ACCELERATION TECHNIQUES FOR DISCRETE-ORDINATES TRANSPORT METHODS WITH HIGHLY FORWARD-PEAKED SCATTERING

A Dissertation

by

BRUNO ROGER FERNAND TURCKSIN

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

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Jean C. Ragusa
	Jim E. Morel
Committee Members,	Wolfgang Bangerth
	Marvin L. Adams
Department Head,	Yassin A. Hassan

December 2012

Major Subject: Nuclear Engineering

ABSTRACT

In this dissertation, advanced numerical methods for highly forward peaked scattering deterministic calculations are devised, implemented, and assessed. Since electrons interact with the surrounding environment through Coulomb interactions, the scattering kernel is highly forward-peaked. This bears the consequence that, with standard preconditioning, the standard Legendre expansion of the scattering kernel requires too many terms for the discretized equation to be solved efficiently using a deterministic method. The Diffusion Synthetic Acceleration (DSA), usually used to speed up the calculation when the scattering is weakly anisotropic, is inefficient for electron transport. This led Morel and Manteuffel to develop an one-dimensional angular multigrid (ANMG) which has proved to be very effective when the scattering is highly anisotropic. Later, Pautz et al. generalized this scheme to multidimensional geometries, but this method had to be stabilized by a diffusive filter that degrades the overall convergence of the iterative scheme. In this dissertation, we recast the multidimensional angular multigrid method without the filter as a preconditioner for a Krylov solver. This new method is stable independently of the anisotropy of the scattering and is increasingly more effective and efficient as the anisotropy increases compared to DSA preconditioning wrapped inside a Krylov solver. At the coarsest level of ANMG, a DSA step is needed. In this research, we use the Modified Interior Penalty (MIP) DSA. This DSA was shown to be always stable on triangular cells with isotropic scattering. Because this DSA discretization leads to symmetric definite-positive matrices, it is usually solved using a conjugate gradient preconditioned (CG) by SSOR but here, we show that algebraic multigrid methods are vastly superior than more common CG preconditioners such as SSOR.

Another important part of this dissertation is dedicated to transport equation and diffusion solves on arbitrary polygonal meshes. The advantages of polygonal cells are that the number of unknowns needed to mesh a domain can be decreased and that adaptive mesh refinement implementation is simplified: rather than handling hanging nodes, the adapted computational mesh includes different types of polygons. Numerical examples are presented for arbitrary quadrilateral and polygonal grids.

DEDICATION

To my dad, Dominique Turcksin

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Jean Ragusa and Dr. Jim Morel. They helped me in my research since the very beginning and are a real source of inspiration for me. I also would like to thank Dr. Wolfgang Bangerth for all the useful discussions we had over the years.

I am also grateful to all the people who made my life in College Station better: Yaqi Wang, Vijay Mahadevan, Tara Pandya, Michael Adams, Fabien Auffret, Marco Delchini, Damien Lebrun-Grandié, and the most important of all my girlfriend Alison Pawlicki. Without these people, I would not have been able to accomplish this.

Lastly, I would like to thank my family. They have always supported and believed in me. In particular, I would like to thank my Dad who has always done everything he could to smooth any bump I encountered.

NOMENCLATURE

AGMG	AGgregation-based algebraic MultiGrid
AMG	Algebraic MultiGrid method
AMR	Adaptive Mesh Refinement method
ANMG	ANgular MultiGrid method
ANMG-DSA	ANMG using DSA at the coarsest level
ANMG-$P1$	ANMG using $P1$SA at the coarsest level
B	Boltzmann
BFP	Boltzmann-Fokker-Planck
BLD	BiLinear Discontinuous finite elements
c	Scattering ratio
CG	Conjugate Gradient method
$\boldsymbol{D}$	Direction-to-moment matrix
DSA	Diffusion Synthetic Acceleration
FP	Fokker-Planck
GLC	Gauss-Legendre-Chebyshev
GMRES	Generalized Minimal RESidual method
$\boldsymbol{L}$	Streaming matrix
$\boldsymbol{M}$	Moment-to-direction matrix

MIP	Modified Interior Penalty
MIS	Maximally Independent Sets
ML	MultiLevel package of Trilinos
MM	Morel and Manteuffel angular multigrid method
$P1$SA	$P1$ Synthetic Acceleration
P_l	Legendre polynomial of degree l
P_l^m	Associated Legendre polynomial of degree l and order m
PAM	Pautz, Adams, and Morel angular multigrid method
PAMNF	PAM with No Filtering
PWLD	PieceWise Linear Discontinuous finite elements
$\tilde{R}$	Mean square stopping power
S	Restricted stopping power
$\tilde{S}$	Stopping power
SI	Source Iteration
S_n	Discrete ordinates method of order n
SGS	Symmetric Gauss-Seidel
SPD	Symmetric Positive-Definite
SSOR	Symmetric Successive OverRelaxation method
T	Half of the restricted momentum transfer
$\tilde{T}$	Half of the momentum transfer
Y_l^m	Spherical harmonic of degree l and order m

$\mathbf{\Omega}$	(μ, φ) Unit vector in the flight direction
μ	Cosine of the directional azimuthal angle
μ_0	Scalar product of $\mathbf{\Omega}$ and $\mathbf{\Omega}'$
ψ	Angular flux
$\mathbf{\Sigma}$	Scattering cross sections matrix
$\Sigma_a(\mathbf{r}, E)$	Absorption macroscopic cross section
$\Sigma_s(\mathbf{r}, E)$	Scattering macroscopic cross section
$\Sigma_s(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}, E' \to E)$	Differential scattering macroscopic cross section
$\Sigma_t(\mathbf{r}, E)$	Total macroscopic cross section
θ	Directional polar angle
φ	Directional azimuthal angle

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

CHAPTER I

INTRODUCTION

A Purpose

The transport of photons and electrons has many applications: satellite electronics shielding, flash x-ray machine design, radiotherapy, and a wide variety of other applications. Radiotherapy uses photons and charged particles to damage the DNA of cancerous cells. When using photons, free electrons are generated and ionize the environment to create free radicals that damage the cells. The absorbed dose, defined as the energy deposited per unit of mass, is used to gauge whether a cell will die from the amount of radiation received or not. Several methods can be applied to compute the dose distribution in the body: semi-analytic, deterministic, and Monte-Carlo methods. Monte-Carlo methods yield very accurate results, however they are slow to converge and remain too slow for effective clinical use [100, 118]. Semi-analytic methods, such as pencil-beam convolution [22, 55] and convolution-superposition [65], employ pre-calculated Monte-Carlo dose kernels, which are then locally scaled to approximate photon and electron transport in the presence of heterogeneities. These methods have issues in the presence of large density gradients such as those found at interfaces between different materials: air, bone, lung and soft tissue [57, 92, 100]. Deterministic methods like the discrete ordinates (S_n) method has been shown to be quite accurate for electron and coupled electron-photon transport [64, 69, 113]. However, devising efficient solution algorithms for S_n multidimensional photon-electron transport is an ongoing active field of research.

A particular difficulty of using the discrete ordinates method arises from the transport of electrons. Charged particles undergo interactions with the background medium and these interactions result in extremely small changes in particle direction and energy. These interactions are well characterized by the Fokker-Planck limit of the Boltzmann equation [37, 72]. In this limit, the directional and energy changes are decoupled; the former is modeled by the continuous scattering operator and the latter is modeled by the continuous-slowing-down operator. The mean free path and the directional change per scattering interaction approach zero while the momentum transfer (also called the transport-corrected scattering cross section) remains fixed.

When the scattering is highly forward-peaked, solving the S_n transport equation can be challenging due to the slow convergence of standard iterative algorithms, such as Source Iteration (SI). To speed up iterative convergence, acceleration schemes such as Diffusion Synthetic Acceleration (DSA) and P1 Synthetic Acceleration (P1SA) are generally used for neutron transport [19]. These methods use a diffusion equation or the P1 equations, and therefore, only the zeroth or the zeroth plus the first flux moment can be accelerated. When the zeroth flux moment alone is accelerated, these schemes are stable [20] (in this discussion, we ignore the possible issues due to the spatial discretization) but they are very inefficient for highly anisotropic scattering. If both the zeroth and the first flux moments are accelerated, the spectral radius of the continuous scheme (i.e., without spatial discretization) with anisotropic scattering is given in multidimensional calculations by [20]:

$$\rho_{ani} = \max\left(\rho_{iso}, \frac{\bar{\mu}_0 c}{1 - \bar{\mu}_0 c}\right) \tag{1.1}$$

where $\rho_{iso}(< 1)$ is the spectral radius when the scattering is isotropic, $\bar{\mu}_0$ $(\in [0, 1])$ is the average scattering cosine, and c $(\in [0, 1])$ is the scattering ratio. We see that when $\mu_0 c > 0.5$, this SI+DSA scheme is unstable. Several modifications have been proposed [20, 101] to stabilize this acceleration scheme: using DSA after several SI iterations or accelerating the zeroth moment at every even SI+DSA iteration and the zeroth and the first moments at every odd SI+DSA iteration. In one-dimensional geometry, using DSA to accelerate both the zeroth and the first flux moment leads to a scheme which is always stable. However, for electron transport and more generally for highly anisotropic kernels, more computationally efficient techniques are required.

The angular multigrid method developed in [71] has proven to be very effective to solve the S_n equations with highly forward-peaked scattering for one-dimensional slab geometry. Unfortunately, the extension of this method to multidimensional geometries, like P1SA, is unstable [84]. Pautz et al. added a diffusive filter to the angular multigrid corrections to stabilize the method which, then, converges faster than DSA alone. However, the spectral radius can become arbitrary close to one for a highly anisotropic and high scattering ratio medium. Even though SI has, for a long time, been the traditional solution technique for S_n, and in [71, 84], it is the technique employed, SI is not the only iterative approach to solve the S_n equations. It can also be tackled using non stationary Krylov solvers such as Generalized Minimal Residual method (GMRES). A code solving the S_n equations using SI (preconditioned with DSA) can easily be modified to use a preconditioned Krylov solver. In [44], the authors summarize the advantageous features of GMRES as follows: “using DSA as preconditioner for GMRES(m) removes the consistency requirement that plagues DSA-accelerated source iteration in multidimensional problems.” Driven by this statement, we will use the multidimensional angular multigrid method as a preconditioner for GMRES in solving highly forward-peaked scattering problems. Our hope is that GMRES will be able to stabilize the proposed scheme without the use of a filter and that the new scheme will have convergence properties similar to those of the one-dimensional scheme. At the coarsest level of the angular multigrid technique, a DSA scheme or a P1SA scheme has to be used. Here, we will use an adaptation of the Modified Interior Penalty DSA (MIP) [105]. This DSA was developed for discontinuous finite elements on triangular cells and it is symmetric and positive-definite (SPD). We will adapt MIP to Bilinear Discontinuous Finite elements (BLD) on rectangular cells and to PieceWise Linear Discontinuous Finite elements (PWLD) [25, 95] on arbitrary polygonal cells. Using MIP requires us to solve a SPD system of linear equations. This has usually been done using conjugate gradient preconditioned by Symmetric Successive OverRelaxation (SSOR), but in this research we will test the effectiveness of algebraic multigrid methods (AMG) to precondition the Krylov solver [38, 103]. Algebraic multigrid methods allow the use of multigrid techniques when there is no grid or when the mesh is unstructured. Instead of using a succession of grids based on the geometry of the problems, the grids are based on properties of the matrix which allows the use of AMG as a black-box solver or preconditioner. MIP will be adapted to polygonal cells due to the advantages of polygonal cells. Polygonal cells can potentially reduce the number of unknowns, while maintaining symmetry within the mesh. We show this potential reduction for a hexagonal cell versus the same space divided using triangles:

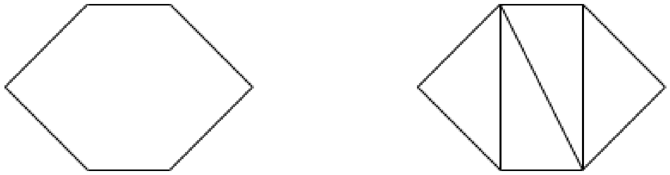


Figure I.1: Hexagonal cell versus triangle cells

We see in Figure I.1 that if there is one unknown per vertex, the hexagonal cell will have 6 unknowns compared to the 12 unknowns of triangle cells. Polygonal cells can also be used for adaptive mesh refinement (AMR) without having to deal with hanging nodes [23, 94, 117]. The left cell on the Figure I.2 is a degenerate pentagon whereas the two cells on the right are quadrilaterals:

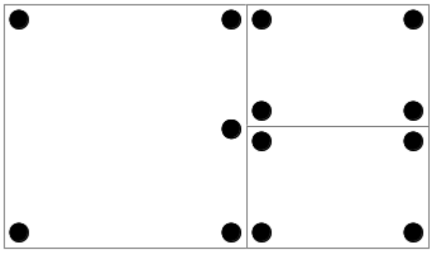


Figure I.2: AMR mesh

B Linear Boltzmann equation

Charged particles transport can be described by the linear Boltzmann equation [3, 68, 69]:

$$\begin{aligned} \mathbf{\Omega}\cdot\boldsymbol{\nabla}\psi(\mathbf{r},\mathbf{\Omega},E)+\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) = &\int_0^\infty dE'\int_{4\pi}d\mathbf{\Omega}' \\ &\Sigma_s(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\to E)\psi(\mathbf{r},\mathbf{\Omega}',E')+Q(\mathbf{r},\mathbf{\Omega},E) \end{aligned} \tag{1.2}$$

where:

- $\mathbf{\Omega}=(\mu,\varphi)$ is a unit vector in the flight direction
- $\mu=\cos(\theta)$, where θ is the directional polar angle
- φ is the directional azimuthal angle
- $\mu_0=\mathbf{\Omega}'\cdot\mathbf{\Omega}$
- $\psi(\mathbf{r},\mathbf{\Omega},E)=vf(\mathbf{r},\mathbf{\Omega},E)$ is the angular flux
- $\Sigma_t(\mathbf{r},E)$ is the total macroscopic cross section given by:

$$\Sigma_t(\mathbf{r},E)=\Sigma_a(\mathbf{r},E)+\Sigma_s(\mathbf{r},E) \tag{1.3}$$

- $\Sigma_a(\mathbf{r}, E)$ is the absorption macroscopic cross section
- $\Sigma_s(\mathbf{r}, E)$ is the scattering macroscopic cross section
- $\Sigma_s(\mathbf{r}, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega}, E' \rightarrow E)$ is the differential scattering macroscopic cross scattering
- $Q(\mathbf{r}, \boldsymbol{\Omega}, E)$ is the volumetric source

In the remainder of this work, macroscopic cross sections will simply be called cross sections when no confusion is possible. Standard boundary conditions can be applied to equation (1.2). The most common is the incoming flux boundary condition:

$$\psi(\mathbf{r}, \boldsymbol{\Omega}, E) = g(\mathbf{r}, \boldsymbol{\Omega}, E) \text{ for } \boldsymbol{\Omega} \cdot \boldsymbol{n} < 0 \text{ and } \mathbf{r} \in \partial\mathcal{D}, \tag{1.4}$$

where $\partial\mathcal{D}$ is the boundary of the domain $\mathcal{D}$. If $g = 0$, equation (1.4) yields the vacuum boundary conditions.

Equation (1.2) depends on space ($\mathbf{r}$), angle ($\boldsymbol{\Omega}$), and energy (E). In practice, the energy variable is treated through a multigroup formalism, with outer iterations between all energy groups to include down/up scattering events (including transfers between particle types). The multigroup equations are solved one group at a time. This within-group problem, which is equivalent to a monoenergetic problem, retains the challenging features of the electron-photon problem we want to address. As such, the methods developed here will be presented for the monoenergetic transport equation. However, the techniques described here apply straightforwardly to the multigroup equations. The monoenergetic equation (1.2) is given by:

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla}\psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_t(\mathbf{r})\psi(\mathbf{r}, \boldsymbol{\Omega}) = \int_{4\pi} d\boldsymbol{\Omega}'\ \Sigma_s(\mathbf{r}, \boldsymbol{\Omega}' \cdot \boldsymbol{\Omega})\psi(\mathbf{r}, \boldsymbol{\Omega}') + Q(\mathbf{r}, \boldsymbol{\Omega}). \tag{1.5}$$

Using:

$$\Sigma_s(\mathbf{r}, \mathbf{\Omega} \cdot \mathbf{\Omega}') = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{s,l}(\mathbf{r}) P_l(\mathbf{\Omega} \cdot \mathbf{\Omega}') \tag{1.6}$$

$$\Sigma_{s,l}(\mathbf{r}) = 2\pi \int_{-1}^{1} d\mu_0 \; P_l(\mu_0) \Sigma_s(\mathbf{r}, \mu_0), \tag{1.7}$$

the scattering term can be represented by a Legendre polynomials P_l expansion:

$$\int_{4\pi} \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) \psi(\mathbf{r}, \mathbf{\Omega}') d\mathbf{\Omega}' = \int_{4\pi} \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \Sigma_{s,l} P_l(\mathbf{\Omega}' \cdot \mathbf{\Omega}) \psi(\mathbf{r}, \mathbf{\Omega}') d\mathbf{\Omega}'. \tag{1.8}$$

Using the addition theorem for spherical harmonics, Y_l^m the spherical harmonics of degree l and order m, and P_l^m the associated Legendre polynomials:

$$P_l(\mathbf{\Omega} \cdot \mathbf{\Omega}') = \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega}) Y_l^{m,*}(\mathbf{\Omega}') \tag{1.9}$$

$$Y_l^m(\mathbf{\Omega}) = (-1)^m \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\mu) e^{im\varphi}, \tag{1.10}$$

equation (1.8) becomes:

$$\begin{aligned} \int_{4\pi} \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}) \psi(\mathbf{r}, \mathbf{\Omega}') d\mathbf{\Omega}' = & \int_{4\pi} \left(\sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \frac{4\pi}{2l+1} \Sigma_{s,l}(\mathbf{r}) \right. \\ & \left. \sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega}) Y_l^{m,*}(\mathbf{\Omega}') \psi(\mathbf{r}, \mathbf{\Omega}') \right) d\mathbf{\Omega}' \qquad (1.11) \\ = & \sum_{l=0}^{\infty} \Sigma_{s,l}(\mathbf{r}) \sum_{m=l}^{l} \phi_{l,m}(\mathbf{r}) Y_l^m(\mathbf{\Omega}), \end{aligned}$$

where we have introduced:

$$\psi(\mathbf{r},\mathbf{\Omega}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \phi_{l,m}(\mathbf{r}) Y_l^m(\mathbf{\Omega}), \tag{1.12}$$

$$\phi_{l,m}(\mathbf{r}) = \int_{4\pi} d\mathbf{\Omega}\ Y_l^{m,*}(\mathbf{\Omega})\psi(\mathbf{r},\mathbf{\Omega}). \tag{1.13}$$

In practice, the scattering expansion is truncated ($\sum_{l=0}^{\infty} \rightarrow \sum_{l=0}^{L}$).

For the derivation of the Boltzmann-Fokker-Planck equation, we will need the following property of the spherical harmonics:

$$\left[\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \left(\frac{1}{1-\mu^2}\right)\frac{\partial^2}{\partial\varphi} + l(l+1)\right] Y_l^m(\mathbf{\Omega}) = 0. \tag{1.14}$$

Equation (1.5) still needs to be discretized in space and angle. A standard method to discretize the space variable is to use discontinuous Galerkin finite elements [18, 107, 112]. The angular discretization that we will use in this work is the S_n or discrete ordinate method [12, 31, 33, 40]. With this discretization, equation (1.5) is replaced by a system of linear equations which use discrete angular fluxes ($\psi(\mathbf{r},\mathbf{\Omega}) \rightarrow \psi(\mathbf{r},\mathbf{\Omega}_d) = \psi_d(\mathbf{r})$) and the integral in equation (1.13) is replaced by a quadrature:

$$\phi_{l,m}(\mathbf{r}) = \sum_d w_d Y_l^{m,*}(\mathbf{\Omega}_d)\psi_d(\mathbf{r}), \tag{1.15}$$

where w_d are the weights associated to the quadrature. Therefore, the S_n discretization of equation (1.5) is given by:

$$\mathbf{\Omega}_d \cdot \boldsymbol{\nabla}\psi_d(\mathbf{r}) + \Sigma_t(\mathbf{r})\psi_d(\mathbf{r}) = \sum_{l=0}^{L} \Sigma_{s,l}(\mathbf{r}) \sum_{m=-l}^{l} \phi_{l,m} Y_l^m(\mathbf{\Omega}_d) + Q_d(\mathbf{r}). \tag{1.16}$$

Equation (1.16) can be written in a more compact way using operators:

$$\boldsymbol{L}\Psi = \boldsymbol{M}\boldsymbol{\Sigma}\boldsymbol{D}\Psi + Q, \tag{1.17}$$

where:

- $\boldsymbol{L}$ is the streaming operator $\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla} + \Sigma_t(\mathbf{r})$
- $\boldsymbol{M}$ is the moment-to-direction operator $\Psi = \boldsymbol{M}\Phi$
- $\boldsymbol{D}$ is the direction-to-moment operator $\Phi = \boldsymbol{D}\Psi$
- $\boldsymbol{\Sigma}$ is the scattering cross-section operator
- Ψ is the vector of angular fluxes ψ_d
- Φ is the vector of angular flux moments $\phi_{l,m}$.

C Organization of the Dissertation

In Chapter II, we introduce the Boltzmann-Fokker-Planck (BFP) equation used to describe the transport of charged particles. To obtain the BFP equation, the Fokker-Planck operator is added in the Boltzmann equation in order to simplify the treatment of the highly forward-peaked scattering kernel. We show that the Fokker-Planck equation is an asymptotic limit of the Boltzmann equation when the mean free path goes to zero and $\bar{\mu}_0$ goes to one. The Fokker-Planck equation is not valid for any forward-peaked scattering kernel and therefore, the BFP approximation has some limitations. In particular, the Henyey-Greenstein kernel and the Rutherford cross section do not satisfy the Fokker-Planck limit. With these limitations in mind, we introduce the Fokker-Planck cross sections that reduce the Fokker-Planck equation and the BFP equation to a Boltzmann equation. Fokker-Planck cross sections cannot be used with any angular S_n quadrature but specific quadratures, known as

Galerkin quadratures, must be adopted. The importance and the properties of these quadratures are explained in details at the end of Chapter II.

In Chapter III, we review in details the iterative solvers and the spatial discretizations used to solve the S_n transport equations. We explain how Source Iteration, Krylov solvers, and Diffusion Synthetic Acceleration can be used to solve the transport equation. We introduce two spatial discretizations for 2D geometries: BiLinear Discontinuous finite elements (BLD) and PieceWise Linear Discontinuous finite elements (PWLD). The BLD finite elements are used on rectangular cells while the PWLD finite elements can be used on any polygonal cells. The purpose of this Chapter is to facilitate expansion in the next chapters.

In Chapter IV, we introduce the angular multigrid methods for transport with highly forward-peaked scattering. We recall prior work on this topic and discuss the issues previously encountered for multidimensional geometries. The original angular multigrid method for one-dimensional geometry showed rapid convergence of source iterations for problems with highly forward-peaked scattering, whereas the standard SI+DSA approach is ineffective. Unfortunately, the generalization to multidimensional geometries required a filter to stabilized the method which resulted in a significantly less efficient scheme than in one-dimensional geometry. When the scattering becomes very anisotropic, this generalized method becomes ineffective. In this Chapter, we show that if the angular multigrid method is recast as a preconditioner for a Krylov solver, the method does not need diffusion filtering for stability and is always effective and efficient.

In Chapter V, we adapt the Modified Interior Penalty (MIP) DSA method developed for triangular cells to quadrilateral and polygonal cells. This DSA discretization is used as the coarsest level of the angular multigrid method developed in Chapter IV. Since MIP is symmetric and positive-definite, the most common method to solve it, is conjugate gradient (CG) preconditioned by SSOR. In Chapter V, we investigate algebraic multigrid methods as CG preconditioners to solve MIP. We show that algebraic multigrid preconditioners are vastly superior to more common CG preconditioners if the matrix associated with MIP is stored.

In Chapter VI, the implementation of the code developed for this research is detailed.

In Chapter VII, we finish with some concluding remarks and suggestions for future developments.

CHAPTER II

CHARGED PARTICLE TRANSPORT

A Introduction

For charged particle transport, the scattering kernel is very pronounced for both an almost zero energy transfer and an almost zero direction change. However, because the scattering mean free path (average distance between two collisions) is very small, particles can undergo a large number of collisions. Therefore, significant changes of energy and direction are common in most applications.

In theory, the linear Boltzmann equation could be used for charged particle transport but the scattering cross sections are so forward peaked that a Legendre expansion of the cross section would require too many terms. Moreover, using a deterministic method to solve the Boltzmann equation often requires cells of the same scale as the very small mean free path [87]. To avoid these difficulties, a Fokker-Planck operator can be used to represent the highly forward peaked scattering. Since this operator cannot represent the large angle scattering collisions, we employ a Legendre expansion for this purpose. The addition of the Fokker-Planck operator to the Boltzmann equation yields the Boltzmann-Fokker-Planck equation [89].

B Boltzmann-Fokker-Planck equation

In this section, we start by deriving the Boltzmann-Fokker-Planck equation. Then, we show that the Fokker-Planck equation is an asymptotic limit of the Boltzmann equation when both the energy transfer and the direction changes during a collision go to zero. We conclude this section with an analysis of the validity of the Fokker-Planck and Boltzmann-Fokker-Planck equations.

1 Derivation of the Boltzmann-Fokker-Planck equation

Following [89], the BFP equation is derived starting from the Boltzmann equation:

$$\begin{aligned}&\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\psi(\mathbf{r},\boldsymbol{\Omega},E)+\left(\Sigma_s(\mathbf{r},E)+\Sigma_a(\mathbf{r},E)\right)\psi(\mathbf{r},\boldsymbol{\Omega},E)=\\&\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\int_0^{\infty}\Sigma_{s,l}\left(\mathbf{r},E'\to E\right)\phi_{l,m}(\mathbf{r},E')Y_l^m(\boldsymbol{\Omega})\;dE'+Q(\mathbf{r},\boldsymbol{\Omega},E).\end{aligned}\tag{2.1}$$

When the microscopic scattering cross section ($\sigma_s(\mathbf{r},\mu_c,E')$) is known ($\mu_c$ is the scattering angle cosine in the center of mass system, whereas we will use μ_L for the scattering angle cosine in the laboratory system), the Legendre expansion of the macroscopic scattering cross section is given by:

$$\Sigma_{s,l}(E'\to E)=N\frac{4\pi}{\beta E'}\sigma_s\left(\mu_c\left(\frac{E'-E}{E'}\right),E'\right)P_l\left(\mu_L\left(\frac{E'}{E}\right)\right),\tag{2.2}$$

where $E\le E'\le\frac{E}{\alpha}$, $\beta=1-\alpha$, $\alpha=\left(\frac{A-1}{A+1}\right)^2$, A is the particle mass ratio, and N is the atom density.

First, the differential scattering cross section is split into two parts:

$$\begin{aligned}\Sigma_s\left(\mu_c\left(\frac{E'-E}{E}\right),E'\right)=&\Sigma_{s,reg}\left(\mu_c\left(\frac{E'-E}{E}\right),E'\right)\\&+\Sigma_{s,sing}\left(\mu_c\left(\frac{E'-E}{E}\right),E'\right),\end{aligned}\tag{2.3}$$

where:

- $\Sigma_{s,reg}$ is the “regular” cross section that does not vary rapidly as μ_c goes to one. By definition, a Legendre polynomial expansion of this cross section converges quickly (i.e., with a few terms).
- $\Sigma_{s,sing}$ is the “singular” cross section which is highly forward peaked and is not

negligible only when $\mu_c \approx 1$.

Then, the scattering term in equation (2.1)

$$q\left(\mathbf{\Omega}, E\right) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \int_0^{\infty} \Sigma_{s,l}(E' \to E)\phi_{l,m}(E')Y_l^m(\mathbf{\Omega})dE' \tag{2.4}$$

is split

$$q = q_{reg} + q_{sing}, \tag{2.5}$$

with

$$\begin{bmatrix} q_{reg}(\mathbf{\Omega}, E) \\ q_{sing}(\mathbf{\Omega}, E) \end{bmatrix} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega}) \begin{bmatrix} q_{l,m,reg}(E) \\ q_{l,m,sing}(E) \end{bmatrix} \tag{2.6}$$

and

$$\begin{bmatrix} q_{l,m,reg}(E) \\ q_{l,m,sing}(E) \end{bmatrix} = \int_E^{E/\alpha} dE' \phi_{l,m}(E') \begin{bmatrix} \Sigma_{s,l,reg}(E' \to E) \\ \Sigma_{s,l,sing}(E' \to E) \end{bmatrix}, \tag{2.7}$$

where $\Sigma_{s,l,reg}$ and $\Sigma_{s,l,sing}$ are the Legendre expansion coefficients corresponding to $\sigma_{s,reg}$ and $\sigma_{s,sing}$ through equation (2.2). Using μ_c instead of E' as integration variable in equation (2.7), we obtain:

$$q_{l,m,sing}(E) = 2\pi N \int_{-1}^{1} \frac{E'}{E} \sigma_{s,sing}(\mu_c, E') P_l\left(\mu_L(\mu_c)\right) \phi_{l,m}(E') d\mu_c \tag{2.8}$$

with

$$\mu_L = \frac{1 + A\mu_c}{(1 + A^2 + 2A\mu_c)^{1/2}}. \tag{2.9}$$

Assuming that $\phi_{l,m}(E')$ is a smooth function ($\sigma_{s,sing}(\mu_c, E')$ is almost singular in μ_c

but smooth in E'), we can perform the following Taylor expansions:

$$\begin{aligned} E'\sigma_{s,sing}(\mu_c, E')\phi_{l,m}(E') =& E\sigma_{s,sing}(\mu_c, E)\phi_{l,m}(E) \\ &+ (E' - E)\frac{\partial}{\partial E}\left(E\sigma_{s,sing}(\mu_c, E)\phi_{l,m}(E)\right) + \dots \end{aligned} \tag{2.10}$$

$$P_l(\mu_L) = P_l(1) - (1 - \mu_L)P_l'(1) + \dots \tag{2.11}$$

with:

$$P_l(1) = 1 \tag{2.12}$$

$$P_l'(1) = \frac{l(l+1)}{2}. \tag{2.13}$$

Using:

$$\mu_c = 1 - \frac{2}{\beta}\frac{E' - E}{E'} \tag{2.14}$$

$$\mu_L = \frac{1}{2}\left[(A+1)\sqrt{\frac{E}{E'}} - (A-1)\sqrt{\frac{E'}{E}}\right] \tag{2.15}$$

and assuming $\mu_c \approx 1$ or $A \ll 1$, we get:

$$\frac{E' - E}{E'} = \frac{2A}{(A+1)^2}(1 - \mu_c) \tag{2.16}$$

$$1 - \mu_L \approx \left(\frac{A}{A+1}\right)^2 (1 - \mu_c). \tag{2.17}$$

To first order in $(1 - \mu_c)$, we get:

$$q_{l,m,sing}(E) = \Sigma_{s,sing}(E)\phi_{l,m}(E) + \frac{\partial}{\partial E}S(E)\phi_{l,m}(E) - l(l+1)T(E)\phi_{l,m}(E) \tag{2.18}$$

where:

$$\Sigma_{s,sing}(E) = 2\pi N \int_{-1}^{1} \sigma_{s,sing}(E,\mu_c)\ d\mu_c \tag{2.19}$$

$$T(E) = N\pi \int_{-1}^{1} (1-\mu_L)\sigma_{s,sing}(E,\mu_c)\ d\mu_c \tag{2.20}$$

$$S(E) = \frac{4E}{A}T(E). \tag{2.21}$$

T is half of the restricted momentum transfer and S is the restricted stopping power. Using equations (1.12) and (1.14), we obtain:

$$q_{sing} = \Sigma_{s,sing}\psi + \frac{\partial}{\partial E}S\psi + T\left(\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \frac{1}{1-\mu^2}\frac{\partial^2}{\partial\varphi^2}\right)\psi. \tag{2.22}$$

Finally, we get the BFP equation:

$$\begin{aligned}\mathbf{\Omega}\cdot\mathbf{\nabla}\psi + (\Sigma_{s,reg}+\Sigma_a)\psi = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega})\int_E^{E/\alpha}\Sigma_{s,l,reg}(E'\to E)\times \\ \phi_{l,m}(E')dE' + \frac{\partial}{\partial E}S\psi + T\left[\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \frac{1}{(1-\mu^2)}\frac{\partial^2}{\partial\varphi^2}\right]\psi + Q.\end{aligned} \tag{2.23}$$

We see that only the regular portion of the scattering, $\Sigma_{s,reg}$, appears in the BFP equation, and that $\Sigma_{s,sing}$ is hidden in the restricted stopping power, S, and the restricted momentum transfer, T.

2 The Fokker-Planck equation as a limit of Boltzmann equation

In [87], Pomraning showed that the Fokker-Planck equation is an asymptotic limit of the Boltzmann equation when the mean free path goes to zero and $\bar{\mu}_0$ goes to one. Since his development will help us to understand the limitations of the Boltzmann-Fokker-Planck, we will briefly recall it here.

First, we assume that the unit of distance is chosen such that the characteristic size of the domain is $O(1)$ and that the scattering mean free path is small (i.e., $\Sigma_s \gg 1$). Next, we scale Σ_s as:

$$\Sigma_s(E) = \frac{\hat{\Sigma}_s(E)}{\Delta}, \tag{2.24}$$

where $\hat{\Sigma}_s = O(1)$ and $\Delta \ll 1$, Δ represents the scattering mean free path. We introduce the fast varying variables:

$$x = \frac{1-\mu_0}{\delta}, \ \delta \ll 1 \tag{2.25}$$

$$y = \frac{E'-E}{\epsilon}, \ \epsilon \ll 1. \tag{2.26}$$

δ and ϵ measure how peaked is the scattering kernel. δ represents the deviation of the cosine of a characteristic scattering angle from one. ϵ represents a characteristic value of the fraction of energy change during a single scattering collision. Next, the scattering kernel is scaled as:

$$\begin{aligned} \Sigma_s(\mu_0, E', E) &= \frac{1}{\Delta}\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E', \frac{E'-E}{\epsilon}\right) \\ &= \frac{1}{\Delta}\hat{\Sigma}_s(x, E', y), \end{aligned} \tag{2.27}$$

where $\hat{\Sigma}_s(x, E', y)$ is $O(1)$ and $\frac{\partial \hat{\Sigma}_s(x,E',y)}{\partial x}$, and $\frac{\partial \hat{\Sigma}_s(x,E',y)}{\partial y}$ are $O(1)$ when $(\epsilon, \delta) \to 0$. This scaling implies that the cross section is large and very peaked about $\mu_0 = 1$ and $E = E'$. The scaled transport equation is given by:

$$\begin{aligned} &\mathbf{\Omega} \cdot \boldsymbol{\nabla}\psi + \left(\Sigma_a + \frac{\hat{\Sigma}_s}{\Delta}\right)\psi = \frac{2\pi}{\Delta}\sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega}) \\ &\int_0^\infty dE' \ \phi_{l,m}(E') \int_{-1}^{1} d\mu_0 \ P_l(\mu_0)\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E', \frac{E'-E}{\epsilon}\right) + Q. \end{aligned} \tag{2.28}$$

We are interested in the asymptotic limit when the three parameters ϵ, δ, and Δ approach zero. Let us consider the following term:

$$K = \frac{2\pi}{\Delta} \int_0^\infty dE' \int_{-1}^1 d\mu_0 \; P_n(\mu_0) \hat{\Sigma}_s \left(\frac{1-\mu_0}{\delta}, E', \frac{E'-E}{\epsilon} \right) \phi_{l,m}(E'). \tag{2.29}$$

Now, we change the integration variables from (μ_0, E') to (x, y) according to equations (2.25) and (2.26):

$$K = \frac{2\pi\epsilon\delta}{\Delta} \int_{-E/\epsilon}^\infty dy \int_0^{2/\delta} dx \; P_l(1-\delta x) \hat{\Sigma}_s(x, E+\epsilon y, y) \phi_{l,m}(E+\epsilon y). \tag{2.30}$$

Next, we perform a Taylor expansion of the integrand about $\epsilon = \delta = 0$. We only keep the linear terms in δ and the quadratic terms in ϵ (we will see later that we only need to compute the linear terms in δ because we assume that the medium is isotropic):

$$\begin{aligned} K =& \frac{2\pi\epsilon\delta}{\Delta} \int_{-E/\epsilon}^\infty dy \int_0^{2/\delta} dx \; \left[P_l(1) - \delta x P_l'(1) + O(\delta^2) \right] \\ & \left[1 + \epsilon y \frac{\partial}{\partial E} + \frac{\epsilon^2 y^2}{2} \frac{\partial^2}{\partial E^2} + O(\epsilon^3) \right] \hat{\Sigma}_s(x, E, y) \phi_{l,m}(E). \end{aligned} \tag{2.31}$$

Now, $-E/\epsilon$ is replaced by $-\infty$ as the lower limit of the integral on y. We assume that the error is at most $O(\epsilon^3)$. This is correct if the scattering kernel falls off exponentially in energy from its maximum at $y = 0$ but if the kernel falls off too slowly, this substitution may increase the error above $O(\epsilon^3)$ and the current development becomes unsatisfactory. When the kernel falls off exponentially in energy, the replacement of the integration limit introduce an exponentially small error. We also neglect the cross terms in angle and energy in equation (2.31). It is not necessary to neglect them but the standard Fokker-Planck operator does not have these terms.

Using equations (2.12) and (2.13), equation (2.31) becomes:

$$\begin{aligned}
K =& \frac{2\pi\epsilon\delta}{\Delta}\int_{-\infty}^{\infty} dy \int_{0}^{2/\delta} dx \left(1 + O(\delta^2 + \epsilon\delta + \epsilon^3)\right) \hat{\Sigma}_s(x, E, y)\phi_{l,m}(E) \\
&- \frac{l(l+1)\pi\epsilon\delta^2}{\Delta}\int_{-\infty}^{\infty} dy \int_{0}^{2/\delta} dx\ x\hat{\Sigma}_s(x, E, y)\phi_{l,m}(E) \\
&+ \frac{2\pi\epsilon^2\delta}{\Delta}\frac{\partial}{\partial E}\int_{-\infty}^{\infty} dy \int_{0}^{2/\delta} dx\ y\hat{\Sigma}_s(x, E, y)\phi_{l,m}(E) \\
&+ \frac{\pi\epsilon^3\delta}{\Delta}\frac{\partial^2}{\partial E^2}\int_{-\infty}^{\infty} dy \int_{0}^{2/\delta} dx\ y^2\hat{\Sigma}_s(x, E, y)\phi_{l,m}(E).
\end{aligned} \tag{2.32}$$

We now go back to the (μ_0, E') variables using the following relationships:

$$x = \frac{1-\mu_0}{\delta} \tag{2.33}$$

$$y = \frac{E - E'}{\epsilon}. \tag{2.34}$$

Equations (2.25) and (2.33) are identical, but equations (2.26) and (2.34) are different (E and E' are interchanged). Therefore, we get:

$$\begin{aligned}
K =& \frac{2\pi}{\Delta}\int_{-\infty}^{\infty} dE' \int_{-1}^{1} d\mu_0 \left(\left(1 + O(\delta^2 + \epsilon\delta + \epsilon^3)\right) \hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E, \frac{E-E'}{\epsilon}\right)\phi_{l,m}(E)\right) \\
&- \frac{l(l+1)\pi}{\Delta}\int_{-\infty}^{\infty} dE' \int_{-1}^{1} d\mu_0 \left((1-\mu_0)\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E, \frac{E-E'}{\epsilon}\right)\phi_{l,m}(E)\right) \\
&+ \frac{2\pi}{\Delta}\frac{\partial}{\partial E}\int_{-\infty}^{\infty} dE' \int_{-1}^{1} d\mu_0 \left((E-E')\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E, \frac{E-E'}{\epsilon}\right)\phi_{l,m}(E)\right) \\
&+ \frac{\pi}{\Delta}\frac{\partial^2}{\partial E^2}\int_{-\infty}^{\infty} dE' \int_{-1}^{1} d\mu_0 \left((E-E')^2\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E, \frac{E-E'}{\epsilon}\right)\phi_{l,m}(E)\right).
\end{aligned} \tag{2.35}$$

Finally, using:

$$\frac{1}{\Delta}\hat{\Sigma}_s\left(\frac{1-\mu_0}{\delta}, E, \frac{E-E'}{\epsilon}\right) = \Sigma_s(\mu_0, E, E') \tag{2.36}$$

and replacing the lower limit of integration on the E' integral by 0, since the probability of scattering to a negative energy is zero, we obtain:

$$K = \Sigma_s \phi_{l,m} - l(l+1)\tilde{T}\phi_{l,m} + \frac{\partial}{\partial E}\tilde{S}\phi_{l,m} + \frac{1}{2}\frac{\partial^2}{\partial E^2}\tilde{R}\phi_{l,m} + O\left(\frac{\delta^2 + \epsilon\delta + \epsilon^3}{\Delta}\right) \tag{2.37}$$

where:

$$\tilde{T}(E) = \pi \int_0^\infty dE' \int_{-1}^1 d\mu_0 \; (1-\mu_0)\Sigma_s(\mu_0, E, E') = O\left(\frac{\delta}{\Delta}\right) \tag{2.38}$$

$$\tilde{S}(E) = 2\pi \int_0^\infty dE' \int_{-1}^1 d\mu_0 \; (E-E')\Sigma_s(\mu_0, E, E') = O\left(\frac{\epsilon}{\Delta}\right) \tag{2.39}$$

$$\tilde{R}(E) = 2\pi \int_0^\infty dE' \int_{-1}^1 d\mu_0 (E-E')^2\Sigma_s(\mu_0, E, E') = O\left(\frac{\epsilon^2}{\Delta}\right). \tag{2.40}$$

The function $\tilde{\alpha} = 2\tilde{T}$ is known as the momentum transfer while $\tilde{S}$ and $\tilde{R}$ are the stopping power and the mean square stopping power, respectively. Note that the difference with the restricted parameters defined earlier is that Σ_s is used instead of $\Sigma_{s,sing}$. Using equation (2.30), substituting equation (2.37) into the scaled transport equation (2.28), and using equation (2.24) for $\hat{\Sigma}_s(E)$, we get:

$$\begin{aligned} &\mathbf{\Omega} \cdot \mathbf{\nabla}\psi + (\Sigma_a + \Sigma_s)\,\psi = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y_l^m \Big(\Sigma_s \phi_{l,m} - l(l+1)\tilde{T}\phi_{l,m} + \\ &\frac{\partial}{\partial E}\tilde{S}\phi_{l,m} + \frac{1}{2}\frac{\partial^2}{\partial E^2}\tilde{R}\phi_{l,m}\Big) + Q + O\left(\frac{\delta^2 + \epsilon\delta + \epsilon^3}{\Delta}\right). \end{aligned} \tag{2.41}$$

Using equation (1.14), the $l(l+1)$ factor can be eliminated; then we can sum over the spherical harmonics according to equation (1.12) to obtain our final result:

$$\begin{gathered}\mathbf{\Omega}\cdot\mathbf{\nabla}\psi(\mathbf{r},\mathbf{\Omega},E)+\Sigma_a(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E)=\tilde{T}(\mathbf{r},E)\left(\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu}+\right.\\ \left.\left(\frac{1}{1-\mu^2}\right)\frac{\partial^2}{\partial\varphi^2}\right)\psi(\mathbf{r},\mathbf{\Omega},E)+\frac{\partial}{\partial E}\tilde{S}(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E)+\\ \frac{\partial^2}{\partial E^2}\tilde{R}(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E)+Q(\mathbf{r},\mathbf{\Omega},E)+O\left(\frac{\delta^2+\epsilon\delta+\epsilon^3}{\Delta}\right).\end{gathered}\tag{2.42}$$

We note that $\Sigma_s(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E)$ has canceled out in this equation.

Equation (2.42) is the Fokker-Planck equation for linear particle transport in an isotropic medium. As mentioned previously, the fact that the scattering kernel is peaked in energy ($E' \approx E$) and angle ($\mu_0 \approx 1$) is a necessary but not sufficient condition for equation (2.42) to be an asymptotic limit of equation (1.2). The additional sufficient condition is that the fall off is either exponential or strongly algebraic. Assuming that equation (2.42) is a valid asymptotic limit, it is clear from equations (2.38) to (2.40) that ϵ, δ and Δ must tend to zero in a correlated way. Looking at equation (2.38), we see that we need $O(\delta) = O(\Delta)$ to have a finite and nonzero angular term in equation (2.42). Similarly, by looking at equation (2.39), we must have $O(\epsilon) = O(\Delta)$ for equation (2.42) to have a finite and nonzero energy term. Therefore, when the scattering becomes more peaked ($\epsilon \to 0$ and $\delta \to 0$), the magnitude of the cross section must increase so that the momentum transfer stay finite and nonzero.

Since $\tilde{R}$ is $O\left(\frac{\epsilon^2}{\Delta}\right)$, we do not need $\tilde{R}$ to retain the leading order behavior in energy transfer. In many applications, the BFP equation used does not have the $\tilde{R}$ term. The reason for retaining $\tilde{R}$, even though it is a higher order term, is that it describes totally different physics than the stopping power ($\tilde{S}$) term: the stopping power term is convective whereas the $\tilde{R}$ term is diffusive. In certain applications, even a small diffusion of the particles in the energy variable can have an important effect and thus, it is important to keep the $\tilde{R}$ term [87]. For instance, $\tilde{R}$ is necessary to obtain thermal equilibrium. The reason that no convective term in angle appears in equation (2.42) (the $\tilde{T}$ term is diffusive) is that we have assumed an isotropic medium; that is there is the same probability for a particle to be scattered to the left or the right, and thus, the mean scattering angle is zero.

3 Limits of the Boltzmann-Fokker-Planck approximation

a Introduction

In this section, we recall the work of Larsen in [62]. We will show the limitation of the Fokker-Planck operator and, therefore, of the Boltzmann-Fokker-Planck equation on two well-known scattering kernels: the Henyey-Greenstein scattering kernel and the Rutherford scattering kernel [62]. The Henyey-Greenstein scattering kernel mimics the angular dependence of light scattering by small particles, whereas the screened Rutherford scattering kernel represents the scattering of electrons by a screened atomic nucleus.

First, we need to define several operators [62]:

$$\mathcal{L}_B\psi(\mathbf{\Omega}) = \int_{4\pi}\left(\sum_{l=0}^{\infty}\frac{2l+1}{4\pi}f_l P_l(\mathbf{\Omega}\cdot\mathbf{\Omega}')\right)\psi(\mathbf{\Omega}')d\mathbf{\Omega}' - \psi(\mathbf{\Omega}) \tag{2.43}$$

$$\mathcal{L}_{FP}\psi(\mathbf{\Omega}) = \left(\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \frac{1}{1-\mu^2}\frac{\partial^2}{\partial\varphi^2}\right)\psi(\mathbf{\Omega}) \tag{2.44}$$

$$\mathcal{L}_1\psi(\mathbf{\Omega}) = \frac{1}{4\pi}\int_{4\pi}\frac{\psi(\mathbf{\Omega}')-\psi(\mathbf{\Omega})}{1-\mathbf{\Omega}'\cdot\mathbf{\Omega}}d\mathbf{\Omega}' \tag{2.45}$$

$$\mathcal{L}_{3/2}\psi(\mathbf{\Omega}) = \frac{1}{4\pi\sqrt{2}}\int_{4\pi}\frac{\psi(\mathbf{\Omega}')-\psi(\mathbf{\Omega})}{(1-\mathbf{\Omega}'\cdot\mathbf{\Omega})^{3/2}}d\mathbf{\Omega}', \tag{2.46}$$

where f_l are the Legendre expansion coefficient of $f(\mathbf{\Omega}\cdot\mathbf{\Omega}')d\mathbf{\Omega}$ which is the probability that a particle, entering a scattering event with a direction $\mathbf{\Omega}'$, will emerge from the event with a direction in $d\mathbf{\Omega}$ about $\mathbf{\Omega}$. If $\psi(\mathbf{\Omega})$ is sufficiently smooth, equations (2.43) to (2.46) become:

$$\mathcal{L}_B\psi(\mathbf{\Omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}(-(1-f_l))Y_l^m(\mathbf{\Omega})\int_{4\pi}Y_l^{m,*}(\mathbf{\Omega}')\psi(\mathbf{\Omega}')d\mathbf{\Omega}' \tag{2.47}$$

$$\mathcal{L}_{FP}\psi(\mathbf{\Omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}(-l(l+1))Y_l^m(\mathbf{\Omega})\int_{4\pi}Y_l^{m,*}(\mathbf{\Omega}')\psi(\mathbf{\Omega}')d\mathbf{\Omega}' \tag{2.48}$$

$$\mathcal{L}_1\psi(\mathbf{\Omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}\left(-\sum_{k=1}^{l}\frac{1}{k}\right)Y_l^m(\mathbf{\Omega})\int_{4\pi}Y_l^{m,*}(\mathbf{\Omega}')\psi(\mathbf{\Omega}')d\mathbf{\Omega}' \tag{2.49}$$

$$\mathcal{L}_{3/2}\psi(\mathbf{\Omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}(-l)Y_l^m(\mathbf{\Omega})\int_{4\pi}Y_l^{m,*}(\mathbf{\Omega}')\psi(\mathbf{\Omega}')d\mathbf{\Omega}'. \tag{2.50}$$

Therefore, the spherical harmonics are eigenfunctions of equations (2.47) to (2.50):

$$\mathcal{L}_BY_l^m(\mathbf{\Omega}) = -(1-f_l)Y_l^m(\mathbf{\Omega}) \tag{2.51}$$

$$\mathcal{L}_{FP}Y_l^m(\mathbf{\Omega}) = -l(l+1)Y_l^m(\mathbf{\Omega}) \tag{2.52}$$

$$\mathcal{L}_1Y_l^m(\mathbf{\Omega}) = \left(-\sum_{k=1}^{l}\frac{1}{k}\right)Y_l^m(\mathbf{\Omega}) \tag{2.53}$$

$$\mathcal{L}_{3/2}Y_l^m(\mathbf{\Omega}) = -lY_l^m(\mathbf{\Omega}). \tag{2.54}$$

b Henyey-Greenstein scattering kernel

The Henyey-Greenstein [49] differential scattering kernel is defined by [62]:

$$f(\mu_0) = \frac{1-\bar{\mu}_0^2}{4\pi(1-2\bar{\mu}_0\mu+\bar{\mu}_0^2)^{3/2}}. \tag{2.55}$$

This kernel was proposed in [49], and is widely used mainly because of its great simplicity. The exact Henyey-Greenstein kernel does not arise from a deeper theory of any known physical process [62]. The Legendre polynomial expansion of the kernel, $f(\mu_0) = \sum_{l=0}^{\infty} \frac{2l+1}{2} f_l P_l(\mu_0)$, is very simple since the expansion coefficients, f_l, are given by:

$$f_l = \bar{\mu}_0^l, \; l \geq 0. \tag{2.56}$$

We look at:

$$\Sigma_s \mathcal{L}_B = \Sigma_s \left(\int_{4\pi} f(\mathbf{\Omega} \cdot \mathbf{\Omega'}) \psi(\mathbf{r}, \mathbf{\Omega'}, E) d\mathbf{\Omega'} - \psi(\mathbf{r}, \mathbf{\Omega}, E) \right) \tag{2.57}$$

with the scaled Henyey-Greenstein kernel:

$$\Sigma_s = \frac{\hat{\Sigma}_s}{\epsilon} \tag{2.58}$$

$$\bar{\mu}_0 = 1 - \epsilon, \tag{2.59}$$

where $\hat{\Sigma}_s$ is fixed and $\epsilon \approx 0$. When the scattering mean free path tends to zero, the mean scattering cosine tends to one. In the absence of absorption, the transport cross section is independent of ϵ:

$$\Sigma_{tr} = \Sigma_s(1 - \bar{\mu}_0) = \frac{\hat{\Sigma}_s}{\epsilon} \cdot \epsilon = \hat{\Sigma}_s. \tag{2.60}$$

When ϵ tends to zero, the eigenvalues of $\Sigma_s \mathcal{L}_s$ become:

$$\begin{aligned} -\Sigma_s(1 - f_l) &= -\frac{\hat{\Sigma}_s}{\epsilon} \left(1 - (1-\epsilon)^l\right) \\ &= -\frac{\hat{\Sigma}_s}{\epsilon} \left(l\epsilon - \frac{l(l-1)}{2}\epsilon^2 + O(\epsilon^3) \right) \\ &= \hat{\Sigma}_s(1+\epsilon)(-l) - \frac{\hat{\Sigma}_s \epsilon}{2}(-l(l+1)) + O(\epsilon^2). \end{aligned} \tag{2.61}$$

Using equations (2.51), (2.52), (2.54) and (2.61), we get:

$$\Sigma_s \mathcal{L}_B = \hat{\Sigma}_s (1+\epsilon)\mathcal{L}_{3/2} - \frac{\hat{\Sigma}_s \epsilon}{2}\mathcal{L}_{FP} + O(\epsilon^2). \tag{2.62}$$

The leading-order approximation, $\mathcal{O}(1)$, to $\Sigma_s \mathcal{L}_B$ is:

$$\Sigma_s \mathcal{L}_B = \Sigma_s (1-\bar{\mu}_0)\mathcal{L}_{3/2} + O(\epsilon). \tag{2.63}$$

The first-order approximation, $\mathcal{O}(\epsilon)$, to $\mathcal{L}_B$ is given by:

$$\Sigma_s \mathcal{L}_B = \Sigma_s (1-\bar{\mu}_0)(2-\bar{\mu}_0)\mathcal{L}_{3/2} - \frac{\Sigma_s}{2}(1-\bar{\mu}_0)^2 \mathcal{L}_{FP} + O(\epsilon^2). \tag{2.64}$$

We see that the Fokker-Planck operator is not an asymptotic limit of the Henyey-Greenstein kernel because $\mathcal{L}_{FP}$ appears only in the first order approximation. The zeroth-order approximation contains only $\mathcal{L}_{3/2}$.

c *Screened Rutherford scattering kernel*

The screened Rutherford differential scattering kernel, which is widely used to model the Coulomb scattering of non-relativistic electrons, is defined by:

$$f(\mu_0) = \frac{\eta(1+\eta)}{\pi(1+2\eta-\mu_0)^2}, \tag{2.65}$$

where $\eta > 0$ is the screening parameter. We consider $\eta \approx 0$, for which $f(\mu_0)$ is the most forward-peaked. The first two Legendre expansion coefficient of $f(\mu_0)$ are given by:

$$f_0 = 1 \tag{2.66}$$

$$f_1 = \bar{\mu}_0 = 1 - 2\eta\left((1+\eta)\ln\left(1+\frac{1}{\eta}\right) - 1\right). \tag{2.67}$$

Using equation (2.65), $f_k = 2\pi \int_{-1}^{1} P_k(\mu) f(\mu)\ d\mu$, and:

$$\mu P_k(\mu) = \frac{(k+1)P_{k+1}(\mu) + kP_{k-1}(\mu)}{2k+1}, \tag{2.68}$$

we obtain the following recurrence formula:

$$kf_{k+1} - (1+2\eta)(2k+1)f_k + (k+1)f_{k-1} = 0, \tag{2.69}$$

or equivalently:

$$\frac{f_{k+1} - f_k}{k+1} - \frac{f_k - f_{k-1}}{k} = 2\eta\frac{2k+1}{k(k+1)}f_k. \tag{2.70}$$

Summing equation (2.70) from $k = 1$ to $m - 1$, using equation (2.67), and then multiplying by m yields:

$$f_m - f_{m-1} - (\bar{\mu}_0 - 1)m = 2\eta m \sum_{k=1}^{m-1} \frac{2k+1}{k(k+1)} f_k. \tag{2.71}$$

Summing equation (2.71) from $m = 1$ to l yields:

$$\begin{aligned} f_l - 1 - (\bar{\mu}_0 - 1)\frac{l(l+1)}{2} &= 2\eta \sum_{m=1}^{l} m \sum_{k=1}^{m-1} \frac{2k+1}{k(k+1)} f_k \\ &= 2\eta \sum_{k=1}^{l-1} \left(\sum_{m=k+1}^{l} m\right) \frac{2k+1}{k(k+1)} f_k \\ &= 2\eta \sum_{k=1}^{l-1} \left(\frac{l(l+1)}{2} - \frac{k(k+1)}{2}\right) \frac{2k+1}{k(k+1)} f_k, \end{aligned} \tag{2.72}$$

where we have used:

$$\sum_{m=1}^{l} m = \frac{l(l+1)}{2}. \tag{2.73}$$

Finally, we find the f_l coefficient:

$$f_l = 1 - (1-\bar{\mu}_0)\frac{l(l+1)}{2} + \eta \sum_{k=1}^{l-1} \left(\frac{l(l+1)}{k(k+1)} - 1 \right) (2k+1) f_k. \tag{2.74}$$

If the summation term is assumed to be zero for $l = 0$ and $l = 1$; equation (2.69) and equation (2.74) are equivalent for all $l \geq 0$ and $\eta > 0$. Next, we assume the following expansion of f_l for $\eta \ll 1$:

$$f_l = f_{l,0} + \eta f_{l,1} + O(\eta^2). \tag{2.75}$$

Introducing this expansion into equation (2.74) and equating the coefficients of η^0 and η^1 (note that $\bar{\mu}_0$ is not expanded for $\eta \ll 1$), we find:

$$f_{l,0} = 1 - (1-\bar{\mu}_0)\frac{l(l+1)}{2} \tag{2.76}$$

$$f_{l,1} = \sum_{k=1}^{l-1} \left(\frac{l(l+1)}{k(k+1)} - 1 \right) (2k+1) \left(1 - (1-\bar{\mu}_0)\frac{k(k+1)}{2} \right) \tag{2.77}$$

From equation (2.67), we know that:

$$1 - \bar{\mu}_0 = O\left(\eta \ln \left(\frac{1}{\eta} \right) \right). \tag{2.78}$$

Therefore, we get:

$$\begin{aligned} f_l =& 1 + \left(\frac{1-\bar{\mu}_0}{2}\right)(-l(l+1)) + \eta\left(\sum_{k=1}^{l-1}\left(\frac{l(l+1)}{k(k+1)} - 1\right)(2k+1)\right) \\ &+ O\left(\eta^2 \ln\left(\frac{1}{\eta}\right)\right). \end{aligned} \tag{2.79}$$

Now using equation (2.73), we get:

$$\sum_{k=1}^{l-1}\left(\frac{l(l+1)}{k(k+1)} - 1\right)(2k+1) = 2l(l+1)\left(\left(\sum_{k=1}^{l}\frac{1}{k}\right) - 1\right) \tag{2.80}$$

for $l \geq 0$.

Thus, equation (2.79) becomes:

$$\begin{aligned} f_l =& 1 + \left(\frac{1-\bar{\mu}_0}{2}\right)\big(-l(l+1) + 2\eta\,(-l(l+1))\big)\left(1 - \sum_{k=1}^{l}\frac{1}{k}\right) \\ &+ O\left(\eta^2 \ln\left(\frac{1}{\eta}\right)\right). \end{aligned} \tag{2.81}$$

Next, we define Σ_s as:

$$\Sigma_s = \frac{\hat{\Sigma}_s}{1-\bar{\mu}_0}. \tag{2.82}$$

Then for $\eta \ll 1$, the eigenvalues of $\Sigma_s \mathcal{L}_B$ are:

$$-\Sigma_s(1-f_l) = \frac{\hat{\Sigma}_s}{2}(-l(l+1)) + \frac{2\eta\hat{\Sigma}_s}{1-\bar{\mu}_0}(-l(l+1))\left(1 - \sum_{k=1}^{l}\frac{1}{k}\right) + O(\eta). \tag{2.83}$$

Using equation (2.53), we finally get:

$$\begin{aligned} \Sigma_s \mathcal{L}_B &= \frac{\sigma}{2}\mathcal{L}_{FP} + \frac{2\eta\sigma}{1-\bar{\mu}_0}\mathcal{L}_{FP}(I + \mathcal{L}_1) + O(\eta) \\ &= \frac{\Sigma_s(1-\bar{\mu}_0)}{2}\mathcal{L}_{FP} + 2\eta\Sigma_s\mathcal{L}_{FP}(I+\mathcal{L}_1) + O(\eta). \end{aligned} \tag{2.84}$$

The first term in this expansion is the standard Fokker-Planck operator which is $O(1)$. The second term in equation (2.84) is $O(1/\ln(1/\eta))$. Therefore, $1/\ln(1/\eta)$ must be very small for the Fokker-Planck description to be valid. Because of this logarithmic behavior, realistic values of η are typically not small enough [88].

In conclusion, we see that the Fokker-Planck operator appears in the asymptotic approximations of $\mathcal{L}_B$ for both Henyey-Greenstein and screened Rutherford scattering. However, for the Henyey-Greenstein scattering, $\mathcal{L}_{FP}$ is dominated by a pseudodifferential operator $\mathcal{L}_{3/2}$, and for the screened Rutherford scattering, $\mathcal{L}_{FP}$ weakly dominates another pseudodifferential operator $\mathcal{L}_{FP}\mathcal{L}_1$. In neither case, the first-order approximations of $\mathcal{L}_B$ can be written only with $\mathcal{L}_{FP}$ [62].

C Fokker-Planck cross section

In this section, we derive the Fokker-Planck scattering cross section such that the Fokker-Planck operator can be approximated by the Boltzmann operator. First, we recall the BFP equation equation (2.23):

$$\begin{aligned}\mathbf{\Omega}\cdot\mathbf{\nabla}\psi + (\Sigma_{s,reg}+\Sigma_a)\psi = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega})\int_E^{E/\alpha}\Bigg(\Sigma_{s,l,reg}(E'\to E)\\ \phi_{l,m}(E')\Bigg)dE' + \frac{\partial(S\psi)}{\partial E} + \frac{\alpha}{2}\left(\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \frac{1}{(1-\mu^2)}\frac{\partial^2}{\partial\varphi^2}\right)\psi + Q,\end{aligned} \tag{2.85}$$

where we used $\alpha = 2T$. Let us define:

$$\mathcal{L}_{FP}^{\alpha}\psi = \frac{\alpha}{2}\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu}\psi \tag{2.86}$$

$$\mathcal{L}_{FP}^{e}\psi = \frac{\partial}{\partial E}S\psi + \frac{1}{2}\frac{\partial^2}{\partial E^2}R\psi. \tag{2.87}$$

We note that $\mathcal{L}_{FP}^{\alpha}$ causes particles to redistribute in direction without energy change, while $\mathcal{L}_{FP}^{e}$ causes particles to redistribute particles in energy without directional change. Therefore, $\mathcal{L}_{FP}^{\alpha}$ can be approximated using the following cross section:

$$\Sigma_s(\mu_0, E' \to E) = \Sigma_s^{\alpha}(\mu_0, E)\delta(E' - E), \tag{2.88}$$

where $\Sigma_s^{\alpha}(\mu_0, E) = \frac{\alpha(E)}{1-\mu_s}\frac{1}{2\pi}\delta(\mu_0 - \mu_s)$ and μ_s is a parameter; while $\mathcal{L}_{FP}^{e}$ should be approximated by a cross section of the form:

$$\Sigma_s(\mu_0, E' \to E) = \Sigma_s^{e}(E' \to E)\frac{1}{2\pi}\delta(\mu_0 - 1). \tag{2.89}$$

1 Legendre polynomial expansion of Σ_s^{α}

Next, we express the Legendre polynomial expansion of $\Sigma_s^{\alpha}(\mu_0)$ (E was dropped for brevity) as it has been done in [59, 69, 72]. We will focus on $\mathcal{L}_{FP}^{\alpha}$ since this research we solve the energy-integrated Boltzmann equation and, therefore, the $\mathcal{L}_{FP}^{e}$ operator does not appear in the equation that we solve. Because Σ_s^{α} does not change particle energy, it corresponds to a within-group cross section. We define:

$$\mathcal{L}_{B}^{\alpha} = \Sigma_s^{\alpha}\mathcal{L}_B \tag{2.90}$$

$$\mathcal{L}_{FP}^{\alpha} = \frac{\alpha}{2}\mathcal{L}_{FP}, \tag{2.91}$$

and thus:

$$\mathcal{L}_{B}^{\alpha}Y_l^m(\mathbf{\Omega}) = (\Sigma_{s,l}^{\alpha} - \Sigma_{s,0}^{\alpha})Y_l^m(\mathbf{\Omega}) \tag{2.92}$$

$$\mathcal{L}_{FP}^{\alpha}Y_l^m(\mathbf{\Omega}) = -\frac{\alpha}{2}l(l+1)Y_l^m(\mathbf{\Omega}). \tag{2.93}$$

Using equations (1.12), (2.92) and (2.93), we can define Σ_s^{α} such that:

$$\mathcal{L}_{B}^{\alpha}\psi = \mathcal{L}_{FP}^{\alpha}\psi, \tag{2.94}$$

by setting:

$$\Sigma_{s,l}^{\alpha} - \Sigma_{s,0}^{\alpha} = -\frac{\alpha}{2} l(l+1), \tag{2.95}$$

with $l = 1, \ldots, L$. Choosing $\Sigma_{s,L}^{\alpha} = 0$ to minimize $\Sigma_{s,0}^{\alpha}$ and equation (2.95) becomes:

$$\Sigma_{s,l}^{\alpha} = \frac{\alpha}{2} \left(L(L+1) - l(l+1) \right), \quad l = 0, \ldots, L. \tag{2.96}$$

Using appropriate quadrature sets and expansion orders, the S_n representation of $\mathcal{L}_{FP}^{\alpha}$ is equivalent to the one obtained by interpolating the discrete angular flux values with a polynomial.

Next, we look at the behavior of Σ_s^{α} when the degree of the expansion is increased. First, we should note that the momentum transfer of Σ_s^{α} is exact for any expansion order:

$$\begin{aligned} 2\pi \int_{-1}^{1} \Sigma_s^{\alpha}(\mu_0)(1-\mu_0) d\mu_0 &= \Sigma_{s,0} - \Sigma_{s,1} \\ &= \frac{\alpha}{2} L(L+1) - \frac{\alpha}{2}(L(L+1)-2) \\ &= \alpha. \end{aligned} \tag{2.97}$$

With equation (2.96), the average cosine of the scattering angle becomes:

$$\begin{aligned} \bar{\mu}_0 &= \frac{\Sigma_{s,1}^{\alpha}}{\Sigma_{s,0}^{\alpha}} \\ &= \frac{L(L+1)-2}{L(L+1)}. \end{aligned} \tag{2.98}$$

It is easily seen that when L increases, $\bar{\mu}_0$ goes to one and Σ_s^{α} becomes increasingly forward-peaked. The total magnitude of $\Sigma_s^{\alpha}(\mu_0)$ becomes unlimited when L goes to ∞:

$$\Sigma_{s,0}^{\alpha} = \frac{\alpha}{2} L(L+1). \tag{2.99}$$

This shows that $\mathcal{L}_{FP}^{\alpha}$ corresponds to a continuous-deflection interaction. The particles are continuously deflected with the mean deflection per unit pathlength given by the momentum transfer.

$\mathcal{L}_{B}^{\alpha}$ converges to $\mathcal{L}_{FP}^{\alpha}$ when μ_s tends to one but it does not converge uniformly. For any fixed value of μ_s, the high-order eigenvalues of $\mathcal{L}_{FP}^{\alpha}$ are grossly underestimated by $\mathcal{L}_{B}^{\alpha}$. Fortunately, this error in the high-order eigenvalues is usually unimportant [72].

D Galerkin quadratures

Until now, we have not described which angular quadrature should be used to correctly treat high orders of anisotropy. We have only stated that we need an appropriate quadrature but we did not explain what we required. In this section, we introduce the Galerkin quadrature. Morel first introduced them in [68]; here, we will introduce them following the presentation made in [85].

First, we start by recalling the definition of ψ and $\phi_{l,m}$:

$$\begin{aligned}\phi_{l,m}(\mathbf{r}) &= \int_{4\pi} d\mathbf{\Omega}'\psi(\mathbf{r},\mathbf{\Omega}')Y_l^{m,*} \\ &= (\boldsymbol{D}\Psi)_{l,m}\end{aligned} \tag{2.100}$$

where $\boldsymbol{D}$ is the direction-to-moment operator. We also have:

$$\begin{aligned}\Psi(\mathbf{r},\mathbf{\Omega}) &= \sum_{l=0}^{\infty}\sum_{m=-l}^{l} Y_l^m(\mathbf{\Omega})\phi_{l,m}(\mathbf{r}) \\ &= \boldsymbol{M}\Phi(\mathbf{r})\end{aligned} \tag{2.101}$$

where $\boldsymbol{M}$ is the moment-to-direction operator. By combining equations (2.100) and (2.101), we obtain:

$$(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{D})\Psi = 0 \tag{2.102}$$

since for analytic transport $\boldsymbol{M} = \boldsymbol{D}^{-1}$. $\boldsymbol{I}$ is the identity operator.

Now, we define [85]:

$$\epsilon = 1 - \langle \bar{\mu}_0 \rangle \tag{2.103}$$

$$\gamma = \left\langle \overline{(1-\mu_0)^2} \right\rangle \tag{2.104}$$

$$A(\mathbf{r}) = \frac{1 - \bar{\mu}_0(\mathbf{r})}{\epsilon} \tag{2.105}$$

$$\Sigma_a = \hat{\Sigma}_a \tag{2.106}$$

$$\Sigma_{s,l} = \frac{\hat{\Sigma}_{s,l}(\mathbf{r})}{\epsilon} \tag{2.107}$$

$$\Sigma_{s,l}(\mathbf{r}) = \Sigma_{s,0}(\mathbf{r}) \left(1 - \frac{l(l+1)}{2} A(\mathbf{r})\epsilon + O(\gamma) \right) \tag{2.108}$$

where $\langle X \rangle$ is a typical value of X. Using equations (2.103) to (2.108) in equation (1.16), we get:

$$\begin{aligned} &\boldsymbol{\Omega}_k \cdot \boldsymbol{\nabla} \psi_k(\mathbf{r}) + \left(\Sigma_a(\mathbf{r}) + \Sigma_{s,0}(\mathbf{r})\right) \psi_k(\mathbf{r}) = \\ &\sum_{l=0}^{N-1} \sum_{m=-l}^{l} Y_l^m(\boldsymbol{\Omega}_k) \phi_{l,m}(\mathbf{r}) \frac{\hat{\Sigma}_{s,0}(\mathbf{r})}{\epsilon} \left(1 - \frac{l(l+1)}{2} A(\mathbf{r})\epsilon + O(\gamma) \right) + Q(\mathbf{r}, \boldsymbol{\Omega}_k) \end{aligned} \tag{2.109}$$

where:

$$\psi_k(\mathbf{r}) = \psi(\mathbf{r}, \boldsymbol{\Omega}_k) \tag{2.110}$$

$$\phi_{l,m}(\mathbf{r}) = \sum_{k=1}^{K} w_k Y_l^{m,*}(\boldsymbol{\Omega}_k) \psi_k(\mathbf{r}). \tag{2.111}$$

w_k and $\boldsymbol{\Omega}_k$ are the quadrature weights and directions of a quadrature set of order N. For triangular quadrature sets, $K = N$ in 1D, $K = \frac{N(N+2)}{2}$ in 2D and $K = N(N+2)$

in 3D. Equation (2.109) yields:

$$\begin{aligned}
&\boldsymbol{\Omega}_k \cdot \boldsymbol{\nabla}\psi_k(\mathbf{r}) + \hat{\Sigma}_a(\mathbf{r})\psi_k(\mathbf{r}) + \frac{\hat{\Sigma}_{s,0}(\mathbf{r})}{\epsilon}\left(\psi_k(\mathbf{r}) - \sum_{l=0}^{N-1}\sum_{m=-l}^{m} Y_l^m(\boldsymbol{\Omega}_k)\phi_{l,m}\right) \\
&= -\frac{\left(\Sigma_{tr}(\mathbf{r}) - \hat{\Sigma}_a(\mathbf{r})\right)}{2}\sum_{l=0}^{N-1}\sum_{m=-l}^{l} l(l+1)Y_l^m(\boldsymbol{\Omega}_k)\phi_l^m(\mathbf{r}) \\
&+ Q(\mathbf{r},\boldsymbol{\Omega}_k) + O\left(\frac{\gamma}{\epsilon}\right).
\end{aligned} \tag{2.112}$$

We insert the asymptotic ansatz:

$$\psi = \psi^{(0)} + \epsilon\psi^{(1)} + \epsilon^2\psi^{(2)} + \dots \tag{2.113}$$

$$\phi_{l,m} = \phi_{l,m}^{(0)} + \epsilon\phi_{l,m}^{(1)} + \epsilon^2\phi_{l,m}^{(2)} + \dots \tag{2.114}$$

into equation (2.109). The terms of $O(1)$ give:

$$\begin{aligned}
\phi_{l,m}^{(0)}(\mathbf{r}) &= \sum_{k=1}^{K} w_k Y_l^{m,*}(\boldsymbol{\Omega}_k)\psi_k^{(0)}(\mathbf{r}) \\
&= (\boldsymbol{D}_N\Psi^{(0)})_{l,m}.
\end{aligned} \tag{2.115}$$

Now we insert the ansatz into equation (2.112) and we look at the terms of $O(\epsilon^{-1})$:

$$\begin{aligned}
\psi_d^{(0)}(\mathbf{r}) &= \sum_{l=0}^{N-1}\sum_{m=-l}^{l} Y_l^m(\boldsymbol{\Omega}_k)\phi_{l,m}^{(0)}(\mathbf{r}) \\
&= (\boldsymbol{M}_N\Phi^{(0)})_d
\end{aligned} \tag{2.116}$$

there is no $O(\gamma)$ term, since $\gamma \to 0$ when $\epsilon \to 0$, i.e., that there are no $O(1)$ components in γ. Equations (2.115) and (2.116) may be combined to give:

$$(\boldsymbol{I} - \boldsymbol{M}_N \boldsymbol{D}_N)\Psi^{(0)} = 0 \tag{2.117}$$

$$(\boldsymbol{I} - \boldsymbol{D}_N \boldsymbol{M}_N)\Phi^{(0)} = 0. \tag{2.118}$$

Therefore, $\Psi^{(0)}$ must be in the kernel of $\boldsymbol{I} - \boldsymbol{M}_N \boldsymbol{D}_N$ and $\Phi^{(0)}$ must be in the kernel of $\boldsymbol{I} - \boldsymbol{D}_N \boldsymbol{M}_N$. Using equation (2.111), equation (2.118) becomes successively:

$$(\boldsymbol{I} - \boldsymbol{D}_N \boldsymbol{M}_N)\, \boldsymbol{D}_N \Psi^{(0)} = 0 \tag{2.119}$$

$$(\boldsymbol{D}_N - \boldsymbol{D}_N \boldsymbol{M}_N \boldsymbol{D}_N)\, \Psi^{(0)} = 0 \tag{2.120}$$

$$\boldsymbol{D}_N (\boldsymbol{I} - \boldsymbol{M}_N \boldsymbol{D}_N)\Psi^{(0)} = 0, \tag{2.121}$$

which is always satisfied if equation (2.117) is satisfied. Therefore, if equation (2.117) is satisfied, equations (2.115) and (2.116) are automatically satisfied.

A sufficient condition to satisfy equation (2.117) is that $\boldsymbol{M}_N \boldsymbol{D}_N = \boldsymbol{I}$. This is of course true if $\boldsymbol{D}_N = \boldsymbol{M}_N^{-1}$ like in analytic transport. Obviously for $\boldsymbol{M}_N$ and $\boldsymbol{D}_N$ to be the inverse of each other, the matrices have to be square. Thus, the number of moments in the scattering expansion must be equal to number of discrete angles. In one-dimension, $\boldsymbol{D}_N = \boldsymbol{M}_N^{-1}$ is satisfied if the quadrature set integrates exactly any polynomials of degree $2N - 1$, like the Gauss-Legendre quadrature does. In multidimensional problems, the standard quadrature sets use more discrete angles than there are scattering moments. Therefore, $\boldsymbol{M}_N$ and $\boldsymbol{D}_N$ are rectangular matrices and they cannot be inverse of each other. In this case, equation (2.117) can only be satisfied if $\Psi^{(0)}$ is in the kernel of $(\boldsymbol{I} - \boldsymbol{M}_N \boldsymbol{D}_N)$. This can be achieved only if $\Psi^{(0)}$ satisfies nonphysical constraints [85]. If equation (2.117) is not satisfied, the asymptotic ansatz is not valid and there is no $O(1)$ solution to equations (2.109) and (2.111).

If we assume that $\boldsymbol{M}_N\boldsymbol{D}_N = \boldsymbol{I}$, then the $O(\epsilon)$ terms in equation (2.111) give:

$$\begin{aligned}\phi_{l,m}^{(1)}(\mathbf{r}) &= \sum_{k=1}^{K} w_k Y_l^{m,*}(\boldsymbol{\Omega}_k)\psi_k^{(1)}(\mathbf{r}) \\ &= (\boldsymbol{D}_N\Psi^{(1)})_{l,m}.\end{aligned} \tag{2.122}$$

In equation (2.112), the $O(1)$ terms give:

$$\begin{aligned}&\boldsymbol{\Omega}_k \cdot \boldsymbol{\nabla}\psi_k^{(0)}(\mathbf{r}) + \hat{\Sigma}_a(\mathbf{r})\psi_k^{(0)}(\mathbf{r}) + \hat{\Sigma}_{s,0}(\mathbf{r})\left(\psi_k^{(1)}(\mathbf{r}) - \sum_{n=0}^{N-1}\sum_{m=-l}^{l} Y_l^m(\boldsymbol{\Omega}_k)\phi_{l,m}^{(1)}(\mathbf{r})\right) \\ &= -\frac{\Sigma_{tr}(\mathbf{r}) - \hat{\Sigma}_a(\mathbf{r})}{2}\sum_{l=0}^{N-1}\sum_{m=-l}^{l} l(l+1)Y_l^m(\boldsymbol{\Omega}_k)\phi_{l,m}^{(0)}(\mathbf{r}) + Q(\mathbf{r},\boldsymbol{\Omega}_k) + O\left(\frac{\gamma}{\epsilon}\right).\end{aligned} \tag{2.123}$$

Now, we want the scattering term on the left side of equation (2.123) to disappear in order to keep only $\psi_k^{(0)}$ and $\phi_{l,m}^{(0)}$. This is possible only if:

$$\psi_d^{(1),*}(\mathbf{r}) = \psi_d^{(1)}(\mathbf{r}), \tag{2.124}$$

where:

$$\begin{aligned}\psi_d^{(1),*} &= \sum_{l=0}^{N-1}\sum_{m=-l}^{l} Y_l^m(\boldsymbol{\Omega}_k)\phi_{l,m}^{(1)}(\mathbf{r}) \\ &= (\boldsymbol{M}_N\Phi^{(1)})_d \\ &= (\boldsymbol{M}_N\boldsymbol{D}_N\Psi^{(1)})_d,\end{aligned} \tag{2.125}$$

which is satisfied because of our previous assumption that $\boldsymbol{M}_N\boldsymbol{D}_N = \boldsymbol{I}$. Equation (2.123) yields:

$$\begin{aligned}\mathbf{\Omega}_k \cdot \boldsymbol{\nabla}\psi_k^{(0)}(\mathbf{r}) + \hat{\Sigma}_a(\mathbf{r})\psi_k^{(0)}(\mathbf{r}) = \frac{\Sigma_{tr}(\mathbf{r}) - \hat{\Sigma}_a(\mathbf{r})}{2}\Bigg(\Bigg(\frac{\partial}{\partial\mu}(1-\mu^2)\frac{\partial}{\partial\mu} + \\ \left(\frac{1}{1-\mu^2}\right)\frac{\partial^2}{\partial\varphi^2}\Bigg)\tilde{\psi}^{(0)}(\mathbf{r},\mathbf{\Omega})\Bigg)_{\mathbf{\Omega}=\mathbf{\Omega}_k} + Q(\mathbf{r},\mathbf{\Omega}_k) + O\left(\frac{\gamma}{\epsilon}\right),\end{aligned} \tag{2.126}$$

where $k = 1,\dots,K$ and $\tilde{\psi}^{(0)}(\mathbf{r},\mathbf{\Omega})$ is an interpolant through the points $P_k = \{\mathbf{\Omega}_k, \psi_k^{(0)}(\mathbf{r},\mathbf{\Omega})\}$. "In one-dimensional geometry, $\tilde{\psi}^{(0)}(\mathbf{r},\mathbf{\Omega})$ is the $(N-1)-$order polynomial interpolant through the points P_k. In multidimensional geometries, $\tilde{\psi}^{(0)}(\mathbf{r},\mathbf{\Omega})$ is the spherical harmonic interpolant through the points P_k" [85]. A quadrature which satisfies the relation $\boldsymbol{D}_N = \boldsymbol{M}_N^{-1}$ is called a "Galerkin" quadrature because Morel derived it by using a Galerkin weighting method [68].

In [68], Morel made the following suggestions to find the correct limit using the S_N:

One-dimensional geometry: The Gauss-Legendre quadrature set is the only quadrature set which is a Galerkin quadrature.

Multidimensional geometry: The standard quadrature sets have fewer moments than discrete angles. Therefore to satisfy $\boldsymbol{D} = \boldsymbol{M}^{-1}$, spherical harmonics of higher order need to be added to increase the number of moments to equal the number of angular flux. Morel in [68] and Reed in [90] proposed an heuristic algorithm to choose the spherical harmonics for multidimensional geometries. If we use the real spherical harmonics defined by:

$$Y_{l,m}^c = \begin{cases} Y_l^0 & \text{if } m = 0 \\ \frac{1}{\sqrt{2}}\left(Y_l^m + (-1)^m Y_l^{-m}\right) & \text{if } m > 0 \end{cases} \tag{2.127}$$

$$Y^s_{l,m} = \frac{1}{i\sqrt{2}}(Y_l^{-m} - (-1)^m Y_l^m) \text{ if } m < 0 \tag{2.128}$$

and a two-dimensional Cartesian geometry triangular quadrature set, the selection rules to build $\boldsymbol{M}$ are the following:

- Take $Y^c_{l,m}$ if $l \in [0, N-1]$, $m \in [0, l]$, and $l + m$ is even.
- Take $Y^s_{l,m}$ if ($l \in [1, N-1]$, $m \in [1, l]$, and $l + m$ is even) or if ($l = N$, $m \in [1, l]$, and $l + m$ is even).

For three dimensional triangular quadrature sets, the spherical harmonics are:

- Take $Y^c_{l,m}$ if ($l \in [0, N-1]$ and $m \in [0, l]$) or if ($l = N$, $m \in [1, N-1]$, and m is odd).
- Take $Y^s_{l,m}$ if ($l \in [1, N]$ and $m \in [1, l]$) or if ($l = N + 1$, $m \in [2, N]$ and m is even).

General necessary conditions and explanations of the heuristic rules above have been analyzed in [91].

Using a standard S_n quadrature may lead to an unphysical solution for charged particle transport. To further demonstrate the importance of Galerkin quadrature, we define the scattering ratio matrix $\boldsymbol{C}$ by:

$$\boldsymbol{C} = \frac{1}{\Sigma_t}\boldsymbol{D}\boldsymbol{M}\boldsymbol{\Sigma}, \tag{2.129}$$

where $\boldsymbol{\Sigma}$ is the scattering matrix containing the moments of the scattering cross sections on its diagonal. $\boldsymbol{C}$ is a diagonal matrix whose entries are the scattering ratios $c_l = \frac{\Sigma_{s,l}}{\Sigma_t} \leq 1$. The fact that $\boldsymbol{M}\boldsymbol{D} = \boldsymbol{I}$ assures a one-to-one relation between the angular flux and the flux moments; furthermore, the orthogonality of all spherical harmonic functions used in the angular flux representation is preserved. However, if a standard quadrature is employed, then $\boldsymbol{D} = \boldsymbol{M}^T\boldsymbol{W}$, where $\boldsymbol{W}$ is the diagonal matrix of weights, and an inexact integration occurs for the set of spherical harmonics than span $\boldsymbol{M}$. Thus, $\boldsymbol{C}$ will differ from $\frac{\boldsymbol{\Sigma}}{\Sigma_t}$ and its eigenvalue could exceed unity. This is equivalent to numerically adding multiplication into the medium and, depending on the amount of leakage present in the geometrical configuration, the steady-state transport equation may not reach a steady state solution [85].

A very important property of the Galerkin quadratures is that they treat delta function scattering exactly. This is very important for two reasons. Firstly, in charged-particle transport, some cross sections have the following form:

$$\Sigma^{k\to g}(\mu_0) = \Sigma^{k\to g}\delta(\mu_0 - 1), \tag{2.130}$$

where $\Sigma^{k\to g}(\mu_0)$ is the differential cross section associated with a transfer from group k to group g. Thus, it is essential that delta function scattering be treated exactly. Secondly and more importantly since we work with energy-integrated equation, for electron scattering, the extended transport correction [63] can be used to reduce the within-group scattering cross sections by two orders of magnitude or more [67]. Without the extended transport correction, the scattering cross-section moments are very large and DSA does not accelerate the convergence of the solver. Because the Galerkin method treats delta function scattering exactly, the extended transport correction does not modify the solution of the problem [67]. This is very interesting since it allows significant reduction of the cross-section moments without loss of accuracy. Showing this property is quite simple given what we have presented thus far. We know that in the analytic case we have:

$$\int_{4\pi} \delta(\mu_0 - 1)\psi(\boldsymbol{\Omega}')\ d\boldsymbol{\Omega}' = \psi(\boldsymbol{\Omega}). \tag{2.131}$$

When the scattering source is discretized, we have $\boldsymbol{\Sigma} = \boldsymbol{I}$. Therefore, it is obvious that:

$$\boldsymbol{D}\boldsymbol{\Sigma}\boldsymbol{M}\Psi = \Psi. \tag{2.132}$$

To really understand the Galerkin quadrature, it is interesting to recall the development made in [68]. First, we expand the one-dimensional angular flux on an interpolatory trial space:

$$\Psi(\mu) = \sum_{d=1}^{N} \psi_d B_d(\mu). \tag{2.133}$$

Methods for generating the interpolatory basis function can be found in [68]. Next, we expand the scattering source on the Legendre polynomials:

$$\mathcal{S}(\mu) = \sum_{l=0}^{\infty} \frac{2l+1}{2} \xi_l P_l(\mu), \tag{2.134}$$

where:

$$\xi_l = \int_{-1}^{1} \mathcal{S}(\mu) P_l(\mu) d\mu. \tag{2.135}$$

Now, we use the interpolatory trial space to approximate $S(\mu)$:

$$\tilde{\mathcal{S}}(\mu) = \sum_{d=1}^{N} \tilde{\mathcal{S}}_d B_d(\mu). \tag{2.136}$$

Since a Galerkin method is used, the residual associated with the trial space approximation must be orthogonal to the weighting space. The residual associated with equation (2.136) is given by:

$$Res(\mu) = \tilde{\mathcal{S}}(\mu) - \mathcal{S}(\mu). \tag{2.137}$$

Since the Legendre polynomials form a basis for the weighting space, we orthogonalize against the Legendre polynomials:

$$\int_{-1}^{1} Res(\mu)P_l(\mu)d\mu = 0.0, \quad l = 0, \dots, N-1. \tag{2.138}$$

Equation (2.138) is satisfied if:

$$\tilde{\xi}_l = \xi_l, \quad l = 0, \dots, N-1 \tag{2.139}$$

$$\tilde{\xi}_l = \int_{-1}^{1} \tilde{\mathcal{S}}(\mu)P_l(\mu)d\mu. \tag{2.140}$$

The main idea of the Galerkin quadrature can be seen on equation (2.139): "the discrete scattering source values are chosen such that the interpolatory representation for that scattering source has the same Legendre moments through degree $N-1$ as the exact scattering source calculated with the interpolatory representation for the angular flux" [68]. Because all the elements of the weighting space can be expressed as a linear combination of Legendre polynomials P_l with $l < N$ and $\int_{-1}^{1} P_l(x)P_m(x)\,dx = 0$ for $l \neq m$, only the first $N-1$ cross-section moments are needed.

Now, we compare the cross-section expansion for the Galerkin method and the S_n method. Both the Galerkin method and the standard S_n method use a Legendre expansion of the cross section. However, whereas the S_n method relies on the accuracy of the truncated Legendre expansion, the Galerkin scattering source is fully consistent with the exact cross section. It is not important whether or not the truncated Legendre cross-section expansion accurately represents the scattering cross section. What is important is that $\tilde{S}(\mu)$ is an accurate approximation of $S(\mu)$. This is a crucial property for charged particle transport. For example, the delta function expansion of finite degree is never converged, and this is an issue if a S_n method with a standard quadrature is used. However, as proved earlier, the scattering source computed by the Galerkin method is treated exactly.

If the scattering is isotropic, the S_n method might appear superior to the Galerkin, since the Galerkin requires the number of flux moments to be equal to the number of angular flux. Fortunately, if the higher order cross-sections moments are zero, only the first rows of $\boldsymbol{D}$ and the first columns of $\boldsymbol{M}$ need to be kept.

A Galerkin quadrature set and its companion quadrature sets (standard quadrature sets associated with the Galerkin quadrature set having weights defined by the first row of the discrete-to-moment matrix) yield different scattering only if the expansion order of the scattering cross section is high enough. When the scattering is isotropic, the Galerkin quadrature set is the same as its companion set. Even when the scattering is highly anisotropic, the results given by the Galerkin set and its companion set can be very close. This is due to the fact the high order moments of the scattering cross section are often, but not always, very small. However, the results can also be quite different.

CHAPTER III

A BRIEF REVIEW OF S_N TRANSPORT: ITERATIVE TECHNIQUES AND DISCRETIZATION ON ARBITRARY POLYGONS

A Introduction

In this Chapter, we briefly recall the S_n transport equations before presenting our research in the next chapters. We review the standard iterative techniques employed to solve the S_n transport equations and we introduce the two spatial discretizations used in this research: the BiLinear Discontinuous finite elements (BLD) [18, 73] and the PieceWise Linear Discontinuous finite elements (PWLD) [25, 95]. The BLD discretization is used on rectangular cells whereas PWLD is used on arbitrary convex polygonal cells. Both of these discretizations give the correct result in the diffusion limit.

In the following presentation, we restrict ourselves to one-group equations with isotropic scattering and source. With these simplifying assumptions, the fundamentals of S_n transport iterative solution techniques and spatial discretization can be presented effectively, while leaving new research material for subsequent chapters. These next chapters will build upon the concepts laid out here.

B The S_n Transport Equations

Given an angular quadrature set $\{\mathbf{\Omega}_d, w_d\}_{1\leq d\leq M}$, the one-group $_n$ transport equation with isotropic source and scattering is:

$$(\mathbf{\Omega}_d \cdot \mathbf{\nabla} + \Sigma_t(\mathbf{r}))\, \psi_d(\mathbf{r}) = \frac{1}{4\pi}\Sigma_s(\mathbf{r})\phi(\mathbf{r}) + \frac{1}{4\pi}S(\mathbf{r}), \quad \text{for } \mathbf{r} \in \mathcal{D},\ 1 \leq d \leq M, \qquad (3.1)$$

with $\psi_d(\mathbf{r}) = \psi(\mathbf{r}, \mathbf{\Omega}_d)$ the angular flux at position $\mathbf{r}$ in direction $\mathbf{\Omega}_d$, Σ_t, and Σ_s the total and scattering cross section, respectively, and $\mathcal{D}$ the spatial domain. The scalar flux is defined and evaluated as follows:

$$\psi(\mathbf{r}) \equiv \int_{4\pi} \psi(\mathbf{r}, \mathbf{\Omega}) \, d\mathbf{\Omega} \simeq \sum_{d=1}^{M} w_d \psi_d \left(\mathbf{r}\right). \tag{3.2}$$

The system of equations is closed assuming incoming boundary conditions on (with $\partial\mathcal{D} = \partial\mathcal{D}^d \cap \partial\mathcal{D}^r$):

$$\psi_d(\mathbf{r}_b) = \begin{cases} \psi_d^{inc}(\mathbf{r}_b), & \mathbf{r}_b \in \partial\mathcal{D}^{d,-} = \{\partial\mathcal{D}^d \text{ such that } \mathbf{\Omega}_d \cdot \boldsymbol{n}_b < 0\} \\ \psi_{d'}(\mathbf{r}_b), & \mathbf{r}_b \in \partial\mathcal{D}_d^{r,-} = \{\partial\mathcal{D}^r \text{ such that } \mathbf{\Omega}_d \cdot \boldsymbol{n} < 0\} \end{cases}, \tag{3.3}$$

where $\boldsymbol{n}_b = \boldsymbol{n}(\boldsymbol{r}_b)$ is the outward unit normal vector on the boundary. The reflecting direction of $\mathbf{\Omega}_d$ at a point $\mathbf{r}_b$ on the boundary is given by:

$$\mathbf{\Omega}_{d'} = \mathbf{\Omega}_d - 2\left(\mathbf{\Omega}_d \cdot \boldsymbol{n}_b\right)\boldsymbol{n}_b. \tag{3.4}$$

We assume the angular quadrature set satisfies the following two conditions for any outward unit normal vector on the reflecting boundary $\partial\mathcal{D}^r$:

- $\forall d = 1, \ldots, M$, the reflected direction $\mathbf{\Omega}_{d'}$ is also in the quadrature set (which is simple to obtain for rectangular geometries);

- the weights of the incident and reflected directions are equal, i.e., $w_d = w_{d'}$

For the time being, we assume that no reflective boundaries exist ($\partial\mathcal{D}^r = 0$). Then, equation (3.1) can be written in a compact form using operators:

$$\boldsymbol{L}\Psi = \boldsymbol{M\Sigma}\Phi + S = q, \tag{3.5}$$

$$\Phi = \boldsymbol{D}\Psi, \tag{3.6}$$

where Ψ is the vector of angular fluxes, Φ the vector flux moments (with isotropic scattering, the only moment required is the scalar flux), q is the total (scattering+external) source, $\boldsymbol{L}$ is the streaming operator, $\boldsymbol{M}$ is the moment-to-direction operator, and $\boldsymbol{D}$ is the direction-to-moment operator. $\boldsymbol{L} = diag(\boldsymbol{L}_1, \dots, \boldsymbol{L}_d, \dots, \boldsymbol{L}_M)$ is a diagonal operator; given the total source q, one can solve independently for the resulting angular fluxed in all directions. Equations (3.5) and (3.6) can be re-arranged in terms of the scalar flux only:

$$\Phi = \boldsymbol{DL}^{-1}\left(\boldsymbol{M\Sigma}\Phi + S\right). \tag{3.7}$$

The action of $\boldsymbol{DL}^{-1}$ is often referred to as a *transport sweep* because for any direction $\boldsymbol{\Omega}_d$, the action of $\boldsymbol{L}_d^{-1}$ can be obtained by traversing the mesh (i.e.,sweeping) in the direction of flow, successively inverting $\boldsymbol{L}_D$ in each set of downstream cells. Thus, one need only to solve a small linear system of equations, cell by cell. The order in which the elements are solved constitutes the graph of the sweep. We have thus far considered only situations where the graph does not present some dependencies (cycles). Note that cycles in the sweep graph can also appear due to reflective boundary conditions. These graph dependencies can either be lagged within the iterative procedure of the solution vector consisting of the scalar flux is augmented by the angular flux unknowns that cause the cycles. We will explain these details in the paragraph related to iterative techniques, but first we generalize our operator notations for situations where we need to keep in the solution vector both the flux moments and some angular fluxes due to dependencies in the graph (non convex meshed and/or reflective boundaries).

If the graph of the sweep presents dependencies, we practically break the transport sweep on these boundaries and introduce the notion of significant angular fluxes. In this situation, we define a matrix $\boldsymbol{N}$ that extracts from the entire angular flux vector all out-going angular fluxes on the boundaries causing a dependency in the graph of the sweep, i.e., the significant angular flux vector is given by:

$$\Psi_{SAF} = \boldsymbol{N}\Psi, \tag{3.8}$$

$\boldsymbol{N}$ is simply an operator that extracts from the entire angular flux vector, the values required to break the graph dependencies. We then split the loss and streaming operator $\boldsymbol{L}$ into two parts:

$$\boldsymbol{L} = \underline{\boldsymbol{L}} - \overline{\boldsymbol{L}}, \tag{3.9}$$

where $\underline{\boldsymbol{L}}$ is the lower block triangular matrix (which can be inverted during a transport sweeps) and $\overline{\boldsymbol{L}}$ is the strictly upper triangular block, causing the dependencies in the sweep ($\overline{\boldsymbol{L}}$ only contains integrals along some incoming edges of a cell). Note that $\boldsymbol{N}^T\boldsymbol{N}$ is a diagonal operator that contains one only for angular flux values that are labeled as "significant". Then, we have:

$$\overline{\boldsymbol{L}}\Psi = \overline{\boldsymbol{L}}\boldsymbol{N}^T\boldsymbol{N}\Psi = \overline{\boldsymbol{L}}\boldsymbol{N}^T\Psi_{SAF}, \tag{3.10}$$

and can, therefore, recast the transport equation as:

$$\underline{\boldsymbol{L}}\Psi = \overline{\boldsymbol{L}}\boldsymbol{N}^T\Psi_{SAF} + \boldsymbol{M\Sigma}\Phi + S, \tag{3.11}$$

$$\Phi = \boldsymbol{D}\Psi. \tag{3.12}$$

C Solution Techniques

1 Unaccelerated Procedures

Equations (3.5) and (3.6) can be solved using the Source Iteration (SI) method (a stationary iterative technique also known as Richardson iteration). The SI technique at the ℓ^{th} iteration is given by:

$$\Phi^{(\ell+1)} = \boldsymbol{D}\boldsymbol{L}^{-1}\left(\boldsymbol{M\Sigma}\Phi^{(\ell)} + S\right). \tag{3.13}$$

Alternatively, a subspace Krylov method (usually GMRES) can be employed to solve the following transport system of equations:

$$\left(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M\Sigma}\right)\Phi = \boldsymbol{D}\boldsymbol{L}^{-1}S. \tag{3.14}$$

Both the SI and the GMRES approaches require transport sweeps (the action of $\boldsymbol{L}^{-1}$ is required in both procedures).

When the scattering ratio $c = \frac{\Sigma_s}{\Sigma_t}$ tends to one in optically thick domains, the number of SI and GMRES iterations can become large. Fourier analyses (for continuous, i.e., undiscretized transport) confirmed that SI rapidly attenuates error modes associated with high frequencies (transport dominated modes) while leaving almost unaffected low-frequency error modes (diffusion dominated modes) [19]. To accelerate the convergence, a DSA preconditioner is needed. In addition, some level of consistency is necessary between the spatial discretization of the transport operator and than of the diffusion operator. In Chapter V, we will adapt the MIP discontinuous finite element discretization of the diffusion equation for arbitrary polygonal grids and employ MIP as a DSA preconditioner.

For completeness, we provide the SI and GMRES solution techniques in the case graph dependencies are present. During one SI iteration, the scalar flux and the angular significant flux are updated as follows:

$$\begin{bmatrix} \Phi \\ \Psi_{SAF} \end{bmatrix}^{(\ell+1)} = \begin{bmatrix} \boldsymbol{D}\underline{\boldsymbol{L}}^{-1}\boldsymbol{M\Sigma} & \boldsymbol{D}\underline{\boldsymbol{L}}^{-1}\overline{\boldsymbol{L}}^{-1}\boldsymbol{N}^T \\ \boldsymbol{N}\underline{\boldsymbol{L}}^{-1}\boldsymbol{M\Sigma} & \boldsymbol{N}\underline{\boldsymbol{L}}^{-1}\overline{\boldsymbol{L}}^{-1}\boldsymbol{N}^T \end{bmatrix} \begin{bmatrix} \Phi \\ \Psi_{SAF} \end{bmatrix}^{(\ell)} + \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{N} \end{bmatrix} \underline{\boldsymbol{L}}^{-1} S. \quad (3.15)$$

Equation (3.15) is simply coded by appending the Ψ_{SAF} to the scalar flux unknowns (after a transport sweep, the operator $\boldsymbol{D}$ is applied to yield the newest scalar flux whereas the operator $\boldsymbol{N}$ is applied to update the significant angular flux). Note that when $\overline{\boldsymbol{L}} = 0$ (i.e., no dependencies in the sweep), we obtain the standard SI formula:

$$\Phi^{(l+1)} = \boldsymbol{D}\boldsymbol{L}^{-1}\left(\boldsymbol{M\Sigma}\Phi^{(l)} + S\right). \quad (3.16)$$

From the SI formula of equation (3.15), it follows that the linear system for a GMRES-based transport solves is simply:

$$\left(\boldsymbol{I} - \boldsymbol{T}\right) x = b, \quad (3.17)$$

with:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{D}\underline{\boldsymbol{L}}^{-1}\boldsymbol{M\Sigma} & \boldsymbol{D}\underline{\boldsymbol{L}}^{-1}\overline{\boldsymbol{L}}\boldsymbol{N}^{T} \\ \boldsymbol{N}\underline{\boldsymbol{L}}^{-1}\boldsymbol{M\Sigma} & \boldsymbol{N}\underline{\boldsymbol{L}}^{-1}\overline{\boldsymbol{L}}\boldsymbol{N}^{T} \end{bmatrix}, \tag{3.18}$$

$$x = \begin{bmatrix} \Phi \\ \Psi_{SAF} \end{bmatrix}, \tag{3.19}$$

$$b = \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{N} \end{bmatrix} \underline{\boldsymbol{L}}^{-1} S. \tag{3.20}$$

2 Synthetic Acceleration and Preconditioning

Ignoring graph dependencies for simplicity of the presentation, the transport equation is:

$$\left(\boldsymbol{L} - \boldsymbol{M\Sigma D}\right)\Psi = S. \tag{3.21}$$

It is often computationally effective to write the above linear system as:

$$\left(\boldsymbol{I} - \boldsymbol{L}^{-1}\boldsymbol{M\Sigma D}\right)\Psi = \boldsymbol{L}^{-1}S \tag{3.22}$$

because $\boldsymbol{L}$ is easier to invert than $(\boldsymbol{L} - \boldsymbol{M\Sigma D})$. Equation (3.22) is equation (3.21) with sweep preconditioning. Therefore, an iterated scheme (the SI technique) yields formally:

$$\Psi^{(\ell+1)} = \boldsymbol{L}^{-1}\left(\boldsymbol{M\Sigma D}\Psi^{(\ell)} + S\right). \tag{3.23}$$

The error equation is:

$$\begin{aligned} \Psi - \Psi^{(\ell+1)} &= \boldsymbol{L}^{-1}\boldsymbol{M\Sigma D}\left(\Psi - \Psi^{(\ell)}\right) \\ &= \boldsymbol{L}^{-1}\boldsymbol{M\Sigma D}\left(\Psi - \Psi^{(\ell+1)} + \Psi^{(\ell+1)} - \Psi^{(\ell)}\right), \end{aligned} \tag{3.24}$$

that is, the transport equation satisfied by the angular error $\epsilon^{(\ell+1)} = \Psi - \Psi^{(\ell+1)}$ is:

$$\left(\boldsymbol{L} - \boldsymbol{M\Sigma D}\right)\epsilon^{(\ell+1)} = \boldsymbol{M\Sigma D}\left(\Psi^{(\ell+1)} - \Psi^{(\ell)}\right). \tag{3.25}$$

This equation is of the same form as equation (3.21) (where the source term is now the scattering due to the difference in successive flux iterates) and, therefore, is just as difficult to solve. However, solving it would provide the exact additive term required to obtain the exact solution:

$$\Psi = \Psi^{(\ell+1)} + \epsilon^{(\ell+1)}. \tag{3.26}$$

Since the diffusion error modes are not efficiently attenuated by the above SI process, it is natural to seek a low-order error equation. Taking the zeroth and first angular moment moment of equation (3.25), one obtains a diffusion equation for the scalar ε:

$$\boldsymbol{A}\epsilon^{(\ell+1)} = \Sigma\left(\Phi^{(\ell+1)} - \Phi^{(\ell)}\right), \tag{3.27}$$

where $\boldsymbol{A}$ is the diffusion operator. However, the scalar correction $\epsilon^{(\ell+1)}$, when added to the previous iterate of the scalar flux $\Phi^{(\ell+1)}$, will not yield the exact scalar flux solution because the low-order error equation is not strictly identical to the transport error equation. However, it is expected that significant speedup can be achieved in the iterative solution technique that can now be described as follows:

1. Perform a transport sweep and obtain the scalar flux after that sweep:

$$\Phi^{(\ell+1/2)} = \boldsymbol{DL}^{-1}\left(\boldsymbol{M\Sigma}\Phi^{(\ell)} + S\right). \tag{3.28}$$

2. Solve for the diffusion error equation corrective addition:

$$\boldsymbol{A}\epsilon^{(\ell+1/2)} = \boldsymbol{\Sigma}\left(\Phi^{(\ell+1/2)} - \Phi^{(\ell)}\right). \tag{3.29}$$

3. Obtain a new estimate of the scalar flux for the next transport sweep:

$$\Phi^{(\ell+1)} = \Phi^{(\ell+1/2)} + \epsilon^{(\ell+1/2)}. \tag{3.30}$$

When the process is recast in a Krylov (GMRES) solver, one obtains the following preconditioned GMRES solve:

$$\left(\boldsymbol{I} + \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\right)\left(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{\Sigma}\right)\Phi = \left(\boldsymbol{I} + \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\right)\boldsymbol{D}\boldsymbol{L}^{-1}S. \tag{3.31}$$

As seen in [115], DSA requires some spatial consistency to converge. Moreover, we also ignored the effect of anisotropic scattering. The discussion of these aspects is left for Chapter IV where DSA's ineffectiveness in such situations is discussed.

D Discontinuous Finite Element Discretization

1 DFEM and sweeps

Using equations (3.5) and (3.6), equation (3.1) can be written:

$$\left(\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla} + \Sigma_t(\mathbf{r})\right)\psi_d = q(\mathbf{r}) \tag{3.32}$$

$$q(\mathbf{r}) = \frac{1}{4\pi}\Sigma_s(\mathbf{r})\phi(\mathbf{r}) + \frac{1}{4\pi}S(\mathbf{r}). \tag{3.33}$$

$q(\mathbf{r})$ is a volumetric source. For anisotropic scattering, equation (3.32) would also include higher angular terms. During a sweep, equation (3.32) is inverted.

Next, the domain $\mathcal{D}$ is meshed into elements K, ψ_d is expanded on the basis function χ_i ($\psi_d = \sum_i \psi_{d,i}\chi_i$), equation (3.32) is multiplied by χ_j, and equation (3.32) is integrated over K:

$$\int_K \left(\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla} + \Sigma_t\right) \left(\sum_i \psi_{d,i}\chi_i\right) \chi_j \ d\mathbf{r} = \int_K \left(\sum_i q_i\chi_i\right) \chi_j \ d\mathbf{r}. \tag{3.34}$$

Applying Stokes' theorem, we obtain:

$$\begin{aligned}
&\oint_{\partial K} \boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b \left(\sum_i \psi_{d,i}\chi_i\right) \chi_j \ d\mathbf{r} - \int_K \left(\sum_i \psi_{d,i}\chi_i\right) \boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla}\chi_j \ d\mathbf{r} + \\
&\int_K \Sigma_t \left(\sum_i \psi_{d,i}\chi_i\right) \chi_j \ d\mathbf{r} = \int_K \left(\sum_i q_i\chi_i\right) \chi_j \ d\mathbf{r},
\end{aligned} \tag{3.35}$$

where $\oint_{\partial K}$ is the integral over the boundary ∂K and $\boldsymbol{n}_b$ is the exterior normal. Using upwind, equation (3.35) becomes:

$$\begin{aligned}
&-\int_K \left(\left(\sum_i \psi_{d,i}\chi_i\right) \boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla}\chi_j + \Sigma_t \left(\psi_{d,i}\chi_i\right) \chi_j\right) \mathbf{r} + \\
&\int_{\partial K^+} \boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b \left(\sum_i \psi_{d,i}\chi_i\right) \chi_j \ d\mathbf{r} = \int_K \left(\sum_i q_i\chi_i\right) \chi_j \ d\mathbf{r} + \\
&\int_{\partial K^-} \left|\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b\right| \left(\sum_i \psi_{d,i}^{\uparrow}\chi_i\right) \chi_j \ d\mathbf{r},
\end{aligned} \tag{3.36}$$

where ∂K^- is the inflow of element K ($\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b < 0$) and ∂K^+ is the outflow face of element face of element K ($\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b > 0$). The angular flux values on an inflow face, denoted by $\psi_d^{\uparrow}$ in equation (3.36), are taken from the upwind neighbor element of that face. We see that equation (3.36) can be inverted for only cell K as soon as $\psi_d^{\uparrow}$ is known, yielding the concept of transport sweep through the mesh.

2 BiLinear Discontinous finite elements

In this research, the BLD basis functions are used only on the rectangular cells. If the cells are arbitrary convex quadrilaterals, the discretization may not exist. The BLD basis functions defined on the following rectangular cell (see Figure III.1):

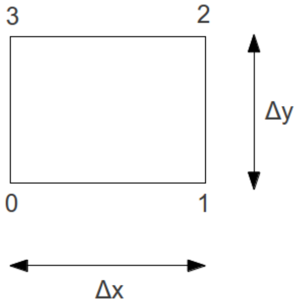


Figure III.1: Rectangular cell

are:

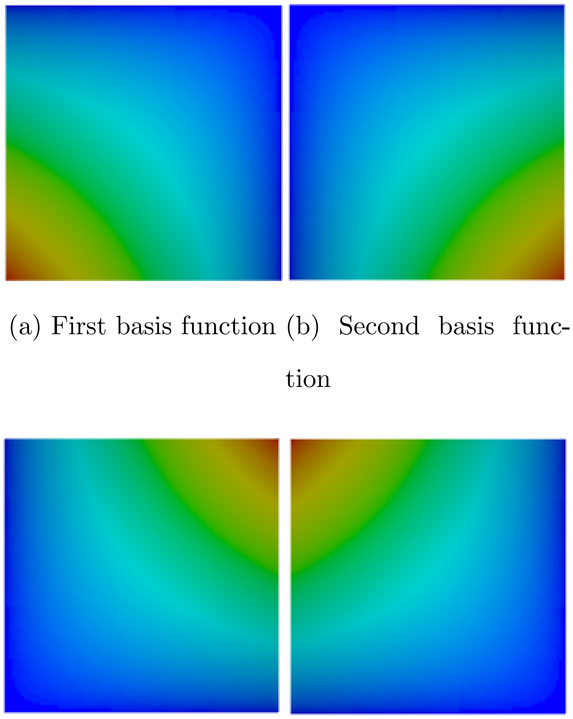
$$\chi_0(x,y) = \frac{(\Delta x - x)}{\Delta x}\frac{(\Delta y - y)}{\Delta y}, \tag{3.37}$$

$$\chi_1(x,y) = \frac{x}{\Delta x}\frac{(\Delta y - y)}{\Delta y}, \tag{3.38}$$

$$\chi_2(x,y) = \frac{x}{\Delta x}\frac{y}{\Delta y}, \tag{3.39}$$

$$\chi_3(x,y) = \frac{(\Delta x - x)}{\Delta x}\frac{y}{\Delta y}, \tag{3.40}$$

with $x \in [0, \Delta x]$ and $y \in [0, \Delta y]$. On a square cell, the basis functions are given in Figure (III.2):



(a) First basis function (b) Second basis function

(c) Third basis function (d) Fourth basis function

Figure III.2: BLD basis function

Given these basis functions, the matrices of equation (3.36) (1D and 2D mass matrix, $\boldsymbol{M}_{ij} = \int_K \chi_i \chi_j \, d\mathbf{r}$ and the “gradient” matrix, $\boldsymbol{G}_{ij} = \int_K \chi_i \boldsymbol{\nabla} \chi_j \, d\mathbf{r}$) can be easily analytically computed on rectangular cells. On “almost” rectangular cells, the integrals have to be computed analytically. On highly distorted cells, these integrals become singular.

3 PieceWise Linear Discontinuous finite elements

Next, we introduce the PieceWise Linear Discontinuous finite elements developed in [25, 95]. The interest of PWLD finite elements is that they can be used on arbitrary polygons. We will see in Chapter V the advantages of using arbitrary polygons instead of triangles or quadrilaterals. To obtain the PWLD basis functions on two-dimensional polygons, we need to introduce the within-cell point c. The coordinates of c are weighted averages of the vertex coordinates:

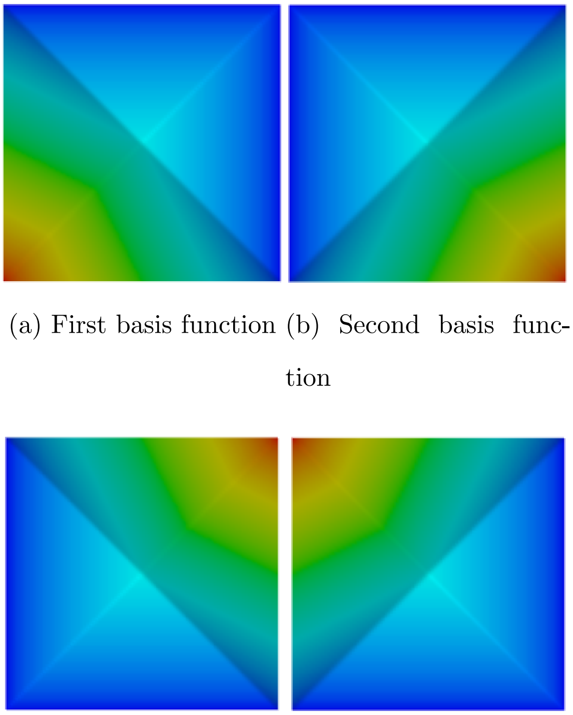
$$x_c = \sum_{j=1}^{N_V} \alpha_j x_j, \tag{3.41}$$

$$y_c = \sum_{j=1}^{N_V} \alpha_j y_j, \tag{3.42}$$

where $\sum_{j=1}^{N_V} \alpha_j = 1$, $\alpha_j \geq 0 \; \forall j$, and N_V is the number of vertices of the cell. The basis function at vertex j is defined by [95]:

$$\chi_j(x, y) = t_j(x, y) + \alpha_j t_c(x, y), \tag{3.43}$$

where the t_j function is the linear functions such that $t_j(x, y)$ is unity at vertex j and zero at the $j - 1$, $j + 1$, and c. The function $t_c(x, y)$ is unity at c and zero at each vertex. In this work, the arbitrary positive weights α_j are chosen to be $\frac{1}{N_V}$. On a square cell with $\alpha_j = \frac{1}{4} \; \forall j$, the basis functions are given in Figure (III.3):



(a) First basis function (b) Second basis function

(c) Third basis function (d) Fourth basis function

Figure III.3: PWLD basis function

On triangular cells, the PWLD basis functions reduces to the standard Linear Discontinuous (LD) basis functions if $\alpha_j = \frac{1}{3}$.

Given the definition of the PWLD finite elements, it may seem complicated to build the mass matrix, $\boldsymbol{M}$, or the gradient matrix, $\boldsymbol{G}$, on an arbitrary polygonal cells. The construction of such matrices can be greatly simplified using "side" sub-cells. A "side" sub-cell is a triangular cell made from two adjacent vertices and the point c. On each "side" sub-cells, the mass matrix, for example, can be build using LD finite elements. To do so, we first need to rewrite the mass matrix $\boldsymbol{M}$:

$$\boldsymbol{M} = \sum_{k=1}^{N_V} \int_{S_k} \chi_i \chi_j \, d\mathbf{r}, \tag{3.44}$$

where S_k are the "side" sub-cells (see Figure III.4). We see that $\boldsymbol{M}$ can be built by looping over all the "side" sub-cells. For a given "side" sub-cell S_k, we have:

$$\int_{S_k} \chi_i \chi_j \, d\mathbf{r} = \int_{S_k} \left(t_i t_j + \alpha t_i t_c + \alpha t_c t_i + \alpha^2 t_c^2\right) d\mathbf{r} \tag{3.45}$$

On S_k, the basis functions t_i, t_j, and t_c, are identical to the LD basis functions. Therefore, if we note $\boldsymbol{M}_{S_k}$, the mass matrix on the "side" sub-cell formed by the vertices 0, 1, and c:

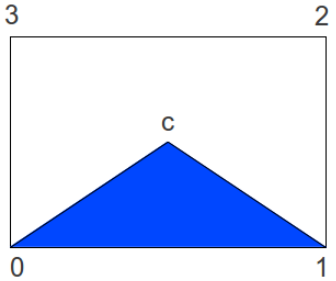


Figure III.4: Sub-cell (in blue) in the cell

$\boldsymbol{M}$ can be built using $\boldsymbol{M}_{S_k}$:

$$\boldsymbol{M}(0,0) = \boldsymbol{M}_{S_k}(0,0) + \alpha\boldsymbol{M}_{S_k}(0,2) + \alpha\boldsymbol{M}_{S_k}(2,0) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.46}$$

$$\boldsymbol{M}(0,1) = \boldsymbol{M}_{S_k}(0,1) + \alpha\boldsymbol{M}_{S_k}(0,2) + \alpha\boldsymbol{M}_{S_k}(2,1) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.47}$$

$$\boldsymbol{M}(0,2) = \alpha\boldsymbol{M}_{S_k}(0,2) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.48}$$

$$\boldsymbol{M}(0,3) = \alpha\boldsymbol{M}_{S_k}(0,2) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.49}$$

$$\boldsymbol{M}(1,0) = \boldsymbol{M}_{S_k}(1,0) + \alpha\boldsymbol{M}_{S_k}(1,2) + \alpha\boldsymbol{M}_{S_k}(2,0) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.50}$$

$$\boldsymbol{M}(1,1) = \boldsymbol{M}_{S_k}(1,1) + \alpha\boldsymbol{M}_{S_k}(1,2) + \alpha\boldsymbol{M}_{S_k}(2,1) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.51}$$

$$\boldsymbol{M}(1,2) = \alpha\boldsymbol{M}_{S_k}(1,2) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.52}$$

$$\boldsymbol{M}(1,3) = \alpha\boldsymbol{M}_{S_k}(1,2) + \alpha^2\boldsymbol{M}_{S_k}(2,2) \tag{3.53}$$

$$\boldsymbol{M}(2,0) = \alpha \boldsymbol{M}_{S_k}(2,0) + \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.54}$$

$$\boldsymbol{M}(2,1) = \alpha \boldsymbol{M}_{S_k}(2,1) + \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.55}$$

$$\boldsymbol{M}(2,2) = \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.56}$$

$$\boldsymbol{M}(2,3) = \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.57}$$

$$\boldsymbol{M}(3,0) = \alpha \boldsymbol{M}_{S_k}(2,0) + \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.58}$$

$$\boldsymbol{M}(3,1) = \alpha \boldsymbol{M}_{S_k}(2,1) + \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.59}$$

$$\boldsymbol{M}(3,2) = \alpha^2 \boldsymbol{M}_{S_k}(2,2) \tag{3.60}$$

$$\boldsymbol{M}(3,3) = \alpha^2 \boldsymbol{M}_{S_k}(2,2). \tag{3.61}$$

To finish building $\boldsymbol{M}$, we need to loop over all of the "side" sub-cells, S_k ($k = 1, \ldots, N_v$), of the cell. The gradient matrix is built similarly.

E Conclusions

In this section, we explained how Source Iteration, Krylov solvers, and Diffusion Synthetic Acceleration can be used to solve the transport equation. The two spatial discretizations, BLD and PWLD finite elements, that we will employ in the next chapters have been presented.

CHAPTER IV

ANGULAR MULTIGRID PRECONDITIONER FOR S_N EQUATIONS WITH HIGHLY FORWARD-PEAKED SCATTERING KERNEL

A Introduction

The discrete ordinates method has been shown to be quite accurate for electron and coupled electron-photon transport [64, 69, 113], which is required in the development of radiation therapy protocols, satellite electronics shielding, flash x-ray machine design, and a wide variety of other applications. Charged particles interact through Coulomb interactions with the background medium. Such interactions predominately result in extremely small changes in particle direction and energy. These interactions are well characterized by the Fokker-Planck limit of the transport equation [37, 72]. In this limit, the directional and energy changes are decoupled with the former modeled by the continuous scattering operator and the latter modeled by the continuous slowing-down operator. In this Chapter, we consider the discrete-ordinate (S_n) angular discretization of the transport equation with a focus upon iterative solution methods for problems with highly forward-peaked scattering characteristic of the Fokker-Planck limit.

When the scattering is highly forward-peaked, solving the S_n transport equation can be challenging due to the slow convergence of the standard iterative algorithm, Source Iteration (SI). To speed up iterative convergence, acceleration schemes such as Diffusion Synthetic Acceleration (DSA) are used. With isotropic or weakly anisotropic scattering, DSA is generally highly effective [19]. This occurs because the quickly varying error modes are strongly attenuated by the transport sweep, whereas the diffusion operator attenuates the slowly varying error modes. However, DSA can be completely ineffective in the Fokker-Planck limit [71] because the diffusion operator does not attenuate all the slowly varying error modes.

To address this deficiency, an angular multigrid method for the one-dimensional S_n equations was developed by Morel and Manteuffel (MM) [71]. This method was extremely efficient yielding a maximum spectral radius for a model infinite-medium problem of 0.6 at a cost of approximatively twice that of DSA. This maximum spectral radius is approached in the Fokker-Planck limit whereas in the same limit, the spectral radius of DSA approaches one.

Pautz, Adams, and Morel (PAM) [84] generalized the MM method to 2-D, but it was found to be stable only for weakly forward-peaked scattering. The instability arose from high-frequency spatial error amplification that occurred in the transfer of error estimates between angular grids (a sequence of different S_n orders). Stabilization was achieved by filtering the error estimates via diffusion operators. However, this filtering was expensive and significantly degraded the effectiveness of the method such that the spectral radius approaches one in the Fokker-Planck limit. Nonetheless, the method was always more efficient than the DSA method for the test problems considered.

In this Chapter, the PAM method with no filtering (PAMNF) is recast as a preconditioner and used in conjunction with the GMRES Krylov method. In this form, stability of the iteration scheme is guaranteed. Krylov subspace methods have been developed to solve large sparse linear systems. Their application to the transport equation has been extensively studied in the past [45, 79, 83, 116] where the importance of preconditioning was highlighted. In [79], the authors used successfully the 1D MM angular method as a preconditioner for GMRES and CGS. These preconditioned Krylov methods were significantly faster than MM. In this research, we compute the eigenvalues of preconditioned system for a model problem and compare the spectrum with those of preconditioners based upon transport sweeps and DSA. It is found that relative to these preconditioners, PAMNF preconditioning moves the eigenvalues away from zero while leaving them constrained to a reasonably small portion of the complex plane. These are desirable properties for a preconditioner because the convergence rate of GMRES is proportional to the size of the eigenvalue cluster and/or the distance between the clusters [39, 116]. The eigenvalues close to zero slow down the convergence of GMRES because they can be viewed as single values that are processed one at the time [39, 116]. We also compare the convergence rates and efficiency of these preconditioners for various test problems with forward-peaked scattering. We find that PAMNF preconditioning is significantly more efficient than DSA preconditioning, and becomes increasingly so as the Fokker-Planck limit is approached. However, unlike the MM method for one-dimensional geometries, the number of iterations required for convergence nonetheless increases as this limit is approached. In spite of this fact, the PAMNF-preconditioned Krylov method achieves good efficiency without the costly filtering associated with the original PAM fixed-point iteration scheme, and appears to be more effective than other existing algorithms for solving the S_n equations with highly forward-peaked scattering.

One key feature of the angular multigrid method is that transport sweeps can strongly damp the high frequency error modes (upper half of the flux moments) with the use of an “optimal” transport correction [71]. This “optimal” transport correction is a variant of the well-known extended transport correction [63, 67].

B Iterative schemes for highly forward-peaked scattering

1 Source Iteration and DSA

Equation (1.17) can be solved using the Source Iteration method, which can be interpreted a Richardson iteration, or a Krylov method. The Source Iteration method at the k^{th} iteration is given by:

$$\Phi^{(k+1)} = \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{\Sigma}\Phi^{(k)} + \boldsymbol{D}\boldsymbol{L}^{-1}Q. \tag{4.1}$$

When the scattering ratio $c = \max_l \left(\frac{\Sigma_{s,l}}{\Sigma_t}\right)$ is close to one, the spectral radius of SI can become arbitrary close to one and the convergence becomes arbitrary slow. The SI+DSA scheme is given by a transport sweep:

$$\Phi^{(k+1/2)} = \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{\Sigma}\Phi^{(k)} + \boldsymbol{D}\boldsymbol{L}^{-1}Q, \tag{4.2}$$

followed by a diffusion synthetic acceleration for the correction:

$$\delta\Phi^{(k)} = \boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\to 0}\left(\Phi^{(k+1/2)} - \Phi^{(k)}\right), \tag{4.3}$$

yielding the next iterate for the flux moments:

$$\Phi^{(k+1)} = \phi^{(k+1/2)} + \boldsymbol{P}_{0/1\to n}\delta\Phi^{(k)}. \tag{4.4}$$

Finally, using equations (4.2) to (4.4), we obtain:

$$\begin{aligned}\Phi^{(k+1)} = &\left((\boldsymbol{I} + \boldsymbol{P}_{0/1\to n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\to 0}\boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{\Sigma} - \boldsymbol{P}_{0/1\to n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\to 0}\right)\Phi^{(k)} \\ &+ (\boldsymbol{I} + \boldsymbol{P}_{0/1\to n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\to 0})\boldsymbol{D}\boldsymbol{L}^{-1}Q.\end{aligned} \tag{4.5}$$

where $\boldsymbol{\mathcal{T}}_0$ is the matrix associated with the DSA operator, $\boldsymbol{R}_{n\rightarrow 0}$ is the restriction matrix of Φ_n (all moments) to Φ_0 (only zeroth moment) and $\boldsymbol{P}_{0/1\rightarrow n}$ the projection matrix of Φ_0 or Φ_1, depending on whether only the zeroth or the zeroth and the first moment are accelerated, onto Φ_n. When only the zeroth moment is accelerated, the scheme is always stable and the spectral radius is max $\left(\rho_{iso}, \frac{\Sigma_{s,1}}{\Sigma_t}\right)$ where ρ_{iso} is the spectral radius when the scattering is isotropic. In multidimensional geometry, when both the zeroth and the first moments are accelerated, the scheme is not always stable and the spectral radius is given by $\left(\rho_{iso}, \frac{\Sigma_{s,1}}{\Sigma_t-\Sigma_{s,1}}\right)$ [20]. For highly forward peaked scattering, accelerating the zeroth moment is ineffective $\left(\frac{\Sigma_{s,1}}{\Sigma_t} \rightarrow 1\right)$, whereas accelerating both moments can be unstable $\left(\frac{\Sigma_{s,1}}{\Sigma_t-\Sigma_{s,1}} > 1\right)$.

C Review of previous angular multigrid work

1 One dimensional geometry: the Morel and Manteuffel (MM) method

As mentioned previously, only the zeroth and the first flux moments can be accelerated with DSA. To accelerate higher moments, other methods have to be used. Morel and Manteuffel proposed an angular multigrid method to accelerate the SI calculation of the one-dimension S_n equations with highly anisotropic scattering [71]. They used a variation of the extended transport correction [63] to attenuate the "upper half" of the flux moments (higher angular frequencies) via transport sweeps. The "lower half" of the flux moments (lower angular frequencies) is accelerated using a sweep of the $S_{n/2}$ equations. These $S_{n/2}$ equations are themselves accelerated using $S_{n/4}$ equations. The order of the transport operator is sequentially divided by two until the S_4 level is reached. At this point, the P_1 equations are used to accelerate the S_4 equations. For the general case where $n/2^i$ is odd, we need to define:

$$Half(n) = \begin{cases} \frac{n}{2}, & \text{if } \frac{n}{2} \text{ is even} \\ \frac{n}{2} + 1, & \text{if } \frac{n}{2} \text{ is odd} \end{cases} \tag{4.6}$$

Using this definition of "Half" to coarsen the angular grid, the sequence of sweeps for an S_{16} base level is $(S_{16} - S_8 - S_4)$ and for a S_{18} base level, the sequence is $(S_{18} - S_{10} - S_6 - S_4)$. Morel and Manteuffel's scheme works as follows:

1. Perform a transport sweep for the S_n equations.

2. Perform a transport sweep for the S_{n_2} equations with a P_{n_2-1} expansions using the S_n residual as the inhomogeneous source, where $n_2 = Half(n)$.

3. Continue coarsening the angular grid by a factor two (i.e., according to the definition of "Half") until a sweep has been performed for the S_4 equations.

4. Solve the P_1 equations ($P1$ synthetic acceleration, $P1SA$) with a P_1 expansion of the S_4 residual as the inhomogeneous source.

5. Add the Legendre moments of the P_1 solution to the Legendre moments of the S_4 iterate to obtain the accelerated S_4 iterate.

6. Continue to add the corrections from each coarse grid to the finer grid above to obtain the accelerated S_n moments.

Every time a transport sweep is performed, the optimal transport correction needs to be used [71]. For a P_{n-1} expansion of the cross sections, the corrected cross sections are given by:

$$\Sigma^*_{s,j} = \Sigma_{s,j} - \frac{\Sigma_{s,n/2} + \Sigma_{s,n-1}}{2} \text{with } j = \{t\} \text{ or } \{s, l\}. \tag{4.7}$$

This correction is said to be optimal because for an infinite homogeneous medium, it minimizes the "high-frequency" angular errors. The smoothing factor is given by:

$$\rho_s = \max\left(|\Sigma_{s,n/2}|/\Sigma_{s,0}, |\Sigma_{s,n/2+1}|/\Sigma_{s,0}, \ldots, |\Sigma_{s,n-1}|/\Sigma_{s,0}\right). \tag{4.8}$$

To compare the effectiveness of the angular multigrid method with DSA, Fokker-Planck scattering cross sections (equation (2.95)) can be used. In one dimensional geometry, DSA becomes less efficient as $\Sigma_{s,l}$ $(0 < l \leq L)$ becomes closer to $\Sigma_{s,0}$. Therefore, in the limit as $L \to \infty$, DSA no longer accelerates the convergence of SI for Fokker-Planck scattering (the spectral radius tends to 1.0). However, the spectral radius of the angular multigrid method has an upper bound of 0.6 when $L \to \infty$. It can be easily shown by using equation (4.8) and the fact that for Fokker-Planck scattering cross sections the cross-section moments decrease monotonically:

$$\rho_s = \frac{\Sigma^*_{s,N/2}}{\Sigma_{s,0}} = \frac{3N-6}{5N-6}, \tag{4.9}$$

which tends to 0.6 when N goes to infinity.

The MM method converges in less iterations than DSA but it is important to look at the cost of each MM iterations: one sweep in each N directions + one DSA iteration + one sweep in each $(\frac{N}{2} + \frac{N}{4} + \ldots)$ directions. Since $\frac{N}{2} + \frac{N}{4} + \ldots \leq N$, the cost of one MM iteration is less than: two sweeps in each N directions + one DSA iteration.

2 Multidimensional geometry: the Pautz-Adams-Morel (PAM) methods

In the multidimensional case, DSA becomes unstable when both the zeroth and the first flux moments are accelerated and $\frac{\Sigma_{s,1}}{\Sigma_t} \geq 0.5$, [20]. In [84], the authors modified the one dimensional angular multigrid method by accelerating only the zeroth flux moment with the DSA and by using S_2 as lowest transport sweep instead of S_4. Even so, the proposed method (PAMNF, with "NF" for no-filtering) was unstable and a filter was needed to stabilize the scheme (PAMF). Therefore, the angular multigrid method was modified as follows [84]:

1. Perform a transport sweep for the S_n equations.

2. Perform a transport sweep for the S_{n_2} equations with a P_{n_2} for 2-D problem and a P_{n_2+1} for 3-D problem expansion for the S_n residual as the inhomogeneous source, where $n_2 = Half(n)$.

3. Continue coarsening the angular grid by a factor two (i.e., according to the definition of "Half") until a sweep has been performed for the S_2 equations.

4. Solve the diffusion equation with a P_0 expansion for the S_2 residual as the inhomogeneous source.

5. Apply a diffusive filter to the corrections from steps 2 and 3 (without this, the method is unstable).

6. Add the corrections from steps 4 and 5 to the Legendre moments of the S_n iterate to obtain the accelerated S_n moments.

The filter stabilizes the method which otherwise would diverge. Without the filtering process, the low frequency modes are well attenuated but instabilities are introduced in higher frequency modes. Filtering eliminates the high frequency corrections which are well attenuated by SI alone but it keeps the low frequency corrections. The filter is given by:

$$\left(-\boldsymbol{\nabla}\cdot\frac{\beta_f}{3\Sigma_f}\boldsymbol{\nabla}+\Sigma_f\right)f_{corr}=\Sigma_f\left(\Phi_{n_2}+P_{n_4\rightarrow n_2}\Phi_{n_4}+\ldots+P_{2\rightarrow n_2}\Phi_2\right), \tag{4.10}$$

where Σ_f is the filter cross section and β_f is the filter tuning parameter. A Fourier analysis shows that given an input amplitude A, the "diffusively filtered" amplitude is:

$$F = \frac{A}{1 + \frac{\beta_f \lambda^2}{3\Sigma_f}}. \tag{4.11}$$

It is clear that the modes with large $|\lambda|$ (high frequencies) are strongly attenuated while low-frequency modes are not. However, the filtering process does not prevent the spectral radius from becoming arbitrary close to one when L becomes large [84].

D Angular multigrid as preconditioner for Krylov Solvers

In this research, we propose to abandon SI as the solver for the S_n equations with highly-forward peaked scattering and to use a Krylov solver instead. The DSA-preconditioned system of equations solved with a Krylov method is:

$$\begin{multline}\left((\boldsymbol{I} + \boldsymbol{P}_{0/1\rightarrow n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\rightarrow 0})(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{\Sigma})\right)\phi = \\ (\boldsymbol{I} + \boldsymbol{P}_{0/1\rightarrow n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{n\rightarrow 0})\boldsymbol{D}\boldsymbol{L}^{-1}Q. \quad (4.12)\end{multline}$$

The angular multigrid scheme can also be recast to be used by a Krylov solver. Here, we have chosen to recast the PAM method without filtering (PAMNF) as a preconditioner for a Krylov solver. The successive corrections of the angular multigrid acceleration form different stages of a preconditioner used in the Krylov solver. Two variations of the PAMNF preconditioner will be tested:

- the coarsest level is DSA (ANMG-DSA) (with the coarsest S_n level being S_2).
- the coarsest level is $P1SA$ (ANMG-P1SA) (with the coarsest S_n level being S_4).

First, we present the angular multigrid using DSA and then, the angular multigrid using $P1SA$. Later, these two versions are compared.

1 ANMG-DSA

Using a method similar to the one we used to write the equation for the preconditioned Krylov solver, we recast the PAMNF for SI as a preconditioner for a Krylov solver. First, we write the SI sweep equation, the successive corrections and the new iterate built from the sweep values plus all the successive corrections:

$$\Phi_n^{(k+1/2)} = \boldsymbol{D}_n\boldsymbol{L}_n^{-1}\boldsymbol{M}_n\boldsymbol{\Sigma}_n\Phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q \tag{4.13}$$

$$\delta\Phi_{n_2}^{(k)} = \boldsymbol{D}_n\boldsymbol{L}_{n_2}^{-1}\boldsymbol{M}_{n_2}\boldsymbol{\Sigma}_{n_2}\boldsymbol{R}_{n\to n_2}\left(\Phi_n^{(k+1/2)} - \Phi_n^{(k)}\right) \tag{4.14}$$

$$\ldots \tag{4.15}$$

$$\delta\Phi_2^{(k)} = \boldsymbol{D}_2\boldsymbol{L}_2^{-1}\boldsymbol{M}_2\boldsymbol{\Sigma}_2\boldsymbol{R}_{4\to 2}\delta\Phi_4 \tag{4.16}$$

$$\delta\Phi_0^{(k)} = \boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{2\to 0}\delta\Phi_2^{(k)} \tag{4.17}$$

$$\Phi_n^{(k+1)} = \Phi_n^{(k+1/2)} + \boldsymbol{P}_{n_2\to n}\delta\Phi_{n_2}^{(k)} + \ldots + \boldsymbol{P}_{2\to n}\delta\Phi_2^{(k)} + \boldsymbol{P}_{0\to n}\delta\Phi_0^{(k)}. \tag{4.18}$$

Now, all the corrections $\delta\Phi_0^{(k)}$ through $\delta\Phi_{n_2}^{k}$ are substituted into the moment update equation, equation (4.18), yielding:

$$\begin{aligned}\Phi_n^{(k+1)} =& \boldsymbol{T}_n\Phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q + \boldsymbol{P}_{n_2\to n}\left(\boldsymbol{T}_{n_2}\boldsymbol{R}_{n\to n_2}\left(\Phi_n^{(k+1/2)} - \Phi_n^{(k)}\right)\right) + \ldots \\ &+ \boldsymbol{P}_{2\to n}\boldsymbol{T}_2\boldsymbol{R}_{4\to 2}\delta\Phi_4^{(k)} + \boldsymbol{P}_{0\to n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{2\to 0}\delta\Phi_2^{(k)} \\ =& \boldsymbol{T}_n\Phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q + \boldsymbol{P}_{n_2\to n}\left(\boldsymbol{T}_{n_2}\boldsymbol{R}_{n\to n_2}\left(\boldsymbol{T}_n\Phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q\right.\right. \\ &\left.\left.-\Phi_n^{(k)}\right)\right) + \ldots + \boldsymbol{P}_{2\to n}\boldsymbol{T}_2\boldsymbol{R}_{4\to 2}\left(\boldsymbol{T}_4\boldsymbol{R}_{8\to 4}\left(\ldots\left(\boldsymbol{T}_n\phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q\right.\right.\right. \\ &\left.\left.\left.-\Phi_n^{(k)}\right)\right)\right) + \boldsymbol{P}_{0\to n}\boldsymbol{\mathcal{T}}_0^{-1}\boldsymbol{R}_{2\to 0}\left(\boldsymbol{T}_2\boldsymbol{R}_{4\to 2}\left(\ldots\left(\boldsymbol{T}_N\Phi_n^{(k)} + \boldsymbol{D}_n\boldsymbol{L}_n^{-1}Q\right.\right.\right. \\ &\left.\left.\left.-\Phi_n^{(k)}\right)\right)\right),\end{aligned} \tag{4.19}$$

where we defined $\boldsymbol{T}_n = \boldsymbol{D}_n \boldsymbol{L}_n^{-1} \boldsymbol{M}_n \boldsymbol{\Sigma}_n$ (the subscript n denotes the S_n level). Thus, we obtain:

$$\begin{aligned}
\Phi_n^{(k+1)} = & \left(\boldsymbol{T}_n + \boldsymbol{P}_{n_2 \to n} \boldsymbol{T}_{n_2} \boldsymbol{R}_{n \to n_2} (\boldsymbol{T}_n - \boldsymbol{I}) + \ldots + \boldsymbol{P}_{2 \to n} \boldsymbol{T}_2 \boldsymbol{R}_{4 \to 2} \right. \\
& (\boldsymbol{T}_4 \boldsymbol{R}_{8 \to 4} (\ldots (\boldsymbol{T}_n - \boldsymbol{I}))) + \boldsymbol{P}_{0 \to n} \boldsymbol{\mathcal{T}}_0^{-1} \boldsymbol{R}_{2 \to 0} \boldsymbol{R}_{2 \to 0} \\
& \left. (\boldsymbol{T}_2 \boldsymbol{R}_{4 \to 2} (\ldots (\boldsymbol{T}_n - \boldsymbol{I}))) \right) \Phi_n^{(k)} + (\boldsymbol{I} + \boldsymbol{P}_{n_2 \to n} \boldsymbol{T}_{n_2} \boldsymbol{R}_{n \to n_2} + \ldots + \\
& \boldsymbol{P}_{2 \to n} \boldsymbol{T}_2 \boldsymbol{R}_{4 \to 2} (\boldsymbol{T}_4 \boldsymbol{R}_{8 \to 4} (\ldots (\boldsymbol{T}_{n_2} \boldsymbol{R}_{n \to n_2}))) + \\
& \boldsymbol{P}_{0 \to n} \boldsymbol{\mathcal{T}}_0^{-1} \boldsymbol{R}_{2 \to 0} (\boldsymbol{T}_2 \boldsymbol{R}_{4 \to 2} (\ldots (\boldsymbol{T}_{n_2} \boldsymbol{R}_{n \to n_2})))\big) \boldsymbol{D}_n \boldsymbol{L}^{-1} Q.
\end{aligned} \tag{4.20}$$

Finally, the linear system to be solved is given by:

$$(\boldsymbol{I} - \boldsymbol{T}_n) \boldsymbol{P}^{MG/DSA} \xi_n = \boldsymbol{D}_n \boldsymbol{L}_n^{-1} Q, \tag{4.21}$$

$$\boldsymbol{P}^{MG/DSA} \Phi_n = \xi_n, \tag{4.22}$$

where the multigrid preconditioner $\boldsymbol{P}^{MG/DSA}$ is:

$$\begin{aligned}
\boldsymbol{P}^{MG/DSA} = & \big(\boldsymbol{I} + \boldsymbol{P}_{n_2 \to n} \boldsymbol{T}_{n_2} \big(\boldsymbol{I} + \boldsymbol{P}_{n_4 \to n_2} \boldsymbol{T}_{n_4} \big(\ldots \big(\boldsymbol{I} + \boldsymbol{P}_{0 \to 2} \boldsymbol{\mathcal{T}}_0^{-1} \boldsymbol{R}_{2 \to 0}\big) \\
& \ldots\big) \boldsymbol{R}_{n_2 \to n_4}\big) \boldsymbol{R}_{n \to n_2}\big).
\end{aligned} \tag{4.23}$$

At this point, it is necessary to choose a DSA for implementation. Various DSA schemes have been reviewed in [19, 20, 21, 110, 111, 115]. We have chosen to employ the Modified Interior Penalty (MIP) DSA scheme developed by Wang and Ragusa [105] (see next Chapter). The MIP-DSA scheme is based on a discretization of the diffusion equation rather than the $P1$ equations. More specifically, MIP uses a bilinear *discontinuous* trial space, which is the same trial space as the one used for the S_n transport equations. However, the MIP equations are not fully consistent with the bilinear-discontinuous spatial discretization of the transport equation. Full consistency requires discretization of the $P1$ equations. The consistency discretized $P1$ equations are of a non-symmetric mixed form. The MIP-based DSA algorithm is always stable for isotropic scattering and the MIP diffusion matrix is symmetric positive definite (SPD), which makes it much easier to invert than the mixed $P1SA$ equation. For instance, one can use a conjugate gradient technique, preconditioned with SSOR to solve the MIP equation.

2 ANMG-P1SA

Using S_4 as the lowest S_n order followed by a $P1SA$ acceleration (instead S_2 followed by DSA) in equations (4.21) and (4.22) yields the following linear system:

$$\left(\boldsymbol{I} - \boldsymbol{T}_n\right) \boldsymbol{P}^{MG/P1SA} \xi_n = \boldsymbol{D}_n \boldsymbol{L}_n^{-1} Q \tag{4.24a}$$

$$\boldsymbol{P}^{MG/P1SA} \phi_n = \xi_n \tag{4.24b}$$

where the multigrid preconditioner $P^{MG/P1SA}$ is now given by:

$$P^{MG/P1SA} = \big(\boldsymbol{I} + \boldsymbol{P}_{n_2 \to n} \boldsymbol{T}_{n_2} \big(\boldsymbol{I} + \boldsymbol{P}_{n_4 \to n_2} \boldsymbol{T}_{n_4} \big(\ldots \big(\boldsymbol{I} + \boldsymbol{P}_{1 \to 4} \boldsymbol{\mathcal{T}}_1^{-1} \boldsymbol{R}_{4 \to 1}\big) \ldots\big) \boldsymbol{R}_{n_2 \to n_4}\big) \boldsymbol{R}_{n \to n_2}\big), \tag{4.25}$$

where $\boldsymbol{\mathcal{T}}_1$ is the matrix associated to the $P1SA$ operator. The $P1SA$ discretization used here is the $P1C$ method, defined in [104, 106]. This $P1SA$ preconditioner is positive definite (PD), but not symmetric. In principle, the analytic $P1$ equations can be put in a second-order diffusion form and discretized using the MIP approach. However, the first moment of the angular flux will be treated with less accuracy than the zeroth moment, which is undesirable.

E Eigenspectrum comparisons

In this section, we compare the eigenvalue spectrum for a given model problem. This is instructive because the convergence of GMRES is proportional to the relative radii of the eigenvalue clusters and/or the maximal distance between two clusters; furthermore, the eigenvalues close to zero are considered as outliers that are processed one at a time and increase the asymptotic error constant [39]. We use a S_8 Gauss-Legendre-Chebyshev Galerkin triangular quadrature. The domain, a $5cm-$side square uniformly discretized using by 25 cells, is homogeneous. Fokker-Planck cross sections, with $\alpha = 1$ and $L = 8$, are employed. For the spatial discretization, BiLinear Discontinuous (BLD) finite elements are used (see next Chapter). Σ_t is chosen to be equal to $\Sigma_{s,0}$. Figs. IV.1-IV.3 show the eigenvalue spectrum for sweep preconditioning (Fig. IV.1), DSA preconditioning (Fig. IV.2), and angular multigrid preconditioning (Fig. IV.3). The eigenvalues were obtained using implicit QR decomposition [47]. Even though the global matrices are never formed in transport solution techniques, we constructed them here for the purposes of the eigenspectrum analysis (specifically, the j^{th} column of any matrix A is obtained by multiplying it by the canonical basis vector e_j).

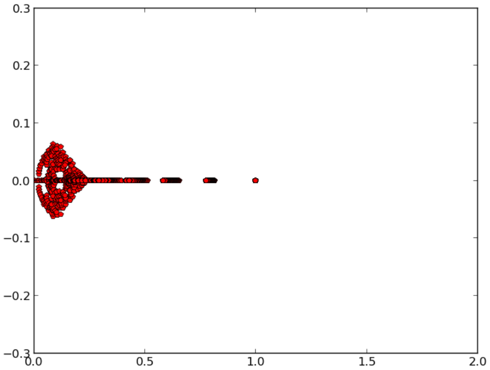


Figure IV.1: Eigenspectrum of the sweep preconditioned system

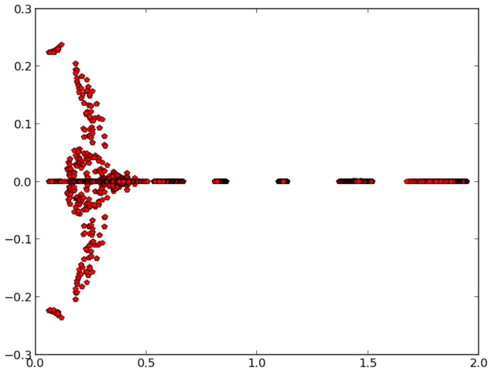


Figure IV.2: Eigenspectrum of the DSA preconditioned system

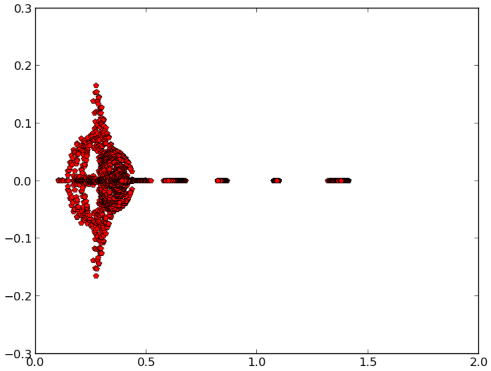


Figure IV.3: Eigenspectrum of the ANMG (DSA variant) preconditioned system

On these figures, we can note that sweep preconditioning is not effective as many eigenvalues are located near zero. DSA moves the eigenvalues away from zero. This explains the faster convergence of GMRES with DSA preconditioning compared to sweep preconditioning. ANMG moves the eigenvalues even further aways from zero than DSA and clusters them more compared to DSA. It is obvious from these figures that ANMG preconditioning should converge much faster than DSA preconditioning. From Figures IV.2 and IV.3, it may seem that a preconditioned (DSA or unfiltered ANMG) SI approach would be convergent (the eigenvalues are in $]0,2[$ and, thus, the spectral radius of the SI schemes would be less than one). In [86], it was shown that the unfiltered angular multigrid was unstable if the size of the cell was small enough (less than 0.1 transport mean-free-path) which was not the case here. Moreover, because of leakage in this test, convergence is improved. In the next section, we will further verify that wrapping a Krylov solver around the unfiltered ANMG results in an efficient scheme.

F Results

In this section, we first compare the number of GMRES iterations needed by ANMG-DSA and ANMG-$P1$SA to solve a model problem. Then, both the number of GMRES iterations and the elapsed time are compared for three methods:

- Sweep preconditioning (S).
- DSA preconditioning (DSA).
- Angular multigrid (DSA variant) preconditioning (ANMG-DSA).

For every test in this section except the first one, BLD finite elements are used and GMRES is restarted every 30 iterations. For the comparison between ANMG-DSA and ANMG-$P1$SA GMRES is restarted every 20 iterations.

1 Comparison between ANMG-DSA and ANMG-$P1$SA

The test uses a $5cm$ square domain, uniformly discretized using 50×50 cells. The homogeneous medium is homogeneous with a uniform isotropic source of intensity $10n/(cm^3s)$ and Fokker-Planck cross sections with $\alpha = 1$. Σ_t is chosen to be equal to $\Sigma_{s,0}$. The quadrature is the Gauss-Legendre-Chebyshev triangular Galerkin quadrature. The GMRES solver is converged to a relative tolerance, $\left(\frac{\|\text{residual}\|_2}{\|\text{right hand side}\|_2}\right)$, of 10^{-5}. $P1$SA is solved using BiCGSTAB with a relative tolerance of 10^{-7}. DSA is solved with CG preconditioned by an algebraic multigrid technique [35, 38] with a relative tolerance of 10^{-7}. The number of GMRES iterations needed to solve ANMG-DSA (multigrid preconditioner with S_2 as coarsest transport level and diffusion solve) and ANMG-$P1$SA (multigrid preconditioner with S_4 as coarsest transport level and $P1$SA solve) are compared (see Table IV.1). The comparison is performed for S_4, S_8 and S_{16} (for which the values of the anisotropy order L are 4, 8, 16, respectively).

Table IV.1: Comparison of the number of GMRES iterations needed in ANMG-DSA and ANMG-$P1$SA

	ANMG-DSA	ANMG-$P1$SA
S_4	21	19
S_8	29	38
S_{16}	54	85

From Table IV.1, it can be seen that ANMG-DSA outperforms ANMG-$P1$SA except for S_4. When the anisotropy of the problem increases, the advantage of ANMG-DSA over ANMG-$P1$SA increases. Furthermore, we note that the $P1$SA equations are more difficult to solve (PD but non symmetric system) than the DSA equations (which are SPD). For these reasons, we recommend using the ANMG-DSA variant of the angular multigrid technique. Consequently, only the ANMG-DSA method will be employed in the later tests.

2 Test Case with a Volumetric Source

In this test, we compare ANMG-DSA to Sweep and DSA preconditioning. A uniform isotropic source of intensity 10 $n/(cm^3 s)$ was used. S_4, S_8, S_{16}, and S_{32} calculations were performed. Fokker-Planck cross sections with $\alpha = 1$ are used and $\Sigma_t = \Sigma_{s,0}$ ($c = 1$). The domain is homogeneous and its size is $5cm \times 5cm$

discretized by 50 × 50 cells. The thickness of the domain varies from 50 to 2690 mean free path (the total cross section varies with L for Fokker-Planck cross sections: $\Sigma_{t,S_4} = 10cm^{-1}, \Sigma_{t,S_8} = 36cm^{-1}, \Sigma_{t,S_{16}} = 136cm^{-1}, \Sigma_{t,S_{32}} = 528cm^{-1}$) but stays constant at five transport mean free path. The relative tolerance on GMRES, which is restarted every 30 iterations, is 10^{-6} whereas the relative convergence on DSA, solved by AGMG (see next Chapter), is 10^{-8}. The solution for S_{32} calculation is given on Figure IV.4

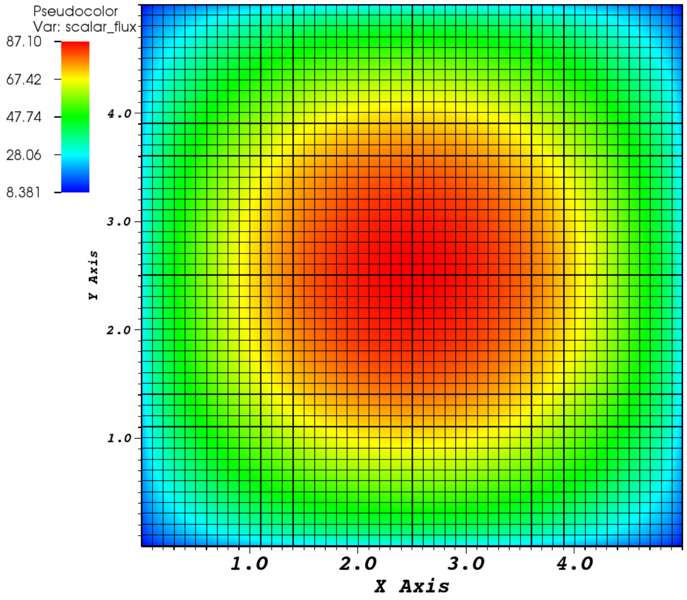


Figure IV.4: Scalar flux for the S_{32} calculation on a homogeneous medium

Table IV.2: Comparison of the number of GMRES iterations needed to solve the volumetric source test problem when $c = 1$ using sweep preconditioning (S), DSA preconditioning, and ANMG-DSA preconditioning on a homogeneous medium

	S	DSA	ANMG-DSA	$\frac{\text{ANMG-DSA}}{\text{DSA}}$
S_4	85	42	27	0.64
S_8	409	102	50	0.49
S_{16}	1526	266	105	0.39
S_{32}	4540	616	225	0.36

Table IV.3: Elapsed time (s) to solve the volumetric source test problem when $c = 1$ using sweep preconditioning (S), DSA preconditioning, and ANMG-DSA preconditioning on a homogeneous medium

	S	DSA	ANMG-DSA	$\frac{\text{ANMG-DSA}}{\text{SA}}$
S_4	9.09322	6.51608	5.22796	0.80
S_8	184.949	52.65	32.7609	0.62
S_{16}	4275.82	740.193	355.939	0.48
S_{32}	138819	17907.9	7357.63	0.41

In Table IV.2, one can note that ANMG-DSA always requires the least number of iterations to converge. ANMG is the fastest method (Table IV.3). It took 38 hours to solve the S_{32} problem with sweep preconditioning but only 2 hours when ANMG-DSA was used. As the anisotropy order is increased (i.e., increasing values of L as a function of the number of directions in the Fokker-Planck cross-section representation), the advantage of ANMG-DSA is clear. The ratio $\left(\frac{\text{number of GMRES iterations for ANMG-DSA}}{\text{number of GMRES iterations for DSA}}\right)$ and the ratio of elapsed times between the DSA and the ANMG-DSA techniques decrease monotonically. We note from these results that ANMG-DSA becomes increasingly superior to the standard DSA as the number of directions becomes larger. The time spent performing the diffusion solve DSA is negligible ($\leq 1\%$ for S_{16} and S_{32} calculations) and is the same for DSA and ANMG-DSA for a given number of CG iterations. We tried to run a problem with a S_{64} GLC Galerkin quadrature but unfortunately we could not. The reason is that the function that allocates the memory for GMRES in Trilinos receives the number of bytes to be allocated through an *unsigned int* and, therefore, the maximum size of the problem per processor is: (number of unknowns+2) × (size of the Krylov space+1) $\times\ 8 \leq 4294967295$. The number of 8 is because *double* are coded using 8 bytes and 4294967295 is the largest number of representable by an *unsigned int*. With S_{64} this problem requires 5158400496 bytes. This number is about 20% larger than what is allowed in Trilinos implementation. Moreover, in the function allocating the memory, the *unsigned int* is cast on a *int* before the allocation is done reducing the size of the largest problem by two. To run the S_{64} problem, GMRES can be restarted more often but this leads to a very slow convergence.

3 Test Case with a Volumetric Source with finer mesh cell sizes

The domain is a $6cm-$side square discretized by 600×600 cells. The quadrature used is a S_4 GLC Galerkin quadrature. There is a uniform source of intensity 10 $n/(cm^3s)$. Fokker-Planck cross sections are used with $\alpha = 1$ and $\Sigma_t = \Sigma_{s,0}$ ($c = 1$). In Table IV.4, we compare the number of GMRES iterations used by Sweep preconditioning, DSA preconditioning, angular multigrid preconditioning. In Table IV.5, the time needed to solve this problem is compared.

Table IV.4: Comparison of the number of GMRES iterations needed to solve a problem whose infinity medium version is unstable for ANMG-DSA with SI

S	DSA	ANMG-DSA
101	47	29

Table IV.5: Elapsed time (s) to solve a problem whose infinity medium version is unstable for ANMG-DSA with SI

S	DSA	ANMG-DSA
2767.87	1730.73	1314.07

We note that ANMG-DSA requires fewer iterations and less time than DSA which itself requires fewer iterations and less time than Sweep preconditioning. Using ANMG-DSA within GMRES is more efficient than DSA within GMRES, contrarily to what happens when SI is used. According to [86], if the medium is infinite, the spectral radius of the unfiltered ANMG method with SI, for such a test with fine mesh cells sizes, is 2.11. Because of leakage, unfiltered ANMG with SI would probably be convergent for this test but it should be less efficient than SI+DSA. The fact that ANMG-DSA with GMRES requires fewer iterations than DSA with GMRES is an indication that ANMG-DSA with GMRES is probably stable even if there is no leakage.

4 Test Case with a Heterogeneous Medium (Beam problem)

In this test, we apply a boundary source of intensity 10 $n/(cm^2s)$ to the entire left side of the domain $y \in [0cm, 5cm]$. The top, the bottom, and the right boundary conditions are vacuum. The beam intensity is only non-zero in the most-normal directions of the quadrature. An S_{16} Galerkin Gauss-Legendre-Chebyshev quadrature is used. The domain is discretized using 50×50 cells and is composed of two materials:

Material 1: for $x \in [0cm, 3cm]$, Fokker-Planck cross section is used with $\alpha = 0.099$, $\Sigma_t = 13.6cm^{-1}$, $c = 0.99$

Material 2: for $x \in [3cm, 5cm]$, Fokker-Planck cross section is used with $\alpha = 9.999$, $\Sigma_t = 1360cm^{-1}$, $c = 0.99$

Like previously, the relative tolerance on GMRES, which is restarted every 30 iterations, is 10^{-6} and the relative tolerance on DSA is 10^{-8}.

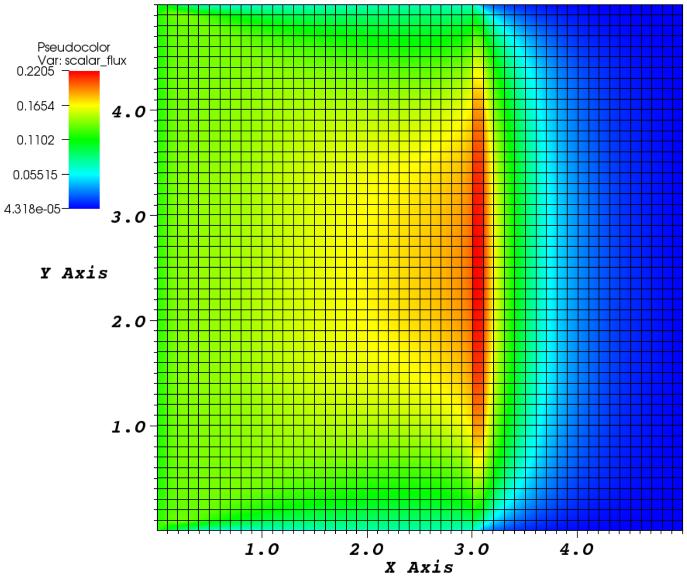


Figure IV.5: Scalar flux for the S_{32} calculation on a heterogeneous medium

The number of GMRES iterations and the elapsed time are given in Table IV.6 and Table IV.7.

Table IV.6: GMRES iterations to solve the heterogeneous problem using sweep preconditioning (S), DSA preconditioning, and ANMG-DSA preconditioning on a heterogeneous medium

S	DSA	ANMG-DSA
5001	283	96

Table IV.7: Elapsed time (s) to solve the heterogeneous problem using sweep preconditioning (S), DSA preconditioning, and ANMG-DSA preconditioning on a heterogeneous medium

S	DSA	ANMG-DSA
13642.8	897.072	394.965

We can see that even for a heterogeneous problem the angular multigrid is the most effective. If we compare Table IV.2 when the S_{16} quadrature is used with Table IV.6, we see that the number of iterations needed with DSA and ANMG-DSA preconditioning are quite similar. Sweep preconditioning, however, requires more iterations for the heterogeneous case because Material 2 is much thicker. We notice the same behavior when comparing Table IV.3 and Table IV.7 for CPU times.

G Conclusions

ANMG-DSA preconditioning is much more efficient than DSA preconditioning when the scattering kernel is highly forward peaked. Unlike MM, the number of iterations needed to solve a problem with ANMG-DSA does not saturate as the anisotropy increases. However, whereas the extra work due to the additional sweeps for MM is at most equivalent to the work done during the sweeps of a standard iteration, the extra work for ANMG-DSA is at most a sweep on $\frac{n^2}{6}+n$ different directions. This number must be compared to the $\frac{n^2}{2}+n$ directions of a standard iteration. Thus, ANMG-DSA becomes cheaper compared to standard iteration as the anisotropy order increases.

CHAPTER V

MODIFIED INTERIOR PENALTY ON ARBITRARY POLYGONAL CELLS

A Introduction

In Chapter IV, we noted that at the coarsest level of ANMG a Diffusion Synthetic Acceleration needs to be solved. Because analytical and closed form solutions are unavailable for most radiation transport problems of practical interest, one typically employs iterative techniques to solve the large system of equations that results from the spatial and angular discretization of the transport equation. Standard iterative techniques for the first-order form of the discrete-ordinate (S_n) transport equation include the Source Iteration (SI) technique and Krylov subspace algorithms (usually GMRes). For highly diffusive materials (i.e., with scattering ratios $c = \Sigma_s/\Sigma_t$ close to 1) and optically thick configurations (i.e., not leakage dominated), these iterative techniques can become quite ineffective, requiring high iteration counts and possibly leading to false convergence. However, SI and GMRes-based transport solves can be accelerated (preconditioned) with DSA approaches [19, 60, 61, 109, 105, 115].

It is well established that the spatial discretization of the DSA equations must be somewhat "consistent" with the one used for the S_n transport equations to yield unconditionally stable and efficient DSA schemes ([19, 60, 61, 109, 105, 115]). However, consistency between the discretized transport equations and the discretized diffusion may not be computationally practical (especially for unstructured arbitrary meshes, [19]). For instance, Warsa, Wareing, and Morel [115] derived a fully consistent DSA scheme for linear discontinuous finite elements on unstructured tetrahedral meshes; their DSA scheme yielded in a P_1 system of equations which was found to be computationally more expensive than partially consistent DSA schemes that are based upon discretizations of a standard diffusion equation. Some partially consistent schemes have been analyzed for linear discontinuous finite element (DFE) discretizations of the transport equation on unstructured meshes, for instance, the modified-four-step (M4S) scheme [60], the Wareing-Larsen-Adams (WLA) scheme [109, 32], and the Modified Interior Penalty (MIP) scheme [105].

We will come back to DSA later but first, we want to point the usefulness of using polygonal or polyhedral cells. Such cell types may present some advantages over traditional cells types (simplices, hexahedra) and have found some applications in radiation transport [25, 95, 114]. Meshing tools such as MSTK [16] and the Computational Geometry Algorithms Library [15] may be employed to process polyhedral meshes. The radiation transport code PDT and the CFD codes Fluent [5, 14] and OpenFOAM [17] offer polyhedral mesh and solver capabilities.

The following features of polygonal and polyhedral cells are noteworthy:

- Reduced number of unknowns per cell. To illustrate this, we assume one unknown per vertex in every cell, which is standard for transport discretizations that perform well in the thick diffusive regime. In the 2D hexagonal example of Figure V.1, the number of unknowns would be six (one unknown per vertex). Using triangular cells, the same hexagon would have to be split into four triangles at least (thus 12 unknowns) or possibly six triangles to preserve symmetry (thus 18 unknowns in that case). Similarly, using quadrilateral cells, the hexagon would be bisected into two quadrilaterals at least (8 unknowns), but divisions into three or four quadrilaterals are also possible (thus, 12 or 16 unknowns).

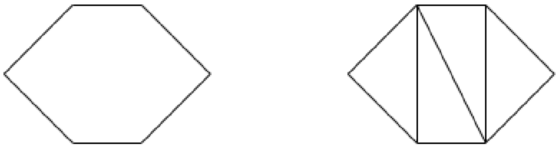


Figure V.1: Discretization using hexagonal cell versus triangle cells

- Transition elements and Adaptive Mesh Refinement. Solvers based on arbitrary polyhedral cells can easily handle cells with various numbers of edges (2D) and faces (3D). This can be particularly useful for simulations with Adaptive Mesh Refinement (AMR) [30, 51, 108], without having to deal with the implementation of data structures to handle hanging nodes [23, 94, 117]. On Figure V.2, the left cell is a degenerate pentagon whereas the two cells on the right are quadrilaterals. A similar illustration can be made for 3D hexahedral AMR meshes: suppose a cell is connect to four cells through one of its faces, such a cell can be thought of as a 9-face polyhedron. Thus, a method based on a piecewise linear discretization can handle locally adapted meshes without any special treatment or further approximation of the coupling between cells.

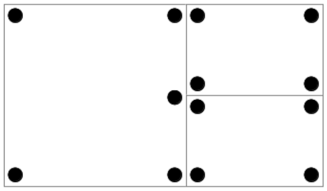


Figure V.2: Example of AMR mesh

Several discretization methods haven been developed for arbitrary polygonal meshes [25, 26, 27, 42, 58, 70, 80, 81, 95, 102, 114]. In this work, we focus on the PWLD discretization [25, 95]. This discretization can be applied for any polygonal cells and the integrals generated by this discretization can be easily computed analytically.

As of today, a lot of the ongoing effort to develop a DSA scheme on polygonal/polyhedral cells focuses on adapting the WLA scheme on polygonal meshes [32, 114]. The WLA scheme is a two-stage process, where first a diffusion solution is obtained using a *continuous* finite element discretization and then a *discontinuous* update is performed cell-by-cell in order to provide an appropriate discontinuous scalar flux correction to the DFE transport solver. In [115], the WLA scheme was found to be a stable and effective DSA technique, though its efficiency degraded as the problem became more optically thick and highly diffusive. To the author's best knowledge, no work is currently done to adapt the M4S technique to polygonal/polyhedral meshes. This is probably due to the fact that, even though the scheme is effective in one-dimensional slab and two-dimensional rectangular geometries, it was found to be divergent as an accelerator for SI in three-dimensional tetrahedral meshes with linear discontinuous elements. Furthermore, the scheme does not yield a Symmetric Positive Definite (SPD) matrix. In this paper, we present an extension of the MIP technique to the PWLD discretization techniques for for arbitrary polygonal/polyhedral meshes. The MIP scheme is based on the standard Interior Penalty (IP) for the discontinuous discretization of diffusion equations. MIP was first derived in [105], where it was applied to triangular unstructured meshes (with locally adapted cells). MIP did not suffer the same degradation of efficiency than WLA when the problem becomes optically thick and highly diffusive and it is therefore an interesting alternative to WLA. MIP uses the same discontinuous trial spaces as the transport finite element discretization. Because MIP produces SPD equations, it has been solved using conjugate gradient (CG) preconditioned by a symmetric successive over-relaxation method (SSOR) in [105]. Here, the effectiveness of algebraic multigrid methods (AMG) to precondition diffusion solver [38, 103] will be tested and compared with CG+SSOR. Algebraic multigrid methods allow the use of multigrid techniques when no grid information is available or when the grid is unstructured. Instead of using a succession of grids based on the geometry of the problems, the "grid levels" are based on properties of the matrix.

B Modified Interior Penalty on arbitrary polygonal cells

First, let us recall the derivation of MIP (see [105]). MIP is based on the Interior Penalty (IP) form of the diffusion equation [53, 105] (weakly imposed boundary conditions are applied to each grid cell and the test functions are averaged over cells). The continuous equations are :

$$\begin{aligned} -\boldsymbol{\nabla}\mathrm{D}\boldsymbol{\nabla}\phi_0 + \Sigma_a\phi_0 &= Q_0 && \text{for } \mathbf{r}\in\mathcal{D}, && (5.1)\\ \frac{1}{4}\phi_0 - \frac{1}{2}\mathrm{D}\boldsymbol{\nabla}\phi_0 &= 0 && \text{for } \mathbf{r}\in\partial\mathcal{D}^d, && (5.2)\\ -\mathrm{D}\partial_n\phi_0 &= J^{inc} && \text{for } \mathbf{r}\in\partial\mathcal{D}^r, && (5.3)\end{aligned}$$

where ϕ_0 is the scalar flux and D is the diffusion coefficient. Applying the IP technique, we obtain:

$$b_{IP}(\phi_0,\phi_0^*) = l_{IP}(\phi_0^*), \tag{5.4}$$

$$\begin{aligned} b_{IP}(\phi_0,\phi_0^*) = &(\Sigma_a\phi_0,\phi_0^*)_{\mathcal{D}} + (\mathrm{D}\boldsymbol{\nabla}\phi_0,\boldsymbol{\nabla}\phi_0^*)_{\mathcal{D}} + \left(\kappa_e^{IP}[\![\phi_0]\!],[\![\phi_0^*]\!]\right)_{E_h^i}\\ &+ ([\![\phi_0]\!],\{\!\{\mathrm{D}\partial_n\phi_0^*\}\!\})_{E_h^i} + (\{\!\{\mathrm{D}\partial_n\phi_0\}\!\},[\![\phi_0^*]\!])_{E_h^i} + \left(\kappa_e^{IP}\phi_0,\phi_0^*\right)_{\partial\mathcal{D}^d}\\ &- \frac{1}{2}\left(\phi_0,\mathrm{D}\partial_n,\phi_0^*\right)_{\partial\mathcal{D}^d} - \frac{1}{2}\left(\mathrm{D}\partial_n\phi_0,\phi_0\right)_{\partial\mathcal{D}^d}, \end{aligned} \tag{5.5}$$

and:

$$l_{IP}(\phi_0^*) = (Q_0,\phi_0^*)_{\mathcal{D}} + \left(J^{inc},\phi_0^*\right)_{\partial\mathcal{D}^r}, \tag{5.6}$$

where ϕ_0 and $\phi_0^*\in W_{\mathcal{D}}^h$ where the finite dimensional polynomial space is $W_{\mathcal{D}}^h = \{\phi_0\in L^2(\mathcal{D});\phi_0|_K\in V_p(K),\forall K\in\mathbb{T}_h\}$ where $V_p(K)$ is the space of polynomials of degree up to p on element K, $(f,g)_{\mathcal{D}} = \sum_{K\in\mathbb{T}_h}(f,g)_K$, $(f,g)_K = \int_K fg\ d\mathbf{r}$, $(f,g)_{E_h^i} =$

$\sum_{e \in E_h^i} (f, g)_e$, $(f, g)_e = \int_e fg\ ds$, $Q_0 = \Sigma_{s,0}\delta\phi_0$, $J^{inc} = \sum_{\boldsymbol{\Omega}_m \cdot \boldsymbol{n}_b > 0} w_m |\boldsymbol{\Omega}_m \cdot \boldsymbol{n}_b| \delta\psi_m$, $\mathbb{T}_h$ is the mesh used to discretize the spatial domain $\mathcal{D}$ into nonoverlapping elements K, E_h^i is the set of interior edges, $\partial\mathcal{D}^d$ is the boundary of the domain with Dirichlet condition, $\partial\mathcal{D}^r$ is the boundary of the domain with reflective condition, Σ_a is the absorption cross section, D is the diffusion coefficient, $\boldsymbol{n}_b$ is the outward normal unit vector, $\partial_n = \boldsymbol{n}_e \cdot \boldsymbol{\nabla}$ where $\boldsymbol{n}_e$ is the normal unit vector associated with a given edge e (on the boundary $\boldsymbol{n}_e = \boldsymbol{n}_b$), $[\![\phi_0]\!] = \phi_0^+ - \phi_0^-$ is the jump of ϕ_0 at the interface between two elements, $\{\!\{\phi_0\}\!\} = \frac{\phi_0^+ + \phi_0^-}{2}$ is the mean of ϕ at the interface between two elements, $\phi_0^\pm = \lim_{s \to 0^\pm} \phi_0(\boldsymbol{r} + s\boldsymbol{n}_e)$, Q_0 represents the volumetric source term due to the successive error in the scattering term, and J^{inc} is the incoming partial current. The penalty parameter κ_e^{IP} is given by:

$$\kappa_e^{IP} = \begin{cases} \dfrac{c\,(p^+)}{2}\dfrac{\mathrm{D}^+}{h_\perp^+} + \dfrac{c\,(p^-)}{2}\dfrac{\mathrm{D}^-}{h_\perp^-} & \text{on interior edges, i.e., } e \in E_h^i \\ c(p)\dfrac{\mathrm{D}}{h_\perp} & \text{on boundary edges, i.e., } e \in \partial\mathcal{D}^d \end{cases} \tag{5.7}$$

where $c(p)$ is given by $c(p) = 2p(p+1)$, p is the polynomial order ($p = 1$ in this research) and $h_\perp$ is the length of the cell in the direction orthogonal to the edge e. On triangular cells, $h_\perp$ equals $\frac{2A}{L_e}$ where A is the area of the triangle and L_e is the length of the edge e.

This discretization of DSA is SPD. Unfortunately, the authors in [105] found that IP yields unstable DSA scheme when the cells are large compared to the mean free path. This phenomenon is due to the fact that in optically thick medium, the ratios $\frac{D^\pm}{h^\pm}$ are very small. Therefore, the penalty coefficient is small and the method is unstable.

This led the authors of [105] to develop another discretization of DSA: the Diffusion Conforming Form (DCF). This discretization starts from the one-group S_n transport equation with isotropic source and scattering:

$$\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla}\psi_d(\mathbf{r}) + \Sigma_t(\mathbf{r})\psi_d = \frac{1}{4\pi}\Sigma_s(\mathbf{r})\phi_0(\mathbf{r}) + \frac{1}{4\pi}Q(\mathbf{r}). \tag{5.8}$$

The variational form of this equation is:

$$b(\psi, \psi^*) = l(\psi^*), \tag{5.9}$$

with:

$$b(\psi, \psi^*) = a(\psi, \psi^*) - \sum_{e\in\partial\mathcal{D}^r}\sum_{\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b<0} 4\pi w_m \left\langle \psi_{d'}, \psi_d^* \right\rangle_e - \left(\Sigma_s\phi_0, \phi_0^*\right)_{\mathcal{D}}, \tag{5.10}$$

$$\begin{aligned} a(\psi, \psi^*) =& \sum_{d=1}^{M} 4\pi w_d \left(\left(\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla} + \Sigma_t\right)\psi_d, \psi_d^*\right)_{\mathcal{D}} + \sum_{d=1}^{M} 4\pi w_d \left\langle [\![\psi_m]\!], \psi_m^{*,+} \right\rangle_{E_h^i} \\ &+ \sum_{e\in\partial\mathcal{D}^r}\sum_{\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b<0} 4\pi \left\langle \psi_d, \psi_d^* \right\rangle_e , \end{aligned} \tag{5.11}$$

$$l(\psi^*) = (Q, \phi^*)_{\mathcal{D}} + \sum_{e\in\partial\mathcal{D}^d}\sum_{\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b<0} 4\pi w_m \left\langle \psi_m^{inc}, \psi_m^* \right\rangle_e , \tag{5.12}$$

where $\langle f, g, \rangle_e = \int_e |\boldsymbol{\Omega}_d{\cdot}\boldsymbol{n}_e| fg\, ds$, $\langle f, g\rangle_{E_h^i} = \sum_{e\in E_h^i} \langle f, g\rangle_e$, and ψ and $\psi^* \in W_{\mathcal{D}}^h$ where the finite dimensional polynomial space is $W_{\mathcal{D}}^h = \{\psi \in L^2(\mathcal{D}); \psi|_K \in V_p(K), \forall K \in \mathbb{T}_h\}$ with $V_p(K)$ is the space of polynomials of degree up to p on element K, $(f, g)_{\mathcal{D}} = \sum_{K\in\mathbb{T}_h} (f, g)_K$.

The operator a consists of the streaming term, the interaction term and the upwind terms. This is the operator that is inverted during a transport sweep. The operator b contains a, the scattering term and the reflective boundary conditions. This operator is inverted upon convergence of SI or the Krylov solver.

The discretized SI at iteration ℓ can be written as:

$$a(\psi^{(\ell+1/2)}, \psi^*) = l(\psi^*) + (\Sigma_s \phi_0^{(\ell)}, \phi_0^*)_{\mathcal{D}} + \sum_{e \in \mathcal{D}^r} \sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b < 0} 4\pi w_d \left\langle \psi_{d'}^{(\ell)}, \psi_d^* \right\rangle_e, \tag{5.13}$$

$$\phi^{(\ell)} = \sum_{d=1}^{M} w_d \psi_d^{(\ell)}. \tag{5.14}$$

If we note the converged angular and scalar fluxes ψ^c and ϕ_0^c, we get:

$$a(\psi^c, \psi^*) = l(\psi^*) + (\Sigma_s \phi_0^c, \phi_0^*)_{\mathcal{D}} + \sum_{e \in \mathcal{D}^r} \sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b < 0} 4\pi w_d \left\langle \psi_{d'}^c, \psi_d^* \right\rangle_e, \tag{5.15}$$

$$\phi_0^c = \sum_{d=1}^{M} w_d \psi_d^c. \tag{5.16}$$

Subtracting equations (5.13) and (5.14) from equations (5.15) and (5.16) respectively, we obtain an angular error equation:

$$a(\epsilon^{(\ell+1/2)}, \psi^*) = \left(\Sigma_s \mathcal{E}^{(\ell)}, \phi_0^*\right)_{\mathcal{D}} + \sum_{e \in \partial\mathcal{D}^r} \sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_b < 0} 4\pi w_d \left\langle \epsilon_{d'}^{(\ell)}, \psi_m^* \right\rangle_e \tag{5.17}$$

and:

$$\mathcal{E}^{(\ell)} = \sum_{d=1}^{M} w_d \epsilon_d^{(\ell)}, \tag{5.18}$$

where the angular error and the scalar error are given by:

$$\epsilon^{(\ell)} = \psi^c - \psi^{(\ell)}, \tag{5.19}$$

$$\mathcal{E}^{(\ell)} = \phi_0^c - \phi_0^{(\ell)}. \tag{5.20}$$

It is important to note that the linear form l has disappeared from equation (5.17) and thus, the external volumetric source and the incident Dirichlet boundary conditions have disappeared. We now introduce:

$$\delta\psi^{(\ell)} = \psi^{(\ell+1/2)} - \psi^{(\ell)} = \epsilon^{(\ell)} - \epsilon^{(l+1/2)}, \tag{5.21}$$

$$\delta\phi_0^{(\ell)} = \phi_0^{(\ell+1/2)} - \phi_0^{(\ell)} = \mathcal{E}^{(\ell)} - \mathcal{E}^{(\ell+1/2)}. \tag{5.22}$$

Finally, we get the final form of the transport equation for the error:

$$b(\epsilon^{(\ell+1/2)}, \psi^*) = \left(\Sigma_s \delta\phi_0^{(\ell)}, \phi_0^*\right)_{\mathcal{D}} + \sum_{e\in\partial\mathcal{D}^r} \sum_{\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b<0} 4\pi w_d \left\langle \delta\psi_{d'}^{(\ell)}, \psi_d^* \right\rangle_e. \tag{5.23}$$

Note that solving equation (5.23) would give the exact correction needed to obtain the converged transport solution:

$$\psi^c = \psi^{(\ell+1/2)} + \epsilon^{(\ell+1/2)}, \tag{5.24}$$

$$\phi_0^c = \phi_0^{(\ell+1/2)} + \mathcal{E}^{(\ell+1/2)}, \tag{5.25}$$

but solving equation (5.23) is as difficult as solving our original transport equation. Therefore, we will replace the transport operator in equation (5.23) by a diffusion operator instead. The solution of this diffusion operator will be denoted by a ˜symbol:

$$\psi^{(\ell+1)} = \psi^{(\ell+1/2)} + \tilde{\epsilon}^{(\ell+1/2)}, \tag{5.26}$$

$$\phi_0^{(\ell+1)} = \phi_0^{(\ell+1/2)} + \tilde{\mathcal{E}}^{(\ell+1/2)}. \tag{5.27}$$

In the remainder of this section, to simplify the notation and the comparison between the final result of the development and equations (5.4) to (5.6), we use ϕ_0 instead of $\tilde{\mathcal{E}}^{(\ell+1/2)}$. If we assume that the primal and dual angular fluxes are linearly anisotropic (diffusion approximation) and we assume Fick's law to be valid:

$$\boldsymbol{J} = -\mathrm{D}\boldsymbol{\nabla}\phi_0, \tag{5.28}$$

$$\boldsymbol{J}^* = \mathrm{D}\boldsymbol{\nabla}\phi_0^*, \tag{5.29}$$

we then obtain:

$$\tilde{\epsilon}^{(\ell+1/2)} = \frac{1}{4\pi}(\phi_0 - 3\mathrm{D}\boldsymbol{\nabla}\phi_0 \cdot \boldsymbol{\Omega}_d), \tag{5.30}$$

$$\tilde{\psi}_d^* = \frac{1}{4\pi}(\phi_0^* + 3\mathrm{D}\boldsymbol{\Omega}\phi_0^* \cdot \boldsymbol{\Omega}_d). \tag{5.31}$$

Substituting equations (5.30) and (5.31) in equation (5.23), we obtain a discontinuous finite elements DSA operator in which the scalar flux ϕ_0 is the only unknown. Using:

$$\sum_{d=1}^{M} w_d = 4\pi, \tag{5.32}$$

$$\sum_{d=1}^{M} w_d\boldsymbol{\Omega}_d = 0, \tag{5.33}$$

$$\sum_{d=1}^{M} w_d\boldsymbol{\Omega}_d \cdot \boldsymbol{\Omega}_d = \frac{4\pi}{3}\boldsymbol{I}, \tag{5.34}$$

we obtain:

$$\sum_{d=1}^{M} 4\pi w_d \left(\Sigma_t\tilde{\epsilon}_d^{(\ell+1/2)}, \tilde{\psi}_d^*\right)_{\mathcal{D}} = (\Sigma_t\phi_0, \phi_0^*)_{\mathcal{D}} - (3\Sigma_t\mathrm{D}\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*)_{\mathcal{D}}\,, \tag{5.35}$$

$$\left(\Sigma_s\tilde{\boldsymbol{\mathcal{E}}}^{(\ell+1/2)}, \tilde{\phi}^*\right)_{\mathcal{D}} = (\Sigma_s\phi_0, \phi_0^*)_{\mathcal{D}}\,. \tag{5.36}$$

If we define:

$$\mathrm{D} = \frac{1}{3\Sigma_t}, \tag{5.37}$$

we get:

$$\begin{aligned}&\sum_{d=1}^{M} 4\pi w_d \left(\Sigma_t \tilde{\epsilon}_d^{(\ell+1/2)}, \tilde{\psi}_d^*\right)_{\mathcal{D}} + \left(\Sigma_s \tilde{\mathcal{E}}^{(\ell+1/2)}, \tilde{\phi}^*\right)_{\mathcal{D}} = \left(\Sigma_a \phi_0, \phi_0^*\right)_{\mathcal{D}} - \\ &\left(\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\mathcal{D}}.\end{aligned} \tag{5.38}$$

Now, we analyze the streaming term:

$$\begin{aligned}\sum_{d=1}^{M} 4\pi w_d \left(\boldsymbol{\Omega}_d \cdot \boldsymbol{\nabla}\tilde{\epsilon}_d^{(\ell+1/2)}, \tilde{\psi}_d^*\right)_{\mathcal{D}} &= \left(\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\mathcal{D}} - \left(\boldsymbol{\nabla}\cdot\mathrm{D}\boldsymbol{\nabla}\phi_0, \phi_0^*\right)_{\mathcal{D}} \\ &= \left(\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\mathcal{D}} - \left(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\phi_0, \phi_0^*\right)_{\mathcal{D}} \\ &= \left(\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\mathcal{D}} + \left(\mathrm{D}\boldsymbol{\nabla}\phi_0, \boldsymbol{\nabla}\phi_0^*\right)_{\mathcal{D}} \\ &\quad + \left(\mathrm{D}\boldsymbol{\nabla}\phi_0^+ \cdot \boldsymbol{n}_e, \phi_0^{*,+}\right)_{E_h^i} \\ &\quad - \left(\mathrm{D}\boldsymbol{\nabla}\phi_0^- \cdot \boldsymbol{n}_e, \phi_0^{*,-}\right)_{E_h^i} \\ &\quad - \left(\mathrm{D}\boldsymbol{\nabla}\phi_0 \cdot \boldsymbol{n_e}, \phi_0^*\right)_{\partial\mathcal{D}},\end{aligned} \tag{5.39}$$

where integration by part was performed and we used:

$$\sum_{d=1}^{M} w_d \boldsymbol{\Omega}_d \cdot \boldsymbol{\Omega}_d \cdot \boldsymbol{\Omega}_d = 0. \tag{5.40}$$

Manipulating the interior edge terms give successively :

$$
\begin{aligned}
\sum_{d=1}^{M} 4\pi w_d \left\langle \llbracket \tilde{\epsilon}_d^{(\ell+1/2)} \rrbracket, \tilde{\psi}_d^{*,+} \right\rangle_{E_h^i} &= \sum_{e \in E_h^i} \sum_{d=1}^{M} 4\pi w_d |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e| \left(\llbracket \tilde{\epsilon}_m^{(\ell+1/2)} \rrbracket, \tilde{\psi}_d^{*,+} \right)_e \\
&= \sum_{e \in E_h^i} \Biggl(\sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e > 0} \frac{w_d}{4\pi} |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e| \bigl(\llbracket \phi_0 \rrbracket - 3 \\
&\quad \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \rrbracket \cdot \boldsymbol{\Omega}_d, \phi_0^{*,+} + 3 \mathrm{D} \boldsymbol{\nabla} \phi_0^{*+} \cdot \boldsymbol{\Omega}_d \bigr)_e - \\
&\quad \sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e < 0} \frac{w_d}{4\pi} |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e| \Biggl(\llbracket \phi_0 \rrbracket - 3 \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \rrbracket \cdot \boldsymbol{\Omega}_d, \\
&\quad \phi_0^{*,-} + 3 \mathrm{D} \boldsymbol{\nabla} \phi_0^{*,-} \cdot \boldsymbol{\Omega}_d \Biggr)_e \Biggr) \\
&= \sum_{e \in E_h^i} \sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e > 0} \frac{w_d}{4\pi} |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}_e| \left(\left(\llbracket \phi_0 \rrbracket - 3 \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \rrbracket \cdot \right.\right. \\
&\quad \boldsymbol{\Omega}_d, \phi_0^{*,+} + 3 \mathrm{D} \boldsymbol{\nabla} \phi_0^{*,+} \cdot \boldsymbol{\Omega}_d \bigr)_e - \bigl(\llbracket \phi_0 \rrbracket + 3 \\
&\quad \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \rrbracket \cdot \boldsymbol{\Omega}_d, \phi_0^{*,-} - 3 \mathrm{D} \boldsymbol{\nabla} \phi_0^{*,-} \cdot \boldsymbol{\Omega}_d \bigr)_e \bigr) \\
&= \frac{1}{4} \left(\llbracket \phi_0 \rrbracket, \llbracket \phi_0^* \rrbracket \right)_{E_h^i} + \left(\llbracket \phi_0 \rrbracket, \{\!\{ \mathrm{D} \boldsymbol{\nabla} \phi_0^* \cdot \boldsymbol{n} \}\!\} \right)_{E_h^i} \\
&\quad - \left(\llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \cdot \boldsymbol{n} \rrbracket, \{\!\{ \phi_0^* \}\!\} \right)_{E_h^i} - \frac{9}{16} \bigl(\llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \rrbracket, \\
&\quad \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0^* \rrbracket \bigr)_{E_h^i} - \frac{9}{16} \bigl(\llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0 \cdot \boldsymbol{n} \rrbracket, \\
&\quad \llbracket \mathrm{D} \boldsymbol{\nabla} \phi_0^* \cdot \boldsymbol{n} \rrbracket \bigr)_{E_h^i} \, ,
\end{aligned}
\tag{5.41}
$$

where we employed the following properties of the angular quadrature:

$$
\sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n} > 0} w_d |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}| \approx \pi, \tag{5.42}
$$

$$
\sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n} > 0} w_d |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}| \boldsymbol{\Omega}_d \approx \frac{2\pi}{3} \boldsymbol{n}, \tag{5.43}
$$

$$
\sum_{\boldsymbol{\Omega}_d \cdot \boldsymbol{n} > 0} w_d |\boldsymbol{\Omega}_d \cdot \boldsymbol{n}| \boldsymbol{\Omega}_m \cdot \boldsymbol{\Omega}_d \approx \frac{\pi}{4} \left(\boldsymbol{I} + \boldsymbol{n}\boldsymbol{n} \right), \tag{5.44}
$$

where $\boldsymbol{nn}$ is a matrix. Even if these properties cannot be strictly satisfied, numerical results show that the error is negligible. Finally, we obtain:

$$b_{DCF}(\phi_0, \phi_0^*) = l_{DCF}(\phi_0^*), \tag{5.45}$$

with:

$$\begin{aligned} b_{DCF}(\phi_0, \phi_0^*) = & \left(\Sigma_a \phi_0, \phi_0^*\right)_{\mathcal{D}} + \left(\mathrm{D}\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0\right)_{\mathcal{D}} + \frac{1}{4}\left(\llbracket\phi_0\rrbracket, \llbracket\phi_0^*\rrbracket\right)_{E_h^i} \\ & + \left(\llbracket\phi_0\rrbracket, \{\!\!\{\mathrm{D}\partial_n\phi_0^*\}\!\!\}\right)_{E_h^i} + \left(\{\!\!\{\mathrm{D}\partial_n\phi_0\}\!\!\}, \llbracket\phi_0^*\rrbracket\right)_{E_h^i} \\ & + \frac{1}{4}\left(\phi_0, \phi_0^*\right)_{\partial\mathcal{D}^d} - \frac{1}{2}\left(\phi_0, \mathrm{D}\partial_n\phi_0^*\right)_{\partial\mathcal{D}^d} \\ & - \frac{1}{2}\left(\mathrm{D}\partial_n\phi_0, \phi_0\right)_{\partial\mathcal{D}^d} - \frac{9}{16}\left(\llbracket\mathrm{D}\boldsymbol{\nabla}\phi_0\rrbracket, \llbracket\mathrm{D}\boldsymbol{\nabla}\phi_0^*\rrbracket\right)_{E_h^i} \\ & - \frac{9}{16}\left(\llbracket\mathrm{D}\partial_n\phi_0\rrbracket, \llbracket\mathrm{D}\partial_n\phi_0^*\rrbracket\right)_{E_h^i} - \frac{9}{16}\left(\mathrm{D}\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\partial\mathcal{D}^d} \\ & - \frac{9}{16}\left(\mathrm{D}\partial_n\phi_0, \mathrm{D}\partial_n\phi_0^*\right)_{\partial\mathcal{D}^d} - \frac{9}{4}\left(\mathrm{D}\partial_n\phi_0, \mathrm{D}\partial_n\phi_0^*\right)_{\partial\mathcal{D}^r}, \end{aligned} \tag{5.46}$$

$$\begin{aligned} l_{DCF}(\phi_0^*) = & \left(Q_0, \phi_0^*\right)_{\mathcal{D}} + \left(J^{inc}, \phi_0^*\right)_{\partial\mathcal{D}^r} - \left(\boldsymbol{\mathcal{Y}}, \mathrm{D}\boldsymbol{\nabla}\phi_0^*\right)_{\partial\mathcal{D}^r} \\ & + 2\left(\boldsymbol{\mathcal{Y}}^{inc}\cdot\boldsymbol{n}, \mathrm{D}\partial_n\phi_0^*\right)_{\partial\mathcal{D}^r}, \end{aligned} \tag{5.47}$$

where $\boldsymbol{\mathcal{Y}}^{inc} = -\sum_{\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b>0} 3w_d\boldsymbol{\Omega}_d|\boldsymbol{\Omega}_d\cdot\boldsymbol{n}_b|\delta\psi_d^{(\ell)}$.

DCF is symmetric but not positive definite positive. DCF is unstable for cell size in between one and four mean-free-paths but it is stable and very efficient for optically thick medium. In this case, $\boldsymbol{\nabla}\phi_0 \approx 0$ and $\partial_n\phi_0 \approx 0$ and the terms $\left(\mathrm{D}\boldsymbol{\Omega}\phi_0^{\pm}, \mathrm{D}\boldsymbol{\nabla}\phi_0^{*,\pm}\right)$ and $\left(\mathrm{D}\partial_n\phi_0^{\pm}, \mathrm{D}\partial_n\phi_0^{\pm}\right)$ are negligible. In this limit, b_{DCF} becomes:

$$\begin{aligned} b_{DCF}(\phi_0, \phi_0^*) = & \left(\Sigma_a \phi_0, \phi_0^*\right)_{\mathcal{D}} + \left(\mathrm{D}\boldsymbol{\nabla}\phi_0, \mathrm{D}\boldsymbol{\nabla}\phi_0\right)_{\mathcal{D}} + \frac{1}{4}\left(\llbracket\phi_0\rrbracket, \llbracket\phi_0^*\rrbracket\right)_{E_h^i} \\ & + \left(\llbracket\phi_0\rrbracket, \{\!\!\{\mathrm{D}\partial_n\phi_0^*\}\!\!\}\right)_{E_h^i} + \left(\{\!\!\{\mathrm{D}\partial_n\phi_0\}\!\!\}, \llbracket\phi_0^*\rrbracket\right)_{E_h^i} + \frac{1}{4}\left(\phi_0, \phi_0^*\right)_{\partial\mathcal{D}^d} \\ & - \frac{1}{2}\left(\phi_0, \mathrm{D}\partial_n\phi_0^*\right)_{\partial\mathcal{D}^d} - \frac{1}{2}\left(\mathrm{D}\partial_n\phi_0, \phi_0\right)_{\partial\mathcal{D}^d}, \end{aligned} \tag{5.48}$$

which is exactly the same as equation (5.5) if $\kappa_e^{IP} = \frac{1}{4}$.

MIP is obtained by replacing the penalty coefficient, κ_e^{IP}, by $\kappa_e^{MIP} = \max\left(\kappa_e^{IP}, \frac{1}{4}\right)$ in equation (5.5). This ensures that MIP will converge in optically thick medium since the penalty coefficient will never be less than $\frac{1}{4}$.

DSA gives only a correction for the scalar flux but by assuming that the angular dependence satisfies a diffusion expansion, the angular correction can be computed using equation (5.30). This correction can be used when some of the boundary conditions are periodic or reflective.

If PWLD finite elements are used instead of LD finite elements, the weak form of MIP is not modified. However when the cells are not triangular, there is no simple way to compute $h_\perp$. To simplify this, we assume that the polygonal cells are not too far from being regular polygonal cells. In such cases, if the cell has an even number of edges, the orthogonal length equals two times the apothem, i.e. two times the segment between the midpoint of a side of the polygon and the center of this polygon $\left(\text{apothem} = 2 \times \frac{\text{area}}{\text{perimeter}}\right)$. If the cell has an odd number of edges, the orthogonal length is given by the apothem plus the circumradius, i.e. the radius of the circle circumscribed to the polygon $\left(\text{circumradius} = \sqrt{\frac{2\times\text{area}}{V\sin\left(\frac{2\pi}{V}\right)}}\right)$. Therefore, $h_\perp$ is given in Table V.1 by:

Table V.1: Orthogonal length of the cell for different cells

Number of edges	3	4	> 4 and even	> 4 and odd
$h_\perp$	$2 \times \frac{\text{area}}{L_e}$	$\frac{\text{area}}{L_e}$	$4 \times \frac{\text{area}}{\text{perimeter}}$	$2 \times \frac{\text{area}}{\text{perimeter}} + \sqrt{\frac{2\times\text{area}}{V\sin\left(\frac{2\pi}{V}\right)}}$

C Algebraic Multigrid

1 Introduction

As mentioned earlier, the most common way to solve a SPD system is to use conjugate gradient preconditioned with SSOR (PCG-SSOR). In this research, we will compare the calculation time using PCG-SGS, which is PCG-SSOR with a damping factor equal to unity, with the time needed by CG preconditioned with an algebraic multigrid method. This is not a new idea: the first multigrid methods developed were geometric multigrid used as stand-alone solvers. In many applications, they achieve the so-called "textbook multigrid efficiency", i.e. "the solution to the governing system of equations [is attained] in a computational work that is a small multiple of the operation counts associated with discretizing the system" [97]. However, in many other applications, multigrid methods, and particularly algebraic multigrid methods, cannot achieve such efficiency [78]. In such cases, they are often used as preconditioner for Krylov subspace methods. AMG methods make very good preconditioners because they can reduce all the error modes. Of course, in some cases, some modes may not be accelerated which can significantly degrades the efficiency of AMG as preconditioner. In [82], the authors used an algebraic multigrid method to precondition the Krylov solver for the even-parity finite element-spherical harmonics (FE-P_N) method. The AMG preconditioner resulted in a 60% reduction in the solution time compared to ILU(0) preconditioning and even more reduction compared to SSOR preconditioning.

We will employ and compare two multigrid approaches: one from the ML package [46] of the Trilinos library, and the AGMG code [77]. ML is a multigrid preconditioning package that uses a smoothed aggregation algebraic multigrid to build a preconditioner for a Krylov method. AGMG is an aggregation-based algebraic multigrid code (written in Fortran 90).

We describe the multigrid principles, using first a two-grid setting. Consider the following system:

$$\boldsymbol{A}_f u_f = b_f, \tag{5.49}$$

defined on the fine grid $\mathbb{T}_f$. The two-grid algorithm is given by :

1. Perform ν_1 pre-smoothing iterations using a smoother (e.g., Jacobi, Gauss-Seidel or ILU) using an initial guess u_0: $u = S^{\nu_1}(u_0, b_f)$.

2. Compute the residual on the fine grid $\mathbb{T}_f$ and restrict it to the coarse grid $\mathbb{T}_c$: $r_c = \boldsymbol{R}(b_f - \boldsymbol{A}_f u)$.

3. Solve with a direct solver the system on the coarse grid: $v = \boldsymbol{A}_c^{-1} r_c$.

4. Interpolate the coarse grid correction to the fine grid and add the correction to u: $u \leftarrow u + \boldsymbol{P}v$.

5. Perform ν_2 post-smoothing iterations: $u = S^{\nu_2}(u, b_f)$.

When using AMG, the matrix $\boldsymbol{A}_c$ on the coarse grid is given by the Galerkin approximation:

$$\boldsymbol{A}_c = \boldsymbol{R}\boldsymbol{A}_f\boldsymbol{P}, \tag{5.50}$$

where $\boldsymbol{P}$ is a prolongation matrix and $\boldsymbol{R}$ is a restriction matrix. Solving the system $\boldsymbol{A}_c v = r_c$ on the coarse grid is generally very expensive. Therefore this step is recursively replaced by γ applications of the two-grid methods until the system can be efficiently inverted with a direct solver. This yields the multigrid method. When $\gamma = 1$, the multigrid method is said to use a $V-$cycle. When $\gamma = 2$, the cycle used is called the $W-$cycle. On Figure V.3, these two cycles are represented:

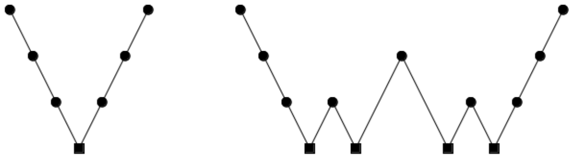


Figure V.3: $V-$ and $W-$cycles

A dot represents a smoothing operation and a square a direct inversion. The grid transfer operators are symbolized by lines.

For the coarsening step, both geometric and algebraic multigrid methods are based on the concept of smooth error. The difference between the two methods is that for geometric multigrid methods, after the smoothing step, the error is *geometrically* smooth relative to the coarse grid [96]. For algebraic multigrid methods, there might not be any grid and thus, only the properties of the matrix can be used. Therefore, the geometrical smoothness of the error cannot be assumed anymore. In fact, after the smoothing step, the error may be not smooth at all from the geometrical point of view. The reason is that the error is considered smooth when the smoother does not change the solution significantly anymore [103]:

$$\|Se\|_{\mathcal{H}} \approx \|e\|_{\mathcal{H}}, \tag{5.51}$$

where S is the smoother, e is the error, and $\|u\|_{\mathcal{H}} = \sqrt{(u,u)_{\mathcal{H}}}$ is the norm associated to the scalar product:

$$(u,v)_{\mathcal{H}} = (\boldsymbol{A}u, v)_2 \,. \tag{5.52}$$

Among the algebraic multigrid methods, there are three main types: the classical Ruge-Stueben AMG (also known as interpolation method), the plain aggregation AMG, and the smoothed aggregation AMG. ML uses smoothed aggregation AMG and AGMG uses plain aggregation AMG. The difference between theses methods is the coarsening step. The coarsening step is the most important step because if the coarsening is too fast, the convergence rates will decrease. However, if the coarsening is too slow, a lot of memory may be required to solve the problem. For classical Ruge-Stueben methods, each variable of the coarse grid is also a variable in the fine grid whereas for the aggregation methods, the variables of the fine grid are aggregates in variables of the coarse grid. There is no simple identification between the variables of the fine grid and the coarse grid. However, all the algebraic multigrid methods use the very important concept of strongly dependent variables [38].

Definition C.1. *Given a threshold value of* $0 \leq \theta \leq 1$*, the variable* u_i *strongly depends on the variable* u_j *if:*

$$-a_{ij} \geq \theta \max_{k \neq i} \left(-a_{ik}\right), \tag{5.53}$$

a_{ij} must be of the same order of magnitude than the largest off-diagonal in equation i or j. A related definition is:

Definition C.2. *If the variable* u_i *strongly depends on the variable* u_j*, then the variable* u_j *strongly influences the variable* u_i*.*

The idea behind the strong dependence is that if the coefficient a_{ij} is large, then a small change in the j^{th} variable will have an important effect on the i^{th} variable. Thus, it is probably a good idea to use the j^{th} variable to interpolate the i^{th} variable or to couple these two variables in an aggregate. This can be easily seen using the concept of smoothed error. For the error to be considered to be smoothed, assuming that $\boldsymbol{A}$ is a M-matrix, i.e., off-diagonal entries of the matrix are less than or equal to zero and the real parts of the eigenvalues of the matrix are positive, the following relationship needs to be satisfied for most i [38]:

$$\sum_{j \neq i} \left(\frac{|a_{ij}|}{a_{ii}} \right) \left(\frac{e_i - e_j}{e_i} \right)^2 \ll 1, \tag{5.54}$$

where e_i is the error associated with the variable i. Since the left side of equation (5.54) is positive, all the products must be small which means that at least one of the two terms of each product has to be small. When the i^{th} variable strongly depends on the j^{th} variable $\frac{|a_{ij}|}{a_{ii}} \approx 1$, $e_i - e_j$ must be small or equivalently $e_i \approx e_j$. This means that the error varies slowly in the direction of strong connection. That is the reason why the coarsening is done along these directions.

2 Classical AMG (interpolation method)

For classical AMG, the variables of the coarse grid are a subset of the variables of the fine grid. The variables can be split in two disjoint sets: C that contains all the coarse variables and F that contains all the other variables. Thus, the error on the fine grid is given by [96]:

$$e_{c,i} = (\boldsymbol{P} e_f)_i = \begin{cases} e_{c,i} & \text{if } i \in C \\ \displaystyle\sum_{k \in B_i} w_{ik} e_{c,k} & \text{if } i \in F \end{cases} \tag{5.55}$$

where B_i is a subset of C whose variables are called interpolatory variables. B_i should be a small subset of C to keep $\boldsymbol{A}_c$ sparse. Now, we assume that $\boldsymbol{A}$ is a M-matrix and we review two typical interpolation methods:

Direct interpolation: First, we define the neighborhood of the i^{th} as the set $N_i =$

$\{j \in C \cup F : j \neq i, a_{ij} \neq 0\}$. After the smoothing step, we can write locally:

$$e_i \approx -\frac{\left(\sum_{j \in N_i} a_{ij} e_j\right)}{a_{ii}}. \tag{5.56}$$

If B_i contains the variables which are strongly dependent on the i^{th} variables, we have:

$$\frac{1}{\sum_{k \in B_i} a_{ik}} \sum_{k \in B_i} a_{ik} e_k \approx \frac{1}{\sum_{j \in N_i} a_{ij}} \sum_{j \in N_i} a_{ij} e_j. \tag{5.57}$$

Using this relation and equation (5.56), we get the following formula for the weights of the interpolation:

$$w_{ik} = -\alpha_i \frac{a_{ik}}{a_{ii}}, \tag{5.58}$$

where $\alpha_i = \frac{\sum_{j \in N_i} a_{ij}}{\sum_{l \in B_i} a_{il}}$. Therefore, it is important that when the coarse variables are chosen, that every variable in F has enough strongly coupled variables in C that are part of B_i. If some of the off-diagonal entries are positive, the same development can be performed as long as these positive terms are small, i.e., variables are not strongly coupled because of these terms. If the positive entries are large, the algebraically smooth error can oscillate. This can happen, for elliptic PDE, when high-order finite elements are used or with bilinear elements on quadrilateral meshes with large aspect ratios. This will negatively affect the performance of AMG.

More complex interpolations: More complex interpolation schemes can be created but they reduce the sparsity of $\boldsymbol{P}$ and $\boldsymbol{R}$, increasing the size of $\boldsymbol{A}_c$. Moreover, the weakly-dependent variables will be associated with smaller weights which means that they will have a small effect. It can therefore be interesting to ignore the smallest values in the interpolation matrix and to rescale the others weights so that the sum of the weights does not change. This can slow down the convergence of the method but it will not make it diverge [96].

A good rule, when coarsening the grid, is to try to have the set of coarse variables form a maximally independent set, i.e. a maximal set where the coarse variables are not strongly coupled to each others, and the variables in F are surrounded by the variables in C. We call B_i^S the set of all strongly connected neighbors of u_i:

$$B_i^s = \{v_j \in B_i | -a_{ij} \geq \theta \max_{k \neq i}(-a_{i,k})\}. \tag{5.59}$$

The interpolatory nodes C_i are:

$$C_i = B_i^s \cap C. \tag{5.60}$$

Adding variables in C_i increases the quality of the interpolation but it diminishes the sparsity of the interpolation matrix and increases the size of $\boldsymbol{A}_c$ which increases the computational cost of the method. Thus, every variable u_i in F and every $u_j \in B_i^s$ should be in C_i or strongly connected to at least one variable in C_i. This rule will make sure that the interpolation is of a good enough quality. We also want C to be a maximal subset of the variables such that the variables in C are not strongly connected to each others. This ensures that the coarsening is fast enough.

3 Smoothed aggregation: the ML package

The smoothed aggregation method uses the concept of strong connections in a manner similar to that of classical AMG. Theory for the plain aggregation method shows that the convergence bound depends of the number of levels [99]. This is a major flaw of the plain aggregation method which was also observed in practice. To counter this, the smoothed aggregation was created. This method converges fast for a lot of different problems including the ones with anisotropic and discontinuous coefficients.

When using a smoothed aggregation scheme, the smoothed interpolation operators, $\boldsymbol{P}_k$, are the transpose of the coarsening operators, $\boldsymbol{R}_k = \boldsymbol{P}_k^T$. Therefore, when the $\boldsymbol{P}_k$ are built, the coarsening is known. First, the graph of the matrix is constructed: if the element (i,j) or (j,i) of the matrix is non-zero, an edge is built between the vertex i and the vertex j [46]. Second, the vertices are aggregated. When using ML on a single processor, two aggregation schemes can be used: the uncoupled scheme or the maximally independent sets (MIS) scheme. The uncoupled scheme attempts to build aggregates of size 3^d where d is the dimension of the problem. The algorithm works as follows [98]:

Step 1: As long as there are points not adjacent to an aggregate:

1. Choose a point which is not adjacent to an aggregate. This point is a new root point.
2. Define a new aggregate as the root point and its neighbors.

Step 2: Add all the points left of the existing aggregates or form new aggregates with them.

The MIS scheme used in ML applied the MIS algorithm of [52] to the graph produced by the matrix $\boldsymbol{A}^2$. These two coarsening schemes use a fixed ratio of coarsening between levels. Once the aggregation is done, a tentative prolongator matrix, $\tilde{\boldsymbol{P}}_k$ is constructed [98]. A example of $\tilde{\boldsymbol{P}}_k$ is given by:

$$\tilde{\boldsymbol{P}}_k(i,j) = \begin{cases} 1 & \text{if } i^{th} \text{ point is contained in } j^{th} \text{ aggregate} \\ 0 & \text{otherwise} \end{cases} \tag{5.61}$$

This tentative prolongator could be used as prolongator but smoothing it yields a more robust scheme. Let $\boldsymbol{S}_k$ be a smoother, for example damped Jacobi, then the prolongator matrix is given by:

$$\boldsymbol{P}_k = \boldsymbol{S}_k \tilde{\boldsymbol{P}}_k. \tag{5.62}$$

As in classical AMG, it can be interesting to ignore small values in the graph since the smoother will be ineffective for the weakly coupled variables. In ML, there is a drop tolerance, *tol*, that is used to ignore entries in the graph if $|a_{ij}| \leq tol \sqrt{|a_{ij}a_{jj}|}$. The tolerance, whose default value is zero, can be changed. In ML, when the matrix is SPD, CG is used to determine the Jacobi damping parameter, which is an approximation of the spectral radius.

By default, the coarsening is stopped when the number of variables is less or equal than 128.

4 Plain aggregation: the AGMG code

Unlike ML, in AGMG the prolongator is not smoothed which results in a cheaper set-up and a decrease of required memory [74]. However, the scheme could be less robust. To counteract this weakness, the aggregation scheme is more complicated. Coarsening algorithms that control the size of the aggregates tends to produce a few badly shaped aggregates. Since the convergence of AMG is bounded by the worst aggregate, even a small number of badly shaped aggregates can have a huge impact on the convergence. In AGMG, the aggregation algorithm has as input the upper bound of the two-grid condition number $\bar{\kappa}_{TG}$. When the aggregates are constructed, their quality is checked. Obviously, this increases the cost of the coarsening and it is important that the coarsening be fast enough. Since the algorithm does not control the size of the aggregates, it is difficult to control the speed of the coarsening.

However, controlling the condition number is much more interesting than controlling the coarsening speed. If the algorithm controls the condition number, it will not create bad aggregates but instead, it may create a few aggregates with a size below the target size but this does not affect the efficiency of the method in a noticeable way [74].

In AGMG, the aggregation is done by a few passes of a pairwise aggregation algorithm. This allows the computation of the aggregate quality to remain very simple and to keep the cost per iteration low. The advantage of controlling the condition number becomes even more important when a $K-$cycle or Krylov-cycle is used instead of the more common $V-$ or $W-$cycles. The difference between the $K-$cycle and the $V-$ or $W-$cycle is that the $K-$cycle uses recursively a few iterations of a Krylov solver preconditioned by a coarser grid to solve the coarse grid problem in the two-grid algorithm [78]. This scheme is nonlinear and when the system is SPD, it requires the use of flexible CG [24, 36, 48, 76] as the Krylov solver. The advantage of the $K-$cycle is increased robustness compared to $V-$ and $W-$cycles. Even when the condition number of the two-grid method is large, the convergence properties of the $K-$cycle can be independent of the number of levels [78]. The computational cost of a $K-$cycle is about the same as the cost of the $W-$cycle. If the number of unknowns does not decrease sufficiently from one level to the next, the $K-$cycle at one level is replaced by a $V-$cycle at this same level. The idea of $K-$cycle is not new since it was already used in Algebraic MultiLevel Iteration (AMLI) methods [56, 75].

Next, we explain the coarsening step in AGMG for an $M-$matrix (SPD). We want to create nonempty disjoint sets G_k, $k = 1, \dots, n_c$ called aggregates with each one of them associated with a variable on a coarser grid. Some of the unknowns are not associated with any variables in the coarse grid and they are in the set G_0. The prolongation matrix is given by:

$$\boldsymbol{P}_{ij} = \begin{cases} 1 \text{ if } i \in G_j \\ 0 \text{ otherwise} \end{cases} \tag{5.63}$$

Thus, $\boldsymbol{P}$ has at most one non-zero entry per row. A row is only composed of zeros if the variable associated to this row is in G_0. A simple method to form the high quality aggregates of a given size would be to test all the possibility. For an obvious reason, this cannot be done in practice. Instead, in AGMG several passes of pairwise aggregation are done. The reason is that when two variables are aggregated, the quality factor of the aggregate $\kappa(G)$ is given by:

$$\kappa(\{i,j\}) = \frac{-a_{ij} + \left(\frac{1}{a_{ii}+s_i+2a_{ij}} + \frac{1}{a_{jj}+s_j+2a_{ij}}\right)^{-1}}{-a_{ij} + \left(\frac{1}{a_{ii}-s_i} + \frac{1}{a_{jj}-s_j}\right)^{-1}}, \tag{5.64}$$

where $s_i = -\sum_{j\neq i} a_{ij}$. κ is only given by the off-diagonal entry connecting these two unknowns, their respective diagonal entries, and the sum of all off-diagonal elements in the corresponding rows. As $|G|$ increases, it becomes more and more costly to compute $\kappa(|G|)$. However, checking that $\kappa(|G|)$ is below a given threshold $\bar{\kappa}_{TG}$ is relatively cheap. It is sufficient to check that:

$$\bar{\kappa}_{TG}\boldsymbol{A}_G - \boldsymbol{M}_G\left(\boldsymbol{I} - \boldsymbol{1}_G\left(\boldsymbol{1}_G^T\boldsymbol{M}_G\boldsymbol{1}_G\right)^{-1}\boldsymbol{1}_G^T\boldsymbol{M}_G\right) \tag{5.65}$$

is nonnegative definite. This can be done in $O(|G|^3)$ operation by verifying that the Cholesky factorization exists, i.e., there is no negative pivot. Therefore, $\kappa(|G|)$ does not need to be computed explicitly to be certain that $\kappa(|G|) \leq \bar{\kappa}_{TG}$. The first pairwise coarsening step is given by:

1. Create the set G_0, i.e., create the set of variables which will not be aggregated.

2. Choose an unknown and find among its unassigned neighbors the one that gives the smallest $\kappa(\{i,j\})$.

3. Check that $\kappa(\{i,j\}) \leq \bar{\kappa}_{TG}$. If the condition is not verified, the variable is left unassociated on the coarse grid.

To increase the size of the aggregates, the temporary coarse grid matrix $\tilde{\boldsymbol{A}}_c$ is computed and the same process we just described is applied. The set G_0 cannot be changed and the quality factor $\tilde{\kappa}(\{i,j\})$ needs to be adapted to reflect the quality of the corresponding aggregate $\kappa(G_i \cup G_j)$ in the original matrix. Therefore, the definition of s_j is slightly modified:

$$\tilde{s}_i = -\sum_{k \in G_i} \sum_{j \in G_i} a_{kj}. \tag{5.66}$$

This change exists to ensure that $\tilde{\kappa}(\{i,j\})$ is a lower bound of $\kappa(G_i \cup G_j)$. Thus, if $\tilde{\kappa}(\{i,j\}) \geq \bar{\kappa}_{TG}$, the pair has to be rejected because it is impossible for $\kappa(G_i \cup G_j)$ to satisfy the condition. A unique characteristic of this coarsening method is that you can, in theory, have an arbitrary number of pairwise coarsening passes without degrading the upper bound of the condition number. In practice, however, the coarsening is stopped if either a given number of passes has been done or the coarsening factor has reached a target value. To conclude the explanation of the coarsening step, we explain how unknowns are picked and how to pick between a pair $\{i,j\}$ and another $\{i,k\}$ if they have the same quality factor. If there is no priority rules, the coarsening would depend of the ordering of the variables or the way off-diagonal entries are stored. In AGMG, the rule chosen tries to increase the regularity of the aggregates because in practice, this increases the coarsening speed of the coarser levels. Even if the coarsening step tries to create regular aggregates on regular grids, the results are still quite good for unstructured grids [74]. The priority rule consists of using a Cuthill-McKee permutation [41] to renumber the variables and using the number associated with the variable as a priority number (the lower number has the priority). The Cuthill-McKee permutation works as follows: the number one is given to a node with minimal degree; the next numbers are given to its neighbors ordered by increasing degree; then their neighbors are given a number by increasing degree. The process is over when all nodes are numbered. There are still some uncertainties in the numbering if there are several variables with minimal degree or when several neighbors of a variables have the same degree. However, these choices do not affect performance [74].

AGMG stops the coarsening when the number of variables is less or equal to 400.

D Results

In this section, we show two Fourier analyses of MIP: one where the S_n order is varied and one where the aspect ratio is varied. We also compare different methods to solve MIP: congugate gradient (CG), conjugate gradient preconditioned with symmetric Gauss-Seidel (PCG-SGS), conjugate gradient preconditioned with ML using uncoupled aggregation (PCG-MLU), conjugate gradient preconditioned with ML using MIS aggregation (PCG-MLM), and AGMG. The options used for ML can be found in the Appendix. Unless otherwise specified, PWLD finite elements are used in this section.

1 Fourier Analyses

Analysing Source Iteration accelerated with DSA is often performed using Fourier analysis [61, 115]. When a Fourier analysis is performed, the error is decomposed into different modes and by inspecting the damping of the different error modes, the effectiveness of the DSA scheme can be studied. The largest damping factor is the spectral radius of the method. The smaller the spectral radius is, the faster the scheme converges. If the spectral radius is greater than one, the method is unstable.

a *S_n order varied*

This Fourier analysis was carried on a square cell, using a Gauss-Legendre-Chebyshev (GLC) quadrature. The medium is homogeneous, the scattering ratio $c = 0.9999$ and periodic boundary conditions are used. The $x-$axis is the mesh size in mean free path and the $y-$axis is the spectral radius. On Figure V.4, there are four curves corresponding to different S_n order: S_2, S_4, S_8, and S_{16}.

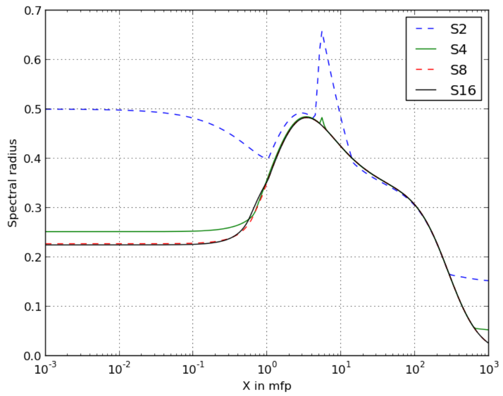


Figure V.4: Fourier analysis as a function of the mesh optical thickness, homogeneous infinite medium case

MIP is stable for every cell size. The spectral radius is always less than 0.5, except for S_2 where it peaks at about 0.7.

b *Aspect ratio varied*

For this Fourier analysis, we use a S_{16} GLC quadrature, a homogeneous medium, $c = 0.9999$ and periodic boundary conditions. The $x-$axis is the mesh size in mean free path in the x direction and the $y-$axis is the spectral radius. On Figure V.5, there are five curves corresponding to five different aspect ratio: $\frac{Y}{X} = \frac{1}{16}$, $\frac{Y}{X} = \frac{1}{4}$, $\frac{Y}{X} = 1$, $\frac{Y}{X} = 4$, $\frac{Y}{X} = 16$, and $\frac{Y}{X} = 100$.

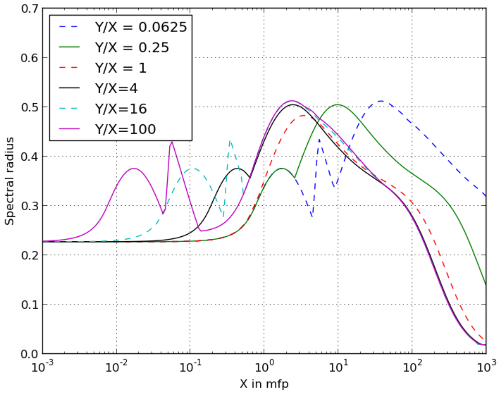


Figure V.5: Fourier analysis as a function of the mesh optical thickness, homogeneous infinite medium case for different aspect ratios

MIP is stable for every aspect ratio and the maximum of the spectral radius peaks at about 0.5. However, we noted that when both c approaches one and the aspect ratio is large, MIP can become ill-conditioned. In Chapter IV, MIP was used for problems with $c = 1$, without any issues because the cells were square (aspect ratio is one).

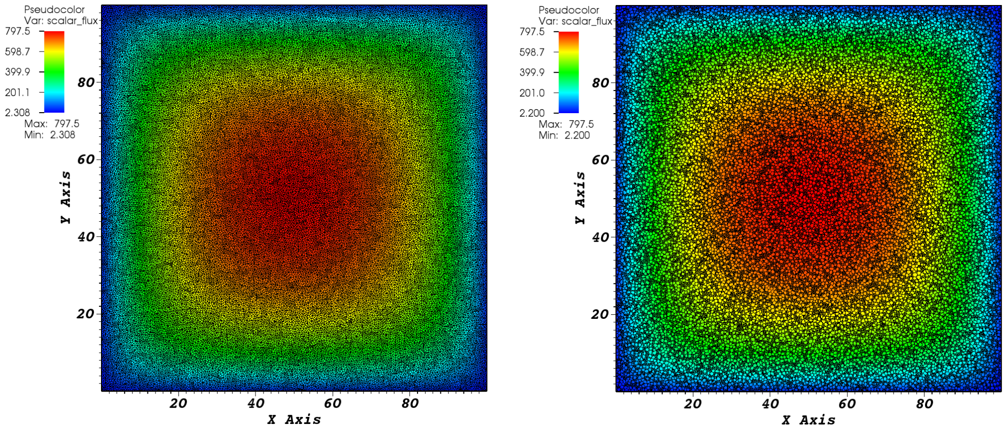
2 Homogeneous medium

Next, we compare different solvers for MIP on a homogeneous medium, 100cm × 100cm, $\Sigma_t = 1\text{cm}^{-1}$ and $\Sigma_s = 0.999\text{cm}^{-1}$, with vacuum boundary conditions and a source of intensity $1\text{cm}^{-3}\text{s}^{-1}$. We use a S_8 Gauss-Legendre-Chebyshev quadrature, a Source Iteration solver with relative tolerance of 10^{-8} and a relative tolerance for MIP of 10^{-10}.

Quadrilateral cells: the mesh is composed of 49236 quadrilateral cells that corresponds to 197052 degrees of freedom.

Polygonal cells: the mesh is composed of 45204 triangles, 823 quadrilaterals, 4978 pentagons, 4155 hexagons, 725 heptagons, and 24 octagons, for a total of 55909 cells and 193991 degrees of freedom. This example will allow us to test MIP and the different preconditioners on a mesh composed of different types of cell.

The meshes and the solutions of these two problems are given on Figure V.6:



(a) Quadrilateral cells

(b) Polygonal cells

Figure V.6: Meshes and scalar fluxes

In Table V.2, the different solvers, used on the quadrilateral cells, are compared:

Table V.2: Comparison of different preconditioners for quadrilateral cells

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
SI iter	7311	24	24	24	24	24
Prec (s)	NA	NA	0.171358	1.8255	9.56078	0.332
MIP (s)	NA	1095.7	1311.76	192.622	197.632	29.9727
CG iter	NA	56649	17332	630	604	578
Total (s)	39176.7	1264.98	1477.95	363.202	367.841	194.568

In this Table, SI iter is the number of Source Iteration iterations needed to solve the problem, Prec is the time in seconds needed to initialize the preconditioner used by CG, MIP is the total time in seconds spent solving DSA during the calculation, CG iter is the total number of CG iterations used to solve MIP, and Total is the time in seconds needed to solve the problem.

Using MIP decreases significantly the number of SI iterations and the calculation time as expected. Using PCG-SGS decreases by a factor of three of the number of CG iterations compared to CG but the time needed to solve MIP is greater. This is because each PCG-SGS iteration is much slower than one unpreconditioned CG iteration. SGS requires basically two triangular solves. It is unclear why these simple solves would be so costly in CPU time so as to actually increase the total solver time while the number of CG iterations has been divided by three. With ML, the number of CG iterations is reduced by a factor of 50 and the MIP calculation time is reduced by a factor three compared to CG. AGMG is by far the most efficient solver, the number of CG iterations is slightly lower than PCG-ML but the MIP calculation is 20 times faster than CG.

The different solvers, used on the polygonal cells, are compared in Table V.3:

Table V.3: Comparison of different preconditioners for polygonal cells using SI

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
SI iter	7311	23	23	23	23	23
Prec (s)	NA	NA	0.06388	1.73379	8.0426	0.388
MIP (s)	NA	877.861	1263.31	198.63	191.989	31.242
CG iter	NA	46262	16712	652	603	555
Total (s)	42666.7	1060.53	1447.53	382.275	384.422	216.946

We see that using different types of cells in the same mesh does not affect the performance of MIP or of the preconditioners.

In the next test, the problem is exactly the same as the previous one using polygonal cells but the SI solver is replaced by GMRES. The comparison is done in Table V.4:

Table V.4: Comparison of different preconditioners for polygonal cells using GMRES

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
GMRES iter	266	12	12	12	12	12
Prec (s)	NA	NA	0.0675611	1.56115	7.89327	0.0331
MIP (s)	NA	546.56	770.244	126.723	120.68	22.3754
CG iter	NA	28653	10274	407	390	351
Total (s)	1549.17	675.319	898.149	261.121	261.937	162.47

The conclusions are the same as in the SI case. The performance of the preconditioners is not affected by the change of solver.

3 Heterogeneous medium

In this example, we use a heterogeneous medium composed of 184 triangles, 3720 quadrilaterals and 2791 regular hexagons of side $0.05cm$ for a total of 6695 cells and 32178 degrees of freedom. The domain is $5.28275cm$ by $4.6cm$. Reflective boundary conditions are used. The quadrature is a S_{16} Gauss-Legendre-Chebyshev quadrature. The SI solver has a relative tolerance of 10^{-8} and the relative tolerance for MIP is 10^{-10}. The domain is composed of three zones (see Figure V.7):

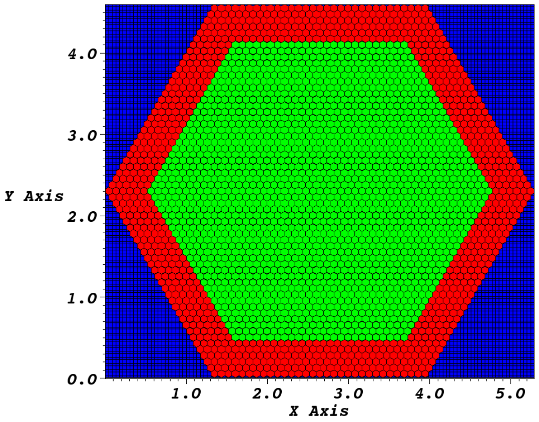


Figure V.7: Zones of the domain discretized by triangles, rectangles, and hexagons

The properties of the different zones are:

Green zone: $\Sigma_t = 1.5\text{cm}^{-1}$, $\Sigma_s = 1.44\text{cm}^{-1}$, source$= 1\text{cm}^{-3}\text{s}^{-1}$

Red zone: $\Sigma_t = 1\text{cm}^{-1}$, $\Sigma_s = 0.9\text{cm}^{-1}$, no source

Blue zone: $\Sigma_t = 1\text{cm}^{-1}$, $\Sigma_s = 0.3\text{cm}^{-1}$, no source

The different solvers are compared in Table V.5:

Table V.5: Comparison of different preconditioners for a heterogeneous medium

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
SI iter	122	18	18	18	18	18
Prec (s)	NA	NA	0.016149	0.336215	1.36803	0.065
MIP (s)	NA	60.2031	123.05	31.7048	30.8669	2.80108
CG iter	NA	12016	6764	423	391	248
Total (s)	413.274	131.297	188.586	101.888	103.734	71.5392

We can see that the comments made for the homogeneous tests are still valid. MIP is effective even with heterogeneous medium and AGMG is still the fastest method. It is interesting to note that, contrary to the homogeneous tests where the number of CG iterations remained similar for all algebraic multigrid preconditioners, for this heterogeneous test AGMG requires significantly fewer iterations than PCG-MLU and PCG-MLM. This difference may be due to the fact that ML was first designed to be used for continuous finite elements discretization and that we are using discontinuous finite elements.

The cross sections of the different zones were taken from [105]. In the next test, they are modified to make the problem more challenging:

Green zone: $\Sigma_t = 1.5\text{cm}^{-1}$, $\Sigma_s = 1.499\text{cm}^{-1}$, source= $1\text{cm}^{-3}\text{s}^{-1}$

Red zone: $\Sigma_t = 1\text{cm}^{-1}$, $\Sigma_s = 0.999\text{cm}^{-1}$, no source

Blue zone: $\Sigma_t = 1\text{cm}^{-1}$, $\Sigma_s = 0.3\text{cm}^{-1}$, no source

The different solvers are compared in Table V.6:

Table V.6: Comparison of different preconditioners for a highly diffusive heterogeneous medium

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
SI iter	278	17	17	17	17	17
Prec (s)	NA	NA	0.0160661	0.368768	1.41632	0.07
MIP (s)	NA	58.422	126.93	33.2225	31.3045	2.924
CG iter	NA	12214	6679	415	386	248
Total (s)	910.566	120.889	190.413	99.7524	97.4666	70.6424

Making the problem more diffusive increases the interest of using DSA but it does not change the behavior of the preconditioners.

4 AMR mesh

In this example from [105], the domain is $10cm \times 10cm$. The left and bottom boundaries are reflective whereas the right and the top boundaries are vacuum. There are 10720 cells: 10482 quadrilaterals, 236 pentagons, and 2 hexagons for a total of 43120 degrees of freedom. As in the previous example, the domain is composed of three zones (see Figure V.8):

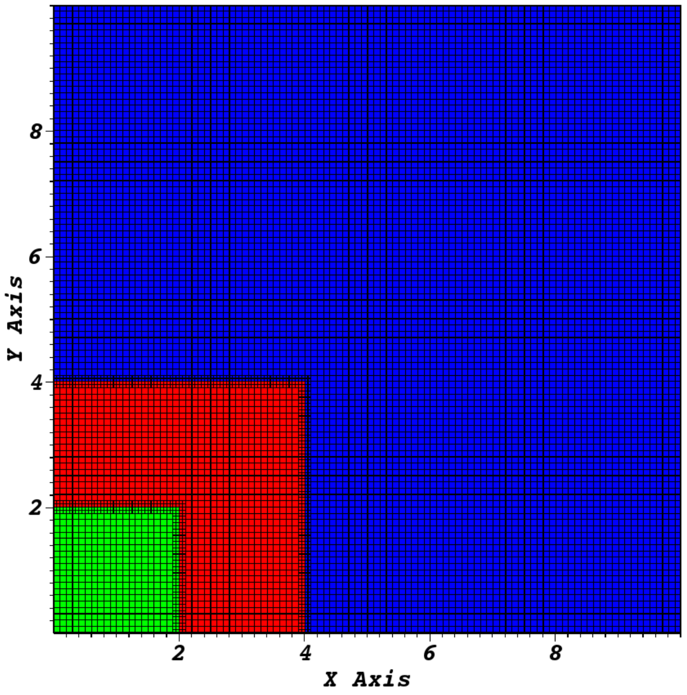


Figure V.8: Zones of the AMR mesh

where:

Green zone: $\Sigma_t = 1.5cm^{-1}$, $\Sigma_s = 1.44cm^{-1}$, source=$1cm^{-3}s^{-1}$

Red zone: $\Sigma_t = 1cm^{-1}$, $\Sigma_s = 0.9cm^{-1}$, no source

Blue zone: $\Sigma_t = 1cm^{-1}$, $\Sigma_s = 0.3cm^{-1}$, no source

The distribution of cells is given on Figure V.9:

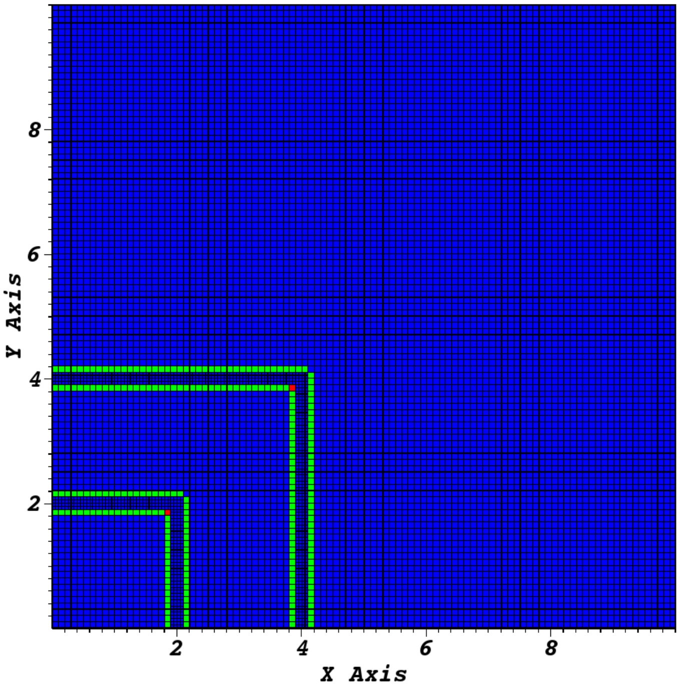


Figure V.9: Polygons distribution

where:

Blue cells are quadrilaterals.

Green cells are pentagons.

Red cells are hexagons.

This mesh is typical of a mesh obtained after one level of adaptive mesh refinement (the cells at the interface of different zones have been refined once). We see that instead of introducing hanging nodes, we have introduced pentagons and hexagons in the mesh. A S_{16} GLC quadrature is employed. The tolerance on SI is 10^{-8} and the tolerance on the CG solvers is 10^{-10}. The different solvers are compared in Table V.7:

Table V.7: Comparison of preconditioners on an AMR mesh

	No-DSA	CG	PCG-SGS	PCG-MLU	PCG-MLM	AGMG
SI iter	184	19	19	19	19	19
Prec (s)	NA	NA	0.043463	0.358002	1.19301	0.0111
MIP (s)	NA	48.1908	81.0992	25.2699	25.0699	2.56198
CG iter	NA	11300	4734	361	361	264
Total (s)	802.985	138.825	172.423	116.018	116.517	94.1963

As expected, the results are similar to our previous test.

5 Rectangular cells

As mentioned previously in this Chapter, AMG can have difficulties when the aspect ratio of rectangular cells is high. Moreover, when the aspect ratio is high and the scattering ratio is close to one, MIP becomes ill conditioned. In the next four examples, the domain is square $100cm \times 100cm$ with vacuum boundaries. There are 10000 cells and we use BLD finite elements for the first three runs and PWLD finite elements for the last one; there are 40000 degrees of freedom. The relative tolerance on SI is 10^{-8} and the relative tolerance on CG is 10^{-10}. We use a S_8 GLC quadrature, $\Sigma_t = 1cm^{-1}$, and $\Sigma_s = 0.999cm^{-1}$. The source is $1n/(cm^3s)$. In the first test, the domain is discretized by 100 cells along x and 100 cells along y. In the second test, the domain is discretized by 250 cells along x and 40 cells along y. Therefore, the aspect ratio is 6.25 for the second test. In the last two tests, the domain is discretized by 1000 cells along x and 10 cells along y (the aspect ratio is

100).

We also compared the effect of the size of the coarsest grid on the convergence. In the following Tables (Table V.8, Table V.9, Table V.10, and Table V.11), we compare No-DSA, CG, and AGMG defined previously with:

PCG-MLU-D: conjugate gradient preconditioned with ML using uncoupled coarsening with a coarsest grid of size less or equal of 128 (default value).

PCG-MLU-M: conjugate gradient preconditioned with ML using uncoupled coarsening with a coarsest grid of size less or equal of 400 (same value than AGMG).

PCG-MLM-D: conjugate gradient preconditioned with ML using MIS coarsening with a coarsest grid of size less or equal of 128 (default value).

PCG-MLM-M: conjugate gradient preconditioned with ML using MIS coarsening with a coarsest grid of size less or equal of 400 (same value than AGMG).

Table V.8: Comparison of preconditioners on rectangular mesh with an aspect ratio of 1.0 with BLD finite elements

	No-DSA	CG	PCG-SGS	PCG-MLU-D	PCG-MLU-M	PCG-MLM-D	PCG-MLM-M	AGMG
SI iter	7311	21	21	21	21	21	21	21
Prec (s)	NA	NA	0.0167298	0.345833	0.392942	1.06411	1.2699	0.057
MIP (s)	NA	39.4564	90.393	23.9443	28.3395	23.5808	28.5564	2.76611
CG iter	NA	8584	4911	377	377	376	376	265
Total (s)	8254.46	66.8963	119.492	48.8737	56.932	48.5779	59.0295	31.6606

Table V.9: Comparison of preconditioners on rectangular mesh with an aspect ratio of 6.25 with BLD finite elements

	No-DSA	CG	PCG-SGS	PCG-MLU-D	PCG-MLU-M	PCG-MLM-D	PCG-MLM-M	AGMG
SI iter	7311	23	23	23	23	23	23	23
Prec (s)	NA	NA	0.0205462	0.32853	0.381816	1.02377	1.111999	0.671
MIP (s)	NA	87.8195	178.535	74.1533	89.1188	73.0178	80.5592	4.20011
CG iter	NA	18985	9820	1219	1219	1206	1206	171
Total (s)	7311.49	117.768	210.944	100.424	120.757	101.374	110.913	32.857

Table V.10: Comparison of preconditioners on rectangular mesh with an aspect ratio of 100 with BLD finite elements

	No-DSA	CG	PCG-SGS	PCG-MLU-D	PCG-MLU-M	PCG-MLM-D	PCG-MLM-M	AGMG
SI iter	7304	24	24	24	24	24	24	24
Prec (s)	NA	NA	0.049191	0.651429	0.451487	1.15778	1.1675	0.383
MIP (s)	NA	361.306	656.815	945.56	986.592	917.151	917.481	3.74259
CG iter	NA	82613	38263	14128	14215	13810	13824	177
Total (s)	8581.25	400.116	698.613	984.163	1027.1	956.392	957.611	43.6815

Table V.11: Comparison of preconditioners on rectangular mesh with an aspect ratio of 100 with PWLD finite elements

	No-DSA	CG	PCG-SGS	PCG-MLU-D	PCG-MLU-M	PCG-MLM-D	PCG-MLM-M	AGMG
SI iter	7304	24	24	24	24	24	24	24
Prec (s)	NA	NA	0.0164239	0.362463	0.778893	1.03128	1.66949	0.052
MIP (s)	NA	372.227	742.902	941.06	925.247	922.258	964.044	6.93176
CG iter	NA	84802	43466	14180	14179	13896	13894	821
Total (s)	9035.6	414.301	784.77	985.796	969.318	966.77	1010.44	44.7032

As predicted, inverting MIP requires a lot more CG iterations when the aspect ratio increases. The only exception is for AGMG which requires fewer iterations when the aspect ratio is 6.25 and 100 than when it is 1.0. As expected, PCG-MLU and PCG-MLM are much more affected by the aspect ratio than CG and PCG-SGS. The number of CG iterations for CG and PCG-SGS is multiplied by two when the aspect ratio is increased from 1.0 to 6.25, whereas it is multiplied by a little more than three for PCG-MLU and PCG-MLM. When the aspect ratio is 100, PCG-MLU and PCG-MLM are the slowest methods. This is not totally unexpected since these examples have been designed to test the limitations of algebraic multigrid preconditioners. It is interesting to note that changing the size of the coarsest grid for PCG-MLU and PCG-MLM does not affect the number of CG iterations. Even if AGMG does not seems to be to affected by the change in the aspect ratio when BLD finite elements are employed, using PWLD finite elements dramatically increases the number of CG iterations when the aspect ratio is 100. AGMG is the only method which is significantly impacted by the change of finite element type; however, it stays by far the fastest method to solve the MIP equations.

E Conclusions

We have adapted the MIP-DSA to PieceWise Linear Discontinuous finite elements and proposed a simple way to compute the penalty coefficient which enables the use of MIP on arbitrary polygonal meshes. The advantage of polygonal cells is the potential reduction of the numbers of unknowns and the possibility to use adaptive mesh refinement without having hanging nodes. We have performed two Fourier analyses of the new MIP-DSA on a rectangular cell and shown that MIP is always stable when the scattering is isotropic. On different examples, we tested MIP on highly unstructured meshed composed of different types of cells. We noticed that the efficiency of MIP does not seem to be degraded on these meshes. We have also compared different preconditioners for CG to solve MIP. Algebraic multigrid methods were found to be the best preconditioner, AGMG being up to 20 times faster than CG without preconditioning which itself was faster than CG preconditioned by SSOR.

CHAPTER VI

JANUS

A Introduction

In this Chapter, we detail the implementation of the transport code developed in this research, Janus. Janus is a two-dimensional one-group S_n transport solver. It uses arbitrary polygonal meshes and implements an angular multigrid preconditioner for highly anisotropic scattering. The ASCII output file generated by Janus can be converted to a silo file [11], using an other C++ code, Apollo. This output file can be read by VisIt [10]. A python code, Diana, can be used to generate the mesh or to convert a mesh generated by Triangle [9] into a mesh readable by Janus. Another python code, Mercury, can be used to generate an input files for Janus. Mercury can help writing an input file by checking that all the data required by Janus are present and that they are written in the right order.

Janus is documented using Doxygen [4]. It is built upon Trilinos 10.4 [8] and GSL (GNU Scientific Library) [7] and uses Autoconf, Automake, and Autotest [1, 2]. Git [6] is used for revision control.

Janus, Diana, and Mercury can be cloned at *git://gitorious.org/transport/janus.git*

Apollo can be cloned at *git://gitorious.org/transport/plot.git*

B Implementation

Janus is composed of the following classes:

PARAMETERS: In this class, the different parameters such as, the type of solver for transport equation (SI or a Krylov method), the type of solver for MIP, the convergence tolerance, the boundary conditions, the intensity of the sources, the cross sections, etc, are read from an input file and stored. If Fokker-Planck cross sections are used, they are computed here. The different cross sections for the angular multigrid are computed by this class and the extended transport correction is applied.

TRIANGULATION: In this class, the geometry, the material IDs, and the source IDs are read. Two different input files can be read. When the mesh uses rectangular cells, the abscissae then the ordinates have to be given in order of increasing value. After that, the materials IDs and the source IDs are read. For instance, if the domain, $1cm \times 1cm$, is decomposed in four identical cells, the input file looks like:

```
rectangle
2 2      // (number of x-divisions, number of y-divisions)
0. 0.5 1.0
0. 0.5 1.0
0 0 0 0  // (material IDs)
0 0 0 0  // (source IDs)
```

The other acceptable type of input file is used for polygonal cells. In that case, the number of edges of a cell is given first, followed by the coordinates of each vertex, the material IDs and finally the source IDs. The vertices must be ordered in an anti-clockwise order but there is no need to order the cell or for two successive cells to be adjacent in the mesh. For instance, a possible input file would be:

```
polygon
```

```
4   // (number of cells)
3 1. 0. 1. 1. 0.5 0.5 0 0
5 0. 0. 0.5 0. 0.5 0.5 0.5 1. 0. 1. 0 0
3 0.5 0. 1. 0. 0.5 0.5 0 0
3 0.5 0.5 1. 1. 0.5 1. 0 0
```

This class assumes that the domain is rectangular. After reading the geometry, the EDGE objects are created. Before an edge is created, it must be checked that the edge does not already exist. To do so, the coordinates of the two vertices of the candidate edge are compared with the coordinates of the vertices of a subset of the existing edge. This subset corresponds to the smallest subset of edges having an abscissa, respectively an ordinate, of one of their vertices equals to an abscissa, respectively an ordinate, of one of the vertices of the candidate edge.

EDGE: This class contains the coordinates of the vertices associated with the edge, the global and local IDs of the edge, the IDs of the cells associated with the edge, the type of cell (interior or boundary and the type of boundary: vacuum, isotropic or most normal direction of the quadrature), the two normal vectors associated with the two cells, the length of the edge, etc.

FINITE_ELEMENT: This class is the base class for BLD and PWLD. It contains all the matrices needed for the DFE discretization of the transport equation and of the diffusion equation.

BLD: This class derives from FINITE_ELEMENT and builds the bilinear finite elements.

PWLD: This class derives from FINITE_ELEMENT and builds the piecewise linear finite elements.

QUADRATURE: This class is the base class for both the GLC and LS classes. QUADRATURE builds the discrete-to-moment and moment-to-discrete matrices and stores the different directions used by the quadrature. The directions on the first octant are computed in GLC and LS. Then, they are deployed over the other octants. After that, the spherical harmonics are computed and evaluated at the given directions. When a Galerkin quadrature is used, selection rules are employed and the discrete-to-moment matrix is computed by inverting the moment-to-discrete matrix. Otherwise, the discrete-to-moment matrix is obtained by transposing the moment-to-discrete matrix and by multiplying it by the weights of the quadrature.

GLC: This class derives from QUADRATURE and computes the weights and the directions used by the Gauss-Legendre-Chebyshev triangular quadrature.

LS: This class derives from QUADRATURE and computes the weights and the directions used by the Level-Symmetric quadrature up to S_{24}.

CELL: This class stores the ID of a cell, a vector of pointers to the edges of a cell, the intensity of the source in the cell, the material properties (Σ_s, Σ_t, and the diffusion coefficient), the FINITE_ELEMENT associated with the cell, etc. The orthogonal lengths are calculated by this class.

DOF_HANDLER: This class builds the *mesh* by creating all the CELL objects and the FINITE_ELEMENT objects associated with them. It is the DOF_HANDLER object that computes the sweep ordering for all directions. First, the edges on the boundary with a known incoming flux are put in a list, *incoming_edges*.

The sweep ordering will continue as long as this list is not empty. The first edge in the list is popped and the associated cell, which has not been accepted in the sweep order, is found. Then, we loop over the edges of the cell to determine which ones are associated with an outgoing flux and which ones are associated with an incoming flux. The cell is accepted if all the edges which are not "outgoing" are in *incoming_edges*. If the cell is rejected, the edge is pushed back at the end of the list. If the cell is accepted, the edges of the cell which are incoming are removed from *incoming_edges*. The others edges are pushed back at the end of the list except if they are on the boundary of the domain.

TRANSPORT_OPERATOR: The calculation is performed in this class. It derives from the Epetra_Operator of Trilinos. TRANSPORT_OPERATOR handles the angular multigrid by calling itself recursively and restricting and projecting the flux moments on the different "angular" grids. It is also in this class that the scattering source is computed and the sweeps are performed.

MIP: This class builds and solves MIP. The first time that *Solve()* is called, the left hand-side is built and stored using a compressed row storage format (CRS). Then, the problem is solved by CG, PCG-SSOR, PCG-MLU, PCG-MLM, or AGMG. If AGMG is employed, there is an extra step to convert the right hand-side to a Fortran data type and the result back to an Epetra_MultiVector.

TRANSPORT_SOLVER: This class builds the PARAMETERS object, the TRIANGULATION object, the QUADRATURE object(s), the DOF_HANDLER object, and the initial TRANSPORT_OPERATOR object. SI and the Krylov solvers are called in *Solve()*. The final result and the mesh are written in a file by this class.

EXCEPTION: This class handles the exceptions that can be thrown.

C Verifications

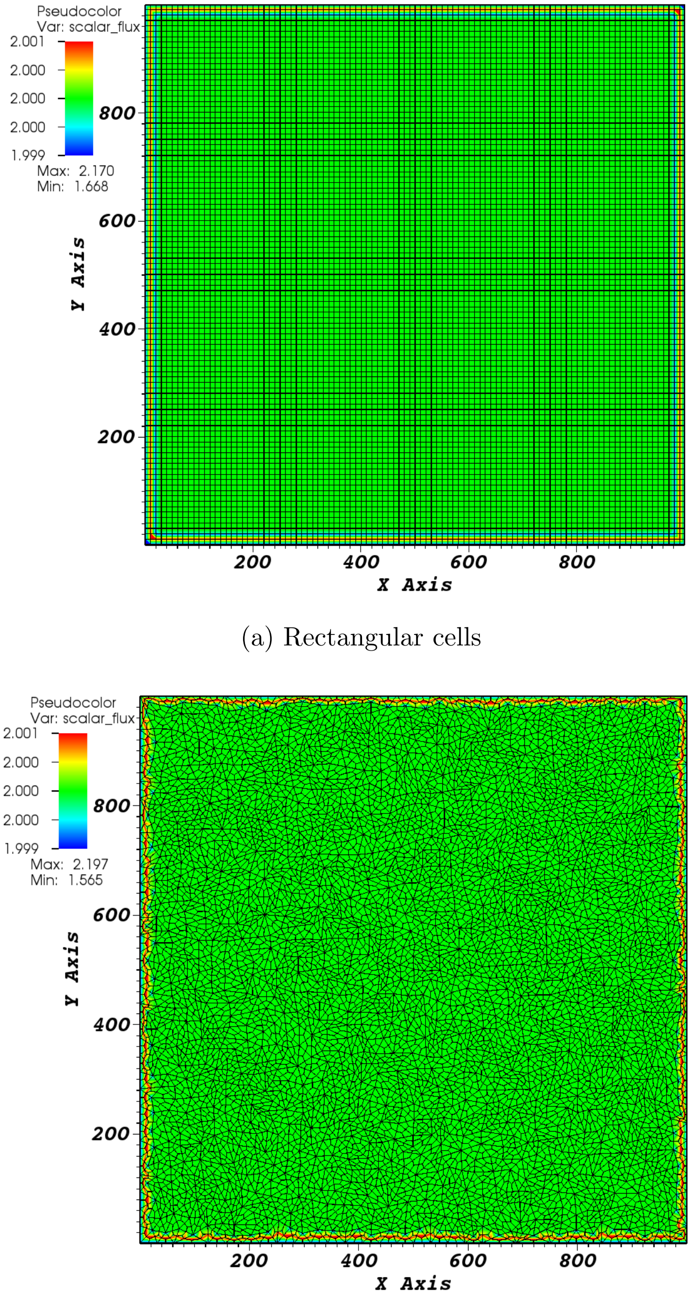
The verification of the code was done through unit testing, comparisons of results with existing codes, and using known solutions of problems. Next, we show two of the tests that were done to check the code.

1 Infinite medium

When the domain is infinite and the medium is homogeneous. The isotropic transport equation reduces to:

$$\phi = \frac{Q}{\Sigma_a}. \tag{6.1}$$

To approximate the infinite medium, we choose a very large total cross section such that the mean free path of the particles is very small compared to the size of the domain. In the following test, the domain is $1000cm \times 1000cm$, $Q = 2n/(cm^3s)$, $\Sigma_t = 10cm^{-1}$ and $\Sigma_s = 9cm^{-1}$. We use vacuum boundary conditions and a S_8 GLC quadrature. We show two tests: one which uses an uniform mesh of 100 by 100 cells and BLD finite elements and the other which uses an unstructured mesh of 9972 quadrilaterals and PWLD finite elements. Given equation (6.1), the scalar flux should be equal to two. In Figures VI.1a and VI.1b, the scalar flux less than or equal to 1.999 is in blue and the scalar flux greater than or equal to 2.001 is in red:



(a) Rectangular cells

(b) Quadrilateral cells

Figure VI.1: Scalar flux

2 Convergence order

In this test, we check the convergence order of PWLD and BLD on a benchmark. We plot the error on average scalar fluxes as a function of the number of degrees of freedom. Since for two-dimensional geometries, the number of degrees of freedom is proportional to the square of the typical element size, the slopes of the graphs should equal one (PWLD and BLD are both second order methods). The test that we chose is the IAEA EIR-2 benchmark problem [54]. This benchmark consists of five regions:

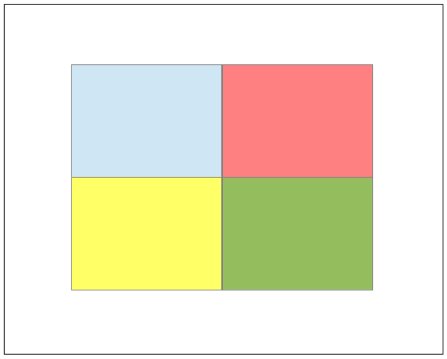


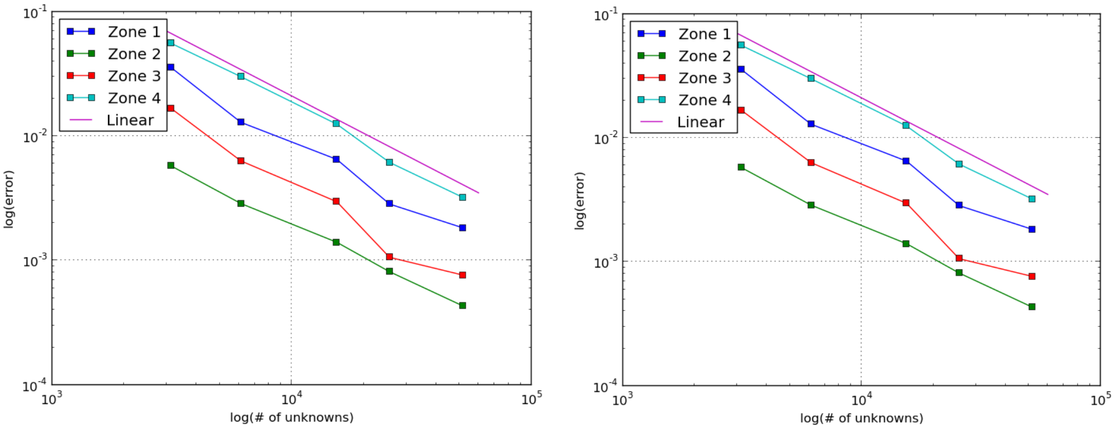
Figure VI.2: Zones of the IAEA EIR-2 benchmark problem

The properties of the different zones are given in Table VI.1:

Table VI.1: Properties of the different zones of the benchmark

Zone	White	Blue	Salmon	Yellow	Green
Source $(n/(cm^3s))$	0	0	1	1	0
Σ_t (cm^{-1})	0.9	0.65	0.7	0.6	0.48
Σ_s (cm^{-1})	0.89	0.5	0.66	0.53	0.2
length (cm)	96	30	30	30	30
height (cm)	86	25	25	25	25

The colored zones are in the middle of the white zone. In Figures VI.3a and VI.3b, we show the convergence of the average scalar flux in the different zones for S_8 Gauss-Legendre-Chebyshev quadrature when BLD and PWLD finite elements are used.



(a) BLD finite elements

(b) PWLD finite elements

Figure VI.3: Convergence of BLD and PWLD

We can see that the curves in all the zones have the right slope, i.e., the right order of convergence.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

A Conclusions

In this dissertation, we first recalled the development of the Boltzmann-Fokker-Planck equation and the limitations of the Fokker-Planck operator. The Boltzmann-Fokker-Planck equation was introduced for charged particle transport because the scattering kernel is highly forward-peaked such that a standard Legendre expansion of the scattering kernel is impractical and would require hundreds of terms. We also recalled a previous work from Pomraning showing that the Fokker-Planck operator is an asymptotic limit of the Boltzmann operator when the scattering is forward-peaked and that the energy transfer during a collision tends to zero. In the Boltzmann-Fokker-Planck equation, the Fokker-Planck operator is used to model the highly forward-peaked scattering collisions whereas the Boltzmann operator is used for the wide angle scattering collisions. The Fokker-Planck operator simplifies the calculation of the transport equation but is valid only if the kernel is peaked enough. For instance, realistic screened Rutherford cross sections are not peaked enough. Then, we introduced the Fokker-Planck cross sections which mimic the Fokker-Planck operator when using the Boltzmann operator. Since Fokker-Planck cross sections are the most forward-peaked cross section (the extended transport correction renders the delta scattering equivalent to no scattering at all), we used them for our tests. Finally, we introduced Galerkin quadratures. Galerkin quadratures are crucial to obtain correct results when the scattering is highly anisotropic. Galerkin quadratures are equivalent to standard quadratures when the scattering is weakly anisotropic but when the scattering is highly anisotropic, using a standard quadrature can introduce non physical solutions.

Next, we introduced the angular multigrid methods to speed up the calculation with highly anisotropic scattering. When the scattering is highly anisotropic, the Diffusion Synthetic Acceleration (DSA) is not effective anymore because it cannot speed up the convergence of high order flux moments. The initial work on this topic was carried out by Morel and Manteuffel. They developed a one-dimensional angular multigrid method to accelerate the convergence of Source Iteration (SI). This angular multigrid method uses an $S_{n/2}$ sweep to correct the S_n sweep. The $S_{n/2}$ correction is itself corrected by a $S_{n/4}$ correction, etc. until the S_4 correction is corrected by $P1$ equations. They showed that when Fokker-Planck cross sections are used the spectral radius of the new method is bounded by 0.6 whereas the spectral radius of DSA can become arbitrary close to unity. Pautz, et al., generalized the angular multigrid method to multidimensional geometries. In this case, the successive correction used an S_2 sweep before a DSA stage at the final level. Unfortunately, the generalized method was unstable. To stabilize it, the corrections need to be filtered with a diffusion operator. The effect of this diffusive filter is to remove the high frequencies of the correction. This method is better than straight SI+DSA but when the anisotropy increases, the spectral radius can again become arbitrary close to one. In this research, we recast the angular multigrid method for multidimensional geometries without filtering as a preconditioner for a Krylov subspace solver. The new method is always more effective and efficient than DSA and is more effective as the anisotropy increases but unlike the one dimensional method, the number of iterations does not saturate with increasing anisotropy.

We also extended the Modified Interior Penalty (MIP) DSA developed for triangular cells to arbitrary polygonal meshes using the PieceWise Linear Discontinuous

(PWLD) finite elements. Arbitrary polygonal meshes can potentially decrease the number of unknowns to mesh a domain and they simplifies adaptive mesh refinement by suppressing hanging nodes. MIP being SPD, is solved using CG. Therefore, we compared different CG preconditioners: Algebraic MultiGrid (AMG) preconditioners and the more common Symmetric Gauss-Seidel (SGS) preconditioner. AMG were shown to be much more efficient than SGS. Among the different AMG methods tested the AGMG code was the fastest and about 20 times faster than CG used without preconditioning.

To end this conclusion, we recall the goals that we defined in the proposal:

- reformulate the angular multigrid method as a preconditioner for Krylov solver for highly forward-peaked scattering.
- adapt MIP for quadrilateral and polygonal cells.
- test algebraic multigrid to solve efficiently MIP.

All these points have been treated in this dissertation.

B Future work

There are several ways to continue this work:

Extension for medical applications:

Modern radiotherapy uses Intensity Modulated Radiation Therapy (IMRT) as one of the methods to treat cancer. IMRT uses photons to ionize the water present in the cells to form free radicals which will damage the DNA of the cancerous cells causing them to die. IMRT allows to have several beams with different intensity profiles. To optimize the intensity profile, it is very common to divide the beams in small beamlets of constant intensity. In real applications, the number of beamlets is around a few thousands. The optimization of the position and the intensity of all these beamlets is a very complex problem and a lot of objective functions and constraints have been proposed [43, 50, 66, 93]. Due to the large number of variables, the number of dose calculations is very high and any increase in speed of these calculations can significantly decrease the time needed for the optimization to finish. To be able to compute the dose in a human body, the existing code needs to be extended to handle three dimensional geometries. BiLinear Discontinuous finite elements should be replaced by TriLinear Discontinuous finite elements. PWLD finite elements in 3D are described in [25]. The code should also be able to use the multigroup cross sections generated by CEPXS [3]. CEPXS is a code generating multigroup Coupled Electron-Photon cross sections (XS). I was developed to [3]:

- generate coupled electron-photon cross sections which can be used by standard discrete ordinates codes.
- model the same physical interactions as Version 2.1 of the Integrated-TIGER-Series (ITS) code package.

AMG for DSA on massively parallel computers:

AMG algorithms have been developed for massively parallel computers [28, 29] and for GPU [34]. While developing an AMG method for massively parallel computers, two steps of the algorithm must be designed carefully: the coarsening step and the smoothing step. In [98], the authors explore different coarsening methods including an uncoupled algorithm. This uncoupled coarsening method coarsens the grid without communication between processors. The problem is that the coarsening depends on the domain partitioning. For the partitioning to be independent of the domain partitioning, communications between the processors are required. Another delicate part of AMG is the smoothing step. It is very important that the smoother scales well. Some of the smoothers used in today's supercomputers will not scale properly with the next generation of supercomputers [29]. Given all these conditions, it will be important to verify that AMG preconditioners are still the most effective on massively parallel computers.

Convergence study of AGMG for MIP and development of AMG for MIP:

A more theoretical study of AGMG, which is the most effective preconditioner, is needed. The convergence properties of AGMG have been studied for non-singular symmetric M-matrices with non-negative row sum. MIP produces SPD matrices but they are not M-matrices due to presence of positive off-diagonal entries. Therefore, there is no theoretical background for the convergence MIP using AGMG which is, in this case, a heuristic method. Studying the convergence of AGMG for MIP could lead in a new AMG algorithm or an adaptation of AGMG for MIP.

Comparison of different AMG methods:

It would be of great interest to compare more AMG schemes, for instance using the ones of hypre [13]. In this research, most of the parameters kept their default values but a more extensive study of the effectiveness of AMG would require tuning each method. The choice of the DSA is very important because the choice of the discretization has a huge impact on the properties of the discretized system. Most of the theory for AMG algorithms has been developed for M-matrices and thus, it might be interesting to derive a DSA technique that produces such matrices.

REFERENCES

[1] Autoconf - GNU Project - Free Software Foundation (FSF). http://www.gnu.org/software/autoconf. [Online; accessed 29-September-2012].

[2] Automake - GNU Project - Free Software Foundation (FSF). http://www.gnu.org/software/automake. [Online; accessed 29-September-2012].

[3] CEPXS/ONELD 1.0 manual user. *RSICC Computer Code Collection*.

[4] Doxygen Manual. http://www.stack.nl/~dimitri/doxygen. [Online; accessed 29-September-2012].

[5] Fluent 6.3 user's guide.

[6] git–everything-is-local. http://git-scm.com. [Online; accessed 29-September-2012].

[7] GSL - GNU Scientific Library. http://www.gnu.org/software/gsl. [Online; accessed 29-September-2012].

[8] The Trilinos Project. http://www.trilinos.sandia.gov. [Online; accessed 29-September-2012].

[9] Triangle A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator. http://www.cs.cmu.edu/~quake/triangle.html. [Online; accessed 29-September-2012].

[10] VisIt. https://wci.llnl.gov/codes/visit. [Online; accessed 29-September-2012].

[11] Welcome to Silo A mesh and field I/O library and scientific database. `https://wci.llnl.gov/codes/silo`. [Online; accessed 29-September-2012].

[12] Discrete Ordinates Approximations to the First- and Second-Order Radiation Transport Equations. Technical report, Sandia National Laboratories, 2002.

[13] User's Manual hypre high performance preconditioners. Technical report, Lawrence Livermore National Laboratory, 2011.

[14] ANSY Fluent, 2012.

[15] CGAL Computational Geometry Algorithms Library, 2012.

[16] MSTK/MeshTools, 2012.

[17] Openfoam the open course cfd toolbox, 2012.

[18] Marvin L. Adams. Discontinuous Finite Element Transport Solutions in Thick Diffusive Problems. *Nuclear Science and Engineering*, 137:298–333, March 2001.

[19] Marvin L. Adams and Edward W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40:3–159, 2002.

[20] Marvin L. Adams and Todd A. Wareing. Diffusion-synthetic acceleration given anisotropic scattering, general quadratures, and multidimensions. *Conference: American Nuclear Society(ANS) annual meeting, San Diego*, 68, June 1993.

[21] M.L. Adams and W.R. Martin. A Method for Synthetically Accelerating Discontinuous Finite Element Transport Calculations. *Transactions of the American Nuclear Society*, 74:159–167, 1987.

[22] Anders Ahnesjo, Mikael Saxner, and Avo Trepp. A pencil beam model for photon dose calculation. *Medical Physics*, 19(2):263–273, 1992.

[23] Douglas N. Arnold, Arup Mukherjee, and Luc Pouly. Locally Adapted Tetrahedral Meshes Using Bisection. *SIAM Journal on Scientific Computing*, 22(2):421–448, 2000.

[24] O. Axelsson and P.S. Vassilevski. A Black-Box Generalized Conjugate Gradients Solver with Inner Iterations and Variable-Step Preconditioning. *SIAM Jounral on Matrix Analysis on Applications*, 12:625–644, 1991.

[25] Teresa S. Bailey. *The piecewise linear discontinuous finite element method applied to the RZ and XYZ transport equations.* PhD thesis, Texas A&M University, 2008.

[26] Teresa S. Bailey, Marvin L. Adams, Brian Yang, and Michael R. Zika. A piecewise linear finite element discretization of the diffusion equation for arbitrary polyhedral grids. *Journal of Computational Physics*, 227(8):3738–3757, 2008.

[27] Teresa S. Bailey, James S. Warsa, Jae H. Chang, and Marvin L. Adams. A Piecewise Bi-Linear Discontinuous Finite Element Spatial Discretization of the Sn Transport Equation. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Rio de Janeiro, Brazil*, May 8-12 2011.

[28] Allison Baker, Martin Schulz, and Ulrike Yang. On the Performance of an Algerbraic Multigrid Solver on Multicore Clusters. In Jose Palma, Michale Dayde, Osni Marques, and Joao Lopes, editors, *High Performance Computing for Computational Science - VECPAR 2010*, volume 6449, pages 102–115. Springer Berlin / Heidelberg, 2011.

[29] Allison H. Baker, Tzanio V. Kolev, and Ulrike Meier Yang. Multigrid Smoothers for Ultraparallel Computing. *SIAM Journal on Scientific Computing*, 33(5):2864–2887, 2011.

[30] Randal S. Baker. A Block Adaptive Mesh Refinement Algorithm for the Neutral Particle Transport Equation. *Nuclear Science and Engineering*, 141(1):1–12, May 2002.

[31] Guillaume Bal and Xavier Warin. Discrete Ordinates Methods in xy Geometry with Spatially Varying Angular Discretization. *Nuclear Science and Engineering*, 127:169–181, 1997.

[32] Anthony P. Barbu and Marvin L. Adams. Semi-Consistent Diffusion Synthetic Acceleration for Discontinuous Discretizations of Transport Problems. *International Conference on Mathematics, Computational Methods and Reactor Physics, Saratoga Springs, New York*, May 3-7 2009.

[33] L. B. Barichello and C. E. Siewen. On the Equivalence Between the Discrete Ordinates and the Spherical Harmonics Methods in Radiative Transfer. *Nuclear Science and Engineering*, 130:79–84, 1998.

[34] Nathan Bell, Steven Dalton, and Luke N. Olson. Exposing Fine-Grained Parallelism in Algebraic Multigrid Methods. Technical Report NVR-2011-002, NVIDIA, 2011.

[35] W. N. Bell, L. N. Olson, and J. B. Schroder. PyAMG : Algebraic Multigrid Solvers in Python v2.0, 2011. Release 2.0.

[36] R. Blaheta. GPCG-Generalized Preconditioned CG Method and Its Used with Non-Linear and Non-Symmetric Displacement Decomposition Preconditioners. *Numerical Linear Algebra with Applications*, 9:527–550, 2002.

[37] C. Borgers and E. W. Larsen. Fokker-Planck approximation of monoenergetic transport processes. *Trans. Am. Nucl. Soc.*, 71:235–236, 1994.

[38] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial*. Second edition, 2000.

[39] S.L. Campbell, I.C.F. Ipsen, C.T. Kelley, and C.D. Meyer. GMRES and the Minimal Polynomial. *BIT*, 36:1996, 1996.

[40] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960.

[41] E Cuthill and J McKee. *Reducing the bandwidth of sparse symmetric matrices*, pages 157–172. ACM Press, 1969.

[42] G.G. Davidson and T.S. Palmer. Finite Element Diffusion on Arbitrary Quadrilaterals using Rational Basis Functions. *Transactions of the American Nuclear Society*, 87, 2002.

[43] J. O. Deasy. Multiple local minima in radiotherapy optimization problems with dose-volume constraints. *American Association of Physicists in Medicine*, 24:1157–1161, July 1997.

[44] Thomas M. Evans, Kevin T. Clarno, and Jim E. Morel. A Transport Acceleration Scheme for Multigroup Discrete Ordinates with Upscattering. *Nuclear Science and Engineering*, 165:292–304, July 2010.

[45] V. Faber and T.A. Manteuffel. A Look at Transport Theory from the Point of View of Linear Algebra. *Lectures Notes in Pure and Applied Mathematics*, 115:31–61, 1989.

[46] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[47] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[48] G.H. Golub and Q. Ye. Inexact Preconditioned Conjugate Gradient Method with Inner-Outer Iterations. 21:1305–1320, 1999.

[49] L.G. Henyey and J.L. Greenstein. Diffuse Radiation in the Galaxy. *Astrophysical Journal*, 93:70–83, 1981.

[50] Qing Hou, Jun Wang, Yan Chen, and James M. Gavin. An optimization algorithm for intensity modulated radiotherapy - The simulated dynamics with dose-volume constraints. *American Association of Physicists in Medicine*, 30:61–68, December 2002.

[51] J. Patrick Jessee, Woodrow A. Fiveland, Louis H. Howell, Phillips Colella, Richard, and B. Pember. An Adaptive Mesh Refinement Algorithm for the Radiative Transport Equation. *Journal of Computational Physics*, 139(2):380–398, January 1998.

[52] Mark T. Jones and Paul E. Plassmann. A Parallel Graph Coloring Heuristic. *SIAM Journal on Scientific Computing*, 14, 1992.

[53] Guido Kanschat. *Discontinuous Galerkin Methods for Viscous Incompressible Flow*. Springer, first edition, 2007.

[54] Hussein Khalil. A Nodal Diffusion Technique for Synthetic Acceleration of Nodal Sn Calculations. *Nuclear Science and Engineering*, 90:263–280, 1985.

[55] Tommy Knööst, Andres Ahnesjö, Per Nilsson, and Lars Webert. Limitations of a pencil beam approach to photon dose calculations in lung tissue. *Physics in Medicine and Biology*, 40(9):1411–1420, 1995.

[56] Johannes Kraus and Svetozat Margenov. *Robust Algebraic Multilevel Methods and Algorithms*. De Gruyter, 2009.

[57] Thomas Krieger and Otto A Sauer. Monte-Carlo- versus pencil-beam-/collapsed-cone-dose calculation in a heterogeneous multi-layer phantom. *Physics in Medicine and Biology*, 50, 2005.

[58] Yu. Kuznetsov, K. Lipnikov, and M. Shashkov. Mimetic Finite Difference method on Polygonal Meshes. Technical Report LA-UR-03-7608, Los Alamos National Laboratory Report, 2003.

[59] Mark Landesman and J.E. Morel. Angular Fokker-Planck Decomposition and Representation Techniques. *Nuclear Science and Engineering*, 103:1–11, 1989.

[60] E. W. Larsen. Unconditionally Stable Diffusion-Synthetic Acceleration Methods for Slab Geometry Discrete Ordinates Equations. Part I: Theory. *Nuclear Science and Engineering*, 82, 1982.

[61] E. W. Larsen. Diffusion-Synthetic Acceleration Methods for Discrete-Ordinates Problems. *Transport Theory and Statistical Physics*, 13:107–126, 1984.

[62] Edward W. Larsen. The Linear Boltzmann Equation in Optically Thick Systems with Forward-Peaked Scattering. *Progress in Nuclear Energy*, 34, 1999.

[63] K. D. Lathrop. Anisotropic Scattering Approximations in the Monoenergetic Botzmann Equation. *Nuclear Science and Engineering*, 21:498–508, 1965.

[64] W.E Nelson Leonard J. Lorence, r. and J.E. Morel. Coupled Electron-Photon Transport Calculations by the Method of Discrete Ordinates. *IEEE Transactions on Nuclear Science*, NS-32:4416–4420, december 1985.

[65] J. M. Lydon. Photon dose calculations in homogeneous media for a treatment planning system using a collapsed cone superposition convolution algorithm. *Physics in Medicine and Biology*, 43(6):1813–1822, 1998.

[66] Simona Marzi, Maurizio Mattia, Paolo Del Giudice, Barbara Caccia, and Marcello Benassi. Optimization of intensity modulated radiation therapy : assessing the complexity of the problem. *Annali dell'Istituto Superiori di Sanita*, 37:225–230, 2001.

[67] J. E. Morel. On the Validity of the Extended Transport Cross-Section Correction for Low-Energy Electron Transport. *Nuclear Science and Engineering*, 71:64–71, 1979.

[68] J. E. Morel. A Hybrid Collocation-Galerkin-Sn Method for Solving the Boltzmann Transport Equation. *Nuclear Science and Engineering*, 101, 1989.

[69] J.E. Morel. Fokker-Planck Calculations Using Standard Discrete Ordinates Transport Codes. *Nuclear Science and Engineering*, 79, 1981.

[70] J.E. Morel. A 3-D Cell-Centered Diffusion Discretization for Arbitrary Polyhedral Meshes. Technical Report CCS-4:02-40(U), Los Alamos National Laboratory Research Note, 2002.

[71] J.E. Morel and T.A. Manteuffel. An Angular Multigrid Acceleration Technique for S_n Equations with Highly Forward-Peaked Scattering. *Nuclear Science and Engineering*, 107:330–342, 1991.

[72] J.E. Morel and D.P. Sloan. A Hybrid Multigroup/Continuous-Energy Monte-Carlo Method for Solving the Boltzmann-Fokker-Planck Equation. *Nuclear Science and Engineering*, 124, 1996.

[73] Jim E. Morel and James S. Warsa. Spatial Finite-Element Lumping Techniques for the Quadrilateral Mesh S_n Equations in X-Y Geometry. *Nuclear Science and Engineering*, 156:325–342, 2007.

[74] Artem Napov and Yvan Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM Journal of Scientific Computing*, 34(2):1079–1109, 2012.

[75] Y. Notay. Algebraic multigrid and algebraic multilevel methods: a theoretical comparison. *Numerical Linear Algebra with Applications*, 12:419–451, 2005.

[76] Yvan Notay. Flexible Conjugate Gradients. *SIAM Journal on Scientific Computing*, 22:1444–1460, 2000.

[77] Yvan Notay. User's Guide to AGMG. Technical report, Universite Libre de Bruxelles, 2011.

[78] Yvan Notay and Panayot S. Vassilevski. Recursive Krylov-based multigrid cycles. *Numerical Linear Algebra with Applications*, 15:473–487, 2008.

[79] Suely Oliveira and Yuanhua Deng. Preconditioned Krylov Subspace Methods for Transport Equations. *Progress in Nuclear Energy*, 33(1-2):155–174, June 1998.

[80] T.S. Palmer. A Point-Centered Diffusion Differencing for Unstructured Meshes in 3-D. *Proceedings of International Conference in Mathematics and Computations, Reactor Physics and Environmental Analyses, Portland, OR*, pages 897–905, April 30 - May 4 1995.

[81] T.S. Palmer. Discretizing the Diffusion Equation on Unstructured Polygonal Meshes in Two Dimensions. *Annals of Nuclear Energy*, 28:1851–1880, 2000.

[82] HyeongKae Park and Cassiano R.E. de Oliveira. Algebraic Multigrid Methods for the Solution of the Finite Element-Spherical Harmonics Equations. *Transactions of the American Nuclear Society*, 91, 2004.

[83] B. W. Patton and J.P. Holloway. Application of Preconditioned GMRES to the Numerical Solution of the Neutron Transport Equation. *Annals of Nuclear Energy*, 29(2):109–136, January 2002.

[84] S.D. Pautz, J.E. Morel, and M.L. Adams. An Angular Multigrid Acceleration Method for S_n Equations with Highly Forward-Peaked Scattering. *Proceedings of the International Conference on Mathematics and Computation, Reactor Physics and Environmental Analyses in Nuclear Applications, Madrid, Spain*, 1:104–113, September 1999.

[85] Shawn D. Pautz and Marvin L. Adams. An Asymptotic Study of Discretized Transport Equations in the Fokker-Planck Limit. *Nuclear Science and Engineering*, 140:51–69, 2002.

[86] Shawn Daniel Pautz. *Discrete Ordinates Transport Methods for Problems with Highly Forward-Peaked Scattering.* PhD thesis, Texas A&M University, 1998.

[87] G.C. Pomraning. The Fokker-Planck Operator as an Asymptotic Limit. *Mathematical models and Methods in Applied Sciences*, 2(1):21–36, 1992.

[88] G.C. Pomraning and Anil K. Prinja. The Pencil Beam Problem for Screened Rutherford Scattering. *Nuclear Science and Engineering*, 130:1–17, 1998.

[89] K. Przybylski and J. Ligou. Numerical Analysis of the Boltzmann Equation Including Fokker-Planck Terms. *Nuclear Science and Engineering*, 81:92–109, 1982.

[90] Wm. H. Reed. Spherical Harmonics Solutions of the Neutron Transport Equation from Discrete Ordinates codes. *Nuclear Science and Engineering*, 49:10–19, 1972.

[91] Richard Sanchez and Jean Ragusa. On the construction of Galerkin Angular Quadratures. *Nuclear Science and Engineering*, 169(2):133–154, October 2011.

[92] J. Seco, E. Adams, M. Bidmead, M. Partridge, and F. Verhaegen. Head-and-neck IMRT treatments assessed with a Monte-Carlo dose calculation engine. *Physics in Medicine and Biology*, 50, 2005.

[93] David M. Shepard, Michael C. Ferris, Gustavo H. Olvera, and T. Rockwell Mackie. Optimizing the Delivery of Radiation Therapy to Cancer Patients. *Society for Industrial and Applied Mathematics*, 41:721–744, 1999.

[94] Pavel Solin, Jakub Cerveny, and Ivo Dolezel. Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. *Mathematics and Computers in Simulation*, 77(1):117–132, 2008.

[95] Hiromi G. Stone and Marvin L. Adams. A piecewise linear finite element basis with application to particle transport. *Nuclear Mathematical and Computational Sciences : A Century in Review, A Century Anew, Gatlinburg, Tennessee*, April 2003.

[96] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1-2):281–309, 2001.

[97] James L. Thomas, Boris Diskin, and Achi Brandt. Textbook Multigrid Efficiency for Fluid Simulations. *Annual Review of Fluid Mechanics*, 35:317–340, January 2003.

[98] Ray S. Tuminaro and Charles Tong. Parallel Smoothed Aggregation Multigrid: Aggregation Strategies on Massively Parallel Machines. *SC Conference*, 0:5, 2000.

[99] Petr Vanek, Jan Mandel, and Marian Brezina. Algebraic Multigrid on Unstructured Meshes. Technical report, 1994.

[100] Oleg N. Vassiliev, Todd A. Wareing, John McGhee, Gregory Failla, Mohammad R. Salehpour, and Firas Mourtada. Validation of a new grid-based Boltzmann equation solver for dose calculation in radiotherapy with photo beams. *Physics in Medicine and Biology*, 55, 2010.

[101] Andrei M. Voloschenko. Experience in the Use of the Consistent P1 Synthetic Acceleration Scheme for Sn Transport Equation in 2D Geometry. *Proceedings of International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analysis in Nuclear Applications, Madrid,Spain*, 1:104–113, September 1999.

[102] E.L. Wachspress. *A Rational Finite Element Basis.* Academic Press, New York, 1975.

[103] Christian Wagner. *Introduction to Algebraic Multigrid.* 1998. Course Notes of an Algebraic Multigrid Course at the University of Heidelberg in the Winter Semester 1998/1999.

[104] Y. Wang and J. C. Ragusa. A P1 Conforming DSA Scheme for SN transport with DGFEM. *International Conference on Advances in Mathematics,Computational Methods, and Reactor Physics, an international conference sponsored by the ANS Saratoga, NY, May 3-7,2009.*

[105] Y. Wang and J.C. Ragusa. Diffusion Synthetic Acceleration for High-Order Discontinuous Finite Element S_n Transport Schemes and Application to Locally Refined Unstructured Meshes. *Nuclear Science and Engineering*, 166:145–166, 2010.

[106] Yaqi Wang. *Adaptive Mesh Refinement Solution Technique for the Multigroup SN Transport Equation Using a Higher-Order Discontinuous Finite Element Method*. PhD thesis, Texas A&M University, 2009.

[107] Yaqi Wang and Jean C. Ragusa. On the Convergence of DGFEM Applied to the Discrete Ordinates Transport Equation for Structured and Unstructured Triangular Meshes. *Nuclear Science and Engineering*, 163(1):56–72, September 2009.

[108] Yaqi Wang and Jean C. Ragusa. Standard and Goal-Oriented Adaptive Mesh Refinement applied to Radiation Transport on 2D Unstructured Triangular Meshes. *Journal of Computational Physics*, 230:763–788, October 2010.

[109] T. A. Wareing, E. W. Larsen, and M. L. Adams. Diffusion Accelerated Discontinuous Finite Element Schemes for the Sn Equations in Slab and X-Y Geometries. *Proc. Int. Topl. Mtg. Adavances in Mathematics, Computations, and Reactor Physics, Pittsburgh, Pennsylvania*, April 28 - May 2 1991.

[110] T.A. Wareing. New Diffusion-Synthetic Accelerations Methods for the S_N Equations with Corner Balance Spatial Differencing. *Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Application, Karlsruhe, Germany*, 2:500, 19-23 April 1993.

[111] T.A. Wareing, E.W. Larsen, and M.L. Adams. Diffusion Acceleration Discontinuous Finite Element Schemes for the S_N Equations in Slab and XY Geometries. *Proceeding International Topical Meeting on Advances in Mathematics, Computations, Reactor Physics, Pittsburgh, Pennsylvania*, pages 11.1 2–1, 28 April - 2 May 1991.

[112] Todd A. Wareing, John M. McGhee, Jim E. Morel, and Shawn D. Pautz. Discontinuous Finite Element S_N Methods on Three-Dimensional Unstructured Grids. *Nuclear Science and Engineering*, 138:256–268, January 2011.

[113] Todd A. Wareing, Jim E. Morel, and John M. McGhee. Coupled Electron-Photon Transport Methods on 3-D Unstructured Grids. *Transactions of the American Nuclear Society*, 83:240–242, 2000.

[114] James S. Warsa. A Continuous Finite Element-Based, Discontinuous Finite Element Method for Sn Transport. *Nuclear Science and Engineering*, 160:385–400, 2008.

[115] James S. Warsa, Todd A. Wareing, and Jim E. Morel. Fully Consistent Diffusion Synthetic Acceleration of Linear Discontinuous S_N Transport Discretiza-

tions on Unstructured Tetrahedral Meshes. *Nuclear Science and Engineering*, 141(3):236–251, July 2002.

[116] James S. Warsa, Todd A. Wareing, and Jim E. Morel. Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional S_N Calculations in Problesm with Matrial Discontinuities. *Nuclear Science and Engineering*, 147(3):218–248, July 2004.

[117] W.Bangerth, R. Hartmann, and G. Kanschat. deal.II–A General-Purpose Object-Oriented Finite Element Library. *ACM Transactions on Mathematical Software*, 33(4), 2007.

[118] Dingkang Zhang and Farzad Rahnema. A heterogeneous coarse mesh method for coupled photon electron transport problems. *Transport Theory ans Statistical Physics*, 40(4):127–152, 2011.

APPENDIX A

ML OPTIONS

Some of the available coarsening schemes in ML [46]:

Uncoupled: Attempt to construct aggregates of optimal size (3^d nodes in d dimensions). Each process works independently and aggregates cannot span processes.

MIS: Uses maximal independent set techniques to define aggregates. Aggregates can span processes. May provide better quality aggregates than **Uncoupled**, but computationally more expensive because it requires matrix-matrix product.

Some of the smoothers:

Jacobi

Symmetric Gauss-Seidel

Some of the coarse solvers:

Jacobi

Symmetric Gauss-Seidel

Amesos-KLU: Use **KLU** through **Amesos**. Coarse grid problem is shipped to processor 0, solved, and solution is broadcast.

Amesos-UMFPACK: Use **UMFPACK** through **Amesos**. Coarse grid problem is shipped to processor 0, solved, and solution is broadcast.

Amesos-MUMPS: Use double precision version of **MUMPS** through **Amesos**.

The MultiLevelPreconditioner class provide default values for five different preconditioner types:

- Classical smoothed aggregation for symmetric and positive definite or nearly symmetric and definite systems (used here)
- Classical smoothed aggregation-based two-level domain decomposition.
- Three-level algebraic domain decomposition.
- Eddy current formulation of Maxwell's equation.
- Energy-based minimizing smoothed aggregation suitable for highly convective nonsymmetric fluid flow problems.

The options used in this work are:

option name: SA

max levels: 10

prec type: $V-$cycle

aggregation type: uncoupled-MIS

aggregation damping factor: 4/3

eigen-analysis type: cg

eigen-analysis iterations: 10

smoother sweeps: 2

smoother damping factor: 1.0

smoother pre or post: both

smoother type: symmetric Gauss-Seidel

coarse type: Amesos-KLU

coarse max size: 128