph.D. thesis, Texas A&M University (2008).

[38] Y. AZMY, "Unconditionally stable and robust adjacent-cell diffusive preconditioning of weighted-difference particle transport methods is impossible," *Journal of Computational Physics*, **182**, *1*, 213–233 (2002).

[39] H. KHALIL, "A nodal diffusion technique for synthetic acceleration of nodal S_n calculations," *Nuclear Science and Engineering*, **90**, *3*, 263–280 (1985).

[40] M. A. L. R. N. A. WILLIAM HAWKINS, TIMMIE SMITH and M. ADAMS, "Efficient Massively Parallel Transport Sweeps," in "Transactions of the American Nuclear Society," (2012), vol. 107, pp. 477–481.

[41] M. P. ADAMS, M. L. ADAMS, W. D. HAWKINS, T. SMITH, L. RAUCHWERGER, N. M. AMATO, T. S. BAILEY, and R. D. FALGOUT, "Provably Optimal Parallel Transport Sweeps on Regular Grids," in "Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, Idaho," (2013).

[42] R. D. FALGOUT and U. M. YANG, "hypre: A Library of High Performance Preconditioners," in P. SLOOT, A. HOEKSTRA, C. TAN, and J. DONGARRA, editors, "Computational Science ICCS 2002," Springer Berlin Heidelberg, *Lecture Notes in Computer Science*, vol. 2331, pp. 632–641 (2002).

[43] V. HENSON and U. YANG, "BoomerAMG: a parallel algebraic multigrid solver and preconditioner," *Applied Numerical Mathematics*, **41**, *5*, 155–177 (2002).

[44] R. J. ZERR, *Solution of the within-group multidimensional discrete ordinates transport equations on massively parallel architectures*, Ph.D. thesis, The Pennsylvania State University (2011).