ABSTRACT

SCHUNERT, SEBASTIAN. Development of a Quantitative Decision Metric for Selecting the Most Suitable Discretization Method for S_N Transport Problems. (Under the direction of Yousry Y. Azmy.)

In this work we develop a quantitative decision metric for spatial discretization methods of the S_N equations. The quantitative decision metric utilizes performance data from selected test problems for computing a fitness score that is used for the selection of the most suitable discretization method for a particular S_N transport application. The fitness score is aggregated as a weighted geometric mean of single performance indicators representing various performance aspects relevant to the user. Thus, the fitness function can be adjusted to the particular needs of the code practitioner by adding/removing single performance indicators or changing their importance via the supplied weights.

Within this work a special, broad class of methods is considered, referred to as nodal methods. This class is naturally comprised of the DGFEM methods of all function space families. Within this work it is also shown that the Higher Order Diamond Difference (HODD) method is a nodal method. Building on earlier findings that the Arbitrarily High Order Method of the Nodal type (AHOTN) is also a nodal method, a generalized finite-element framework is created to yield as special cases various methods that were developed independently using profoundly different formalisms.

A selection of test problems related to a certain performance aspect are considered: an Method of Manufactured Solutions (MMS) test suite for assessing accuracy and execution time, Lathrop's test problem for assessing resilience against occurrence of negative fluxes, and a simple, homogeneous cube test problem to verify if a method possesses the thick diffusive limit.

The contending methods are implemented as efficiently as possible under a common S_N transport code framework to level the playing field for a fair comparison of their computational load. Numerical results are presented for all three test problems and a qualitative rating of each method's performance is provided for each aspect: accuracy/efficiency, resilience against negative fluxes, and possession of the thick diffusion limit, separately. The choice of the most efficient method depends on the utilized error norm: in $\mathcal{L}_p$ error norms higher order methods such as the AHOTN method of order three perform best, while for computing integral quantities the linear nodal (LN) method is most efficient. The most resilient method against occurrence of negative fluxes is the simple corner balance (SCB) method.

A validation of the quantitative decision metric is performed based on the NEA box-in-box suite of test problems. The validation exercise comprises two stages: first prediction of the contending methods' performance via the decision metric and second computing the actual

scores based on data obtained from the NEA benchmark problem. The comparison of predicted and actual scores via a penalty function (ratio of predicted best performer's score to actual best score) completes the validation exercise. It is found that the decision metric is capable of very accurate predictions (penalty $< 10\%$) in more than 83% of the considered cases and features penalties up to 20% for the remaining cases. An exception to this rule is the third test case NEA-III intentionally set up to incorporate a poor match of the benchmark with the "data" problems. However, even under these worst case conditions the decision metric's suggestions are never detrimental. Suggestions for improving the decision metric's accuracy are to increase the pool of employed data, to refine the mapping of a given configuration to a case in the database, and to better characterize the desired target quantities.

Development of a Quantitative Decision Metric for Selecting the Most Suitable Discretization Method for S_N Transport Problems

by
Sebastian Schunert

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Nuclear Engineering

Raleigh, North Carolina

2013

APPROVED BY:

Ilse Ipsen	Dmitriy Anistratov
John Mattingly	Rachel N. Slaybaugh Co-chair of Advisory Committee

Yousry Y. Azmy
Co-chair of Advisory Committee

DEDICATION

To my parents.

BIOGRAPHY

Sebastian Schunert was born in Banteln, Germany on February 23, 1983. He attended the Gymnasium Alfeld (High School), graduating 2002. He started studying Mechanical Engineering in Braunschweig, Germany in Fall 2003 receiving his Vordiplom (B.S.) in 2005 and his Diplom (M.S. equivalent) in 2008. From August 2007 until July 2008 he worked with Dr. Yousry Azmy at The Pennsylvania State University. In January 2009 he started his Ph.D. under the supervision of Dr. Yousry Azmy at North Carolina State University. His research interests include radiation transport theory, numerical analysis and finite element methods.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my academic advisors, Dr. Yousry Azmy and Dr. Rachel Slaybaugh, for their guidance during my time as an exchange student at The Pennsylvania State University, and later as a Ph.D student at North Carolina State University.

I want to express gratitude towards my current and former office-mates: Jose Duo, Max Rosa, Kursat Bekar, Mike Ferrer, Andrew Bielen, Dan Gill, Josh Hykes, Sean O'Brien, Sameer Vhora, Brian Powell and Noel Nelson. Typically forgotten in other's acknowledgement lists but kindly remembered in this one: Joe Zerr. Without you I might have finished a lot quicker, but it would not have been half as much fun.

To my parents - you instilled in me all the attributes I needed to succeed. I would like to thank you for your support and faith in me during the time I spent at NC State.

Finally, my thanks goes to the Global Village Coffee House: You are doing more for NCSU's academia than any one person ever could.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Solving particle transport problems is of great interest in many disciplines of science and engineering such as nuclear reactor design, astro-physics and health-physics. Traditionally, two vastly different methods were developed to obtain approximate solutions of transport problems, namely stochastic methods (typically referred to as Monte Carlo methods) and deterministic methods; only the latter shall be of interest in this work.

Common to all deterministic algorithms that approximate solutions to transport problems is that they attempt to solve some form of the linear particle transport equation. The two most prominent forms of the linear transport equation are the integro-differential and the integral forms[2], both of which find wide application as the starting point for the derivation of numerical solution methods, but only the former will be considered in this work.

The primary, dependent variable in a deterministic transport calculation is the angular flux, which depends on the six independent phase space variables space and velocity[1], and, possibly the time variable as well. In order to obtain a system of equations that can be solved on a digital computer, all these variables have to be discretized. Because of the lack of alternatives, the energy variable is almost invariably discretized via the multigroup formalism[3], which integrates the transport equation separately over a finite number of energy bins and defines multigroup constants that conserve relevant quantities, e.g. reaction rates, integrated over each energy bin.

For the discretization of the directional variable, two main flavors have developed over the course of the years, namely the S_N and the P_N methods. The S_N method is a collocation method in angle first introduced in [4], while the P_N[5] method projects the transport equation onto the orthogonal set of spherical harmonics[6] that is appropriately truncated, thus closing the resulting system of equations. Throughout this work the S_N method is adopted. Within the framework of this thesis, we are concerned with the discretization of the S_N equations which

[1] Usually expressed as direction of motion and particle energy.

are a set of linear hyperbolic equations in space. The discretization of the spatial variable of the S_N equations will be referred to as spatial discretization. Therefore, the exact solution of the S_N equations is considered the reference solution for all the following discussion. This means that the discussion is confined to the multigroup-S_N realm.

In contrast to the discretization in angle and energy, numerous schemes have been proposed for the discretization of the spatial variables, particularly for the multigroup S_N equations comprising classical finite difference methods, finite volume methods, short characteristic methods, nodal methods and discontinuous finite element methods. This work will elaborate on the relationships between some of these broad classes of methods. The abundance of spatial discretization methods can be attributed to the plethora of typical requirements for the discretization methods. Often, the prospective user has a list of properties ranked from essential to important to nice-to-have. This "check-list" is highly application specific and will therefore vary significantly by application and purpose of the calculation. In the following, a list of possible properties that users might require from spatial discretization methods is compiled:

A. Essential Properties Required of S_N Discretization:

1. Conservation of neutrons: The discretization method satisfies a discrete version of the balance equation.
2. Algebraic linearity: The discretization method does not introduce non-linearities into the underlying numerical method equation or the iterative solution process.

B. Important Properties Required of S_N Discretization:

1. Accuracy: Given a fixed mesh size, the method is close to the exact solution in a norm relevant to the user.
2. Second order truncation error: Assuming a sufficient number of partial derivatives is bounded, the method features a truncation error larger than unity.
3. Pointwise/cellwise convergence: The numerical solution converges to the true solution everywhere even if the exact solution is non-smooth.
4. Resolution of the diffusion limit: In the thick diffusion limit the discretization of the S_N equations satisfies, to leading order, a discretization of the Diffusion equation.
5. Execution time: Given a fixed mesh size and spatial expansion order (collectively described by the number of degrees of freedom), the execution time of the method is small.
6. Computational efficiency: Refers to how much execution time is necessary to achieve a certain level of solution accuracy (assuming iterative error sufficiently reduced).

7. Positivity: Given a positive source and positive inflow into a mesh cell, the discretization method does not produce (or is less susceptible to producing) unphysical negative cell-averaged and cell face angular fluxes.

C. Desirable Properties of S_N Discretization:

1. Robustness: No unphysical oscillations are present even in the presence of strong material heterogeneities, voids, and the like.
2. Minimal spreading of a beam in vacuum (numerical diffusion).

The above list of desired properties is partially adapted from [7] and [8], but it is by no means complete. The properties in A are satisfied by all methods considered in this work (i.e. all methods are conservative and algebraically linear), and are therefore not discussed in any detail. Subset B is within the main focus of this work, while subset C is beyond the scope of this work.

The coexistence of a multitude of spatial discretization schemes is due to the lack of a consistent best performer according to these measures among the set of available schemes. In addition, some of the listed properties are mutually exclusive, e.g. A.2, B.2, and B.7: Algebraically linear, second order methods that preclude negative fluxes cannot exist[9]. The choice of a suitable discretization method is driven by the needs of the user and thus the characteristics of the specific problem to be solved.

This work is concerned with investigating the performance of various spatial discretization schemes of the one-group, multidimensional S_N equations on Cartesian grids with respect to the properties listed above to guide the decision making process in real world applications. To this end three sets of test problems are employed to evaluate the performance of a selection of spatial discretization methods and rank their performance based on suitable performance metrics. A fitness function, adjustable to a given "check-list" of requirements, is designed that aggregates the data obtained from these test problems into a single number indicating how suitable a discretization method is for a specific problem. The fitness function explicitly allows for augmenting the current set with new properties that are not considered within this work, for example robustness in the presence of strong material heterogeneities and/or voids.

The four properties that this work focuses on are execution time, accuracy, positivity, and possession of the discrete diffusion limit; accuracy and execution time, are frequently aggregated into the method's computational efficiency.

Three test problems are used to characterize the performance of the considered discretization methods with respect to the selection of desired properties. This work's results are based on the assumption that the data obtained from the test cases are representative of more complex, real world applications.

A set of pre-existing, promising discretization methods was selected, including discontinuous finite element methods (DGFEM) ([10], [11], [12]), the simple corner balance method (SCB) [7], AHOTN methods[13], linear-nodal(LN), and linear-linear (LL) methods[14] and the arbitrary polynomial order extensions of the Diamond Difference method (HODD) ([15], [16]). All were implemented for three-dimensional Cartesian geometry.

In addition, a novel method that explicitly tracks and eliminates lines and planes of non-smoothness originating from "inconsistent" boundary conditions was developed and implemented. This method uses the *Step approximation* in all cells that are intersected by lines and planes of non-smoothness. It can be considered an extension of Duo's *Singular Characteristic Tracking* algorithm[17] to three spatial coordinates. However, it is important to point out that this extension is highly non-trivial because of the tremendous increase in complexity of the tracking and cell-splitting algorithms involved. The new method is labeled *SCT-Step* method.

All implementations place a high premium on reducing the computational overhead to a minimum in order to level the playing field for a fair comparison with respect to the efficiency aspect discussed before. In addition to implementing these methods, analysis was performed showing that several classes of methods can be recast as discontinuous finite element methods thus unifying the treatment of methods that we previously thought to be unrelated.

The test problems presented within this work are utilized to measure the performance of the implemented spatial discretization methods. The first set of test problems focuses on measuring the discretization method's accuracy and execution time. It is based on an extension of Larsen's two-dimensional homogeneous square test[18] to three spatial dimensions and scattering media. This is achieved by utilizing the *Method of Manufactured Solution (MMS)* [19] approach which allows for securing knowledge of the underlying exact solution without compromising the necessary complexity of the test problem. In contrast to slab geometry, realistic multi-dimensional S_N problems support at most bounded first order partial derivatives of the angular flux[20] thus limiting the attainable convergence rate (with mesh refinement) of any standard spatial discretization method. Therefore, viable test cases need to take the limited exact solution smoothness into consideration because it directly affects the solution accuracy of deployed spatial discretization methods ([21], [22], [17]). The implemented three-dimensional MMS test suite explicitly accommodates an arbitrary degree of smoothness of the exact underlying solution.

The second family of test problems is based in Lathrop's test case [9]. It is designed to be a challenging test for methods' resilience against negative fluxes. It consists of a small source region (region I) enclosed in a large, typically optically thick, source-free region (region II). The solution in region II often suffers from the occurrence of negative fluxes. A family of test problems is created by varying the total cross sections and scattering ratios of the involved materials in regions I and II.

Finally, the third problem tests whether the selected methods possess the thick diffusion limit. Consider a configuration with the property that the optical thickness increases while particle removal, absorption and leakage across the boundaries, vanishes. In this configuration, a typical length scale over which the flux changes significantly is not determined by the particle's mean free path but by the (much larger) diffusion length. Therefore, sufficiently accurate results can be obtained on very coarse meshes (with respect to the particles mean free path) if the method possesses the diffusion limit. A method is said to possess the diffusion limit if the corresponding discretization limits to a discretization of the diffusion equation in configurations as described above. The third set of challenge problems features a homogeneous medium with vacuum boundary conditions on all outside boundaries. The material properties are subjected to scaling using a small parameter such that in the limit of small values, the problem approaches the diffusion limit. The methods' solutions are then compared to the limiting solution of the diffusion problem. For a selection of discretization methods, analysis was performed corroborating the results of the thick-diffusion test problem.

The final goal of this work is the construction of a performance metric aggregating data measuring vastly different properties of the selected discretization methods. The approach taken within this work computes a single fitness value for spatial discretization methods based on user-selected properties that are ranked by importance to the user. The fitness value has the property that it ranges from zero to unity, with zero being the worst and unity being the best score.

This metric, once validated, would be of great utility in production-level S_N codes where the user would set their requirements as input to the code then the decision metric would automatically choose among the various discretization methods implemented in the code that best suits the user's demands.

1.1 The Transport Equation

The linear Boltzmann transport equation describes the evolution of the flux of neutral particles, i.e. neutrons or photons, in a host medium. It can be obtained from the general Boltzmann transport equation by neglecting particle-particle interactions, the dependence of the material properties of the host medium on the particle flux, and assuming that no electric force field is present. Heuristically, it can be derived as a detailed balance of particle production and loss mechanisms in phase space (cf. [23], [3]). The linear transport equation with boundary and

initial conditions in a form that is general enough for our purposes is given by:

$$\begin{aligned}
&\frac{1}{v}\frac{\partial\psi}{\partial t} + \hat{\Omega}\cdot\nabla\psi + \sigma_t(\vec{r},E,t)\psi(\vec{r},\hat{\Omega},E,t) = \\
&\qquad \int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' \sigma_s(\vec{r},\hat{\Omega}\cdot\hat{\Omega}',E'\to E,t)\psi(\vec{r},\hat{\Omega}',E',t) + \\
&\qquad \frac{\kappa(E)}{4\pi}\int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' \nu(\vec{r},E',t)\sigma_f(\vec{r},E',t)\psi(\vec{r},\hat{\Omega}',E',t) + \frac{q(\vec{r},E,t)}{4\pi} \text{ if } \vec{r}\in\mathcal{D} \\
&\qquad \psi(\vec{r},\hat{\Omega},E,0) = \psi_0(\vec{r},\hat{\Omega},E) \\
&\qquad \psi(\vec{r},\hat{\Omega},E,t) = \psi_B(\vec{r},\hat{\Omega},E,t) \text{ if } \vec{r}\in\partial\mathcal{D} \text{ and } \hat{n}\cdot\hat{\Omega}<0,
\end{aligned} \tag{1.1}$$

where

- $\vec{r} = (x,y,z)^T$: Vector of Cartesian spatial coordinates
- $\hat{\Omega} = (\mu,\eta,\xi)^T$: Unit vector of direction cosines with respect to coordinate axes x, y and z.
- E: Energy.
- t: Time.
- $\psi(\vec{r},\hat{\Omega},E,t)$: Angular flux.
- $\sigma_s(\vec{r},\hat{\Omega}\cdot\hat{\Omega}',E'\to E,t)$: Double differential, macroscopic scattering cross section.
- $\sigma_t(\vec{r},E,t)$, $\sigma_f(\vec{r},E,t)$: Macroscopic total collision and fission cross section, respectively.
- $\kappa(E)$: Fission spectrum.
- ν: Fission yield.
- $q(\vec{r},E,t)$: Isotropic external distributed source.
- $\psi_0(\vec{r},\hat{\Omega},E)$: Initial condition: known flux at $t=0$.
- $\psi_B(\vec{r},\hat{\Omega},E,t)$: Explicit boundary conditions: known incoming flux on the boundary.
- $\hat{n}$: Outward normal vector defined on the boundary $\partial\mathcal{D}$.

The independent variables $\vec{r}$, $\hat{\Omega}$, and E constitute the six-dimensional phase space, and the dependent variable, the angular flux ψ, is a distribution over the independent variables.

In this work we are only concerned with steady state solutions of the one-group transport equation in a non-multiplying medium featuring isotropic scattering. To this end, the derivative

of the angular flux with respect to time is set to zero, and the time arguments are dropped in the angular flux and the cross sections. Further, as the medium is non-multiplying and scattering is isotropic: $\sigma_f = 0$ and $\sigma_s(\vec{r}, \hat{\Omega}\cdot\hat{\Omega}', E' \to E) = \sigma_s(\vec{r}, E' \to E)/4\pi$. Thus the transport equation can be written as:

$$\hat{\Omega}\cdot\nabla\psi + \sigma_t(\vec{r}, E)\psi(\vec{r}, \hat{\Omega}, E) = \frac{1}{4\pi}\int_0^\infty dE'\sigma_s(\vec{r}, E' \to E)\phi(\vec{r}, E') + \frac{q(\vec{r}, E)}{4\pi} \text{ for } \vec{r} \in \mathcal{D}, \quad (1.2)$$

where the scalar flux ϕ has been introduced:

$$\phi(\vec{r}, E) = \int_{4\pi} d\hat{\Omega}\psi(\vec{r}, \hat{\Omega}, E). \quad (1.3)$$

For the purpose of discretizing the energy variable, we apply the operator $\int_0^\infty dE\cdot$ to the continuous-energy transport equation, Eq. 1.2, using the following definitions:

$$\begin{aligned}
\hat{\psi}(\vec{r}, \hat{\Omega}) &= \int_0^\infty dE\psi(\vec{r}, \hat{\Omega}, E) \\
\hat{\phi}(\vec{r}) &= \int_0^\infty dE\phi(\vec{r}, E) \\
\sigma_s(\vec{r}, E') &= \int_0^\infty dE\sigma_s(\vec{r}, E' \to E) \\
\hat{\sigma}_k(\vec{r}) &= \frac{\int_0^\infty dE\sigma_k(\vec{r}, E)\psi(\vec{r}, \hat{\Omega}, E)}{\hat{\psi}} \text{ for } k = t, s \\
\hat{q}(\vec{r}) &= \int_0^\infty dEq(\vec{r}, E),
\end{aligned}$$

Using these expressions, Eq. 1.2 can be rewritten as

$$\hat{\Omega}\cdot\nabla\hat{\psi} + \hat{\sigma}_t(\vec{r})\hat{\psi}(\vec{r}, \hat{\Omega}) = \frac{1}{4\pi}\hat{\sigma_s}(\vec{r})\hat{\phi}(\vec{r}) + \frac{\hat{q}(\vec{r})}{4\pi} \text{ for } \vec{r} \in \mathcal{D}. \quad (1.4)$$

For the sake of convenience we omit the hat above all quantities in the remainder of the discussion.

The one-group transport equation, Eq. 1.4, depends continuously on the five remaining phase space variables, namely space $\vec{r}$, and direction of motion of the particles $\hat{\Omega}$. As this work is concerned with the spatial discretization in particular, we discretize the directional variable via the S_N method and use the resulting equations as the starting point of all further discussions. The S_N method proceeds by solving the transport equation only along discrete rays $\hat{\Omega}_n = (\mu_n, \eta_n, \xi_n)^T$, with $n = 1, .., N$, approximating the integration over the angular variables by a quadrature rule $\{\hat{\Omega}_n, w_n\}_{n=1,..,N}$ satisfying $\sum_{n=1}^{N} w_n = 4\pi$. Applying the S_N formalism to

Eq. 1.4 yields:

$$\begin{aligned}
\hat{\Omega}_n \cdot \nabla \psi_n + \sigma_t(\vec{r})\psi_n(\vec{r}) &= \frac{1}{4\pi}\sigma_s(\vec{r})\phi_N(\vec{r}) + \frac{q(\vec{r})}{4\pi} \text{ for } n = 1, .., N \text{ and } \vec{r} \in \mathcal{D} \\
\phi_N(\vec{r}) &= \sum_{n=1}^{N} w_n \psi_n(\vec{r}) \\
\psi_n(\vec{r}) &= \psi_B(\vec{r}, \hat{\Omega}_n) \text{ for } n = 1, .., N \text{ and } \vec{r} \in \partial\mathcal{D} \text{ and } \hat{n} \cdot \hat{\Omega} < 0, \quad (1.5)
\end{aligned}$$

where $\psi_n(\vec{r}) \approx \psi(\vec{r}, \hat{\Omega}_n)$ is an approximation of the true one-group angular flux at $\hat{\Omega}_n$.

The set of S_N equations Eq. 1.5 continuously depends on the spatial variables $\vec{r} = (x, y, z)^T$. All comparisons between reference and numerical solution is made within the S_N framework, i.e. the reference as well as the numerical solution both adopt the S_N approximation for the discretization of the angular variables. Consequently, discretization errors are entirely due to the applied spatial discretization and not due to the finite number of utilized discrete directions N.

1.2 Solution of the One-Group S_N Equations

This section briefly introduces methods to iteratively solve the one-group S_N equations in their first order form. As this work is concerned with spatial discretization methods and not with the iterative solution of the S_N equations, this section shall not aspire for completeness, but rather introduce the concept of the space-angle mesh sweep, the source iteration, and the GMRES solution of the S_N equations to the unfamiliar reader.

1.2.1 Space-Angle Mesh Sweep

Let us first discuss the solution of the S_N equations in the absence of scattering leading to a set of decoupled first order partial differential equations:

$$\begin{aligned}
\hat{\Omega}_n \cdot \nabla \psi_n + \sigma_t(\vec{r})\psi_n(\vec{r}) &= \frac{q(\vec{r})}{4\pi} \text{ for } n = 1, .., N \text{ and } \vec{r} \in \mathcal{D} \\
\phi_N(\vec{r}) &= \sum_{n=1}^{N} w_n \psi_n(\vec{r}). \quad (1.6)
\end{aligned}$$

Anticipating the detailed discussion of spatial discretization schemes, the solution of a single S_N equation along direction n with a given source term can be accomplished by using a mesh sweep[23] if the discretization uses only information from upstream cells. A mesh sweep starts in the corner cell of the domain featuring three boundary faces such that all upstream information is determined by the boundary conditions. After obtaining the solution for the corner cell

including the face fluxes separating this cell and its downstream neighbors, the next downstream cell can be solved by using the already obtained solution for the corner. By repeating this basic step, the spatial mesh can be swept in the downstream direction for each $\hat{\Omega}_n$ until the solution in all cells is obtained.

Mathematically, the mesh sweep recognizes that the global system of discretized equations for each direction $\hat{\Omega}_n$, i.e. the algebraic system comprising the streaming and total interaction operators, is lower or upper triangular (depending of the numbering of the unknowns), and the mesh sweep resembles forward or backward substitution, respectively.

Performing mesh sweeps for all directions in the quadrature set ($n = 1, ..., N$) and applying the quadrature formula completes a single space-angle sweep. If the considered problem is in fact non-scattering, a single space-angle mesh sweep returns the full solution of the S_N equations. In the presence of scattering, an iterative algorithm is necessary for obtaining the solution of the S_N equations.

1.2.2 Source Iterations

The right hand side of the S_N equations, Eq. 1.5, comprises the weighted sum of the angular fluxes along all directions, thus coupling the equations across discrete ordinates. Typically, for the sake of performance, the solution of the S_N equations progresses one discrete ordinate at a time via a space-angle sweep such that a viable iteration scheme must decouple the discrete ordinates within iterations. The idea of the predominantly used *Source Iteration* method is to guess the angular flux, compute the scattering source and right hand side of Eq. 1.5, compute the angular fluxes for all $n = 1, ..., N$ using a space-angle sweep, and then recompute the scattering source. Formally, this can be written as

$$\begin{aligned} \hat{\Omega}_n \cdot \nabla \psi_n^{p+1} + \sigma_t(\vec{r}) \psi_n^{p+1}(\vec{r}) &= \frac{1}{4\pi} \sigma_s(\vec{r}) \phi_N^p(\vec{r}) + \frac{q(\vec{r})}{4\pi} \\ \phi_N^{p+1}(\vec{r}) &= \sum_{n=1}^{N} w_n \psi_n^{p+1}(\vec{r}), \end{aligned} \tag{1.7}$$

where p is the iteration index. Switching to more convenient operator notation borrowed from [24], Eq. 1.7 is recast as:

$$\begin{aligned} \psi_n^{p+1} &= \mathbf{L}^{-1} \left(\mathbf{S} \phi^p + q \right) \\ \phi^{p+1} &= \mathbf{D} \psi^{p+1}. \end{aligned} \tag{1.8}$$

where $\mathbf{L}$, $\mathbf{S}$, and $\mathbf{D}$ are the streaming/collision, scattering, and quadrature operators, respectively. It is understood that the streaming/collision operator is inverted matrix-free within the framework of a space-angle sweep.

1.2.3 GMRES Solution of S_N equations

Recently, attention has arisen to utilize the Generalized Minimal Residual (GMRES)[25] solver for the solution of the one-group S_N equations. The specifics of the GMRES method are detailed in [25]. Here, it shall be sufficient to mention that GMRES only requires matrix-vector multiplications of a matrix A to solve the linear system $Ax = b$. Within the S_N setting, GMRES is utilized as follows[24]. First Eq. 1.8 is manipulated to obtain

$$\mathbf{D}\psi_n = \phi = \mathbf{D}\mathbf{L}^{-1}\left(\mathbf{S}\phi + q\right) \Rightarrow \left(\mathbf{I} - \underbrace{\mathbf{D}\mathbf{L}^{-1}\mathbf{S}}_{\mathbf{A}}\right)\phi = \mathbf{D}\mathbf{L}^{-1}q, \tag{1.9}$$

where iteration indices are dropped. Then, it is recognized that only the matrix-vector product involving $\mathbf{A}$ is required. Therefore, the solution of the S_N equations uses four simple steps:

Before starting GMRES iterations:

0. Perform a single space-angle sweep on the fixed source $b = \mathbf{D}\mathbf{L}^{-1}q$ to obtain the right-hand side for GMRES solution.

Matrix Vector Product: $\left(\mathbf{I} - \mathbf{A}\right)v$

1. Compute the scattering source $s = \mathbf{S}v$.
2. Space-angle sweep on the scattering source $v' = \mathbf{D}\mathbf{L}^{-1}s$.
3. Return $v - v'$.

Given an implementation of the source iteration scheme it is straight forward to implement a GMRES solver subroutine because it relies on the same basic functions that are instrumental to source iterations.

1.3 Thesis Outline

This thesis is organized as follows: in chapter 2 all contending discretization methods are reviewed. It is demonstrated that several of these methods can be recast as discontinuous finite-element methods thus creating a common framework of related methods. Subsequently, the utilized test cases are introduced in chapter 3. The implementation of the contending methods along with two new algorithms regarding spatial discretization methods of the S_N equations are discussed in chapter 4. In chapter 5 numerical results of the contending discretization methods for all three test problems are presented and a qualitative ranking of methods' properties regarding efficiency, positivity and possession of the thick diffusion limit is presented. The data

obtained from the "numerical experiments" in chapter 5 serves as the basis of the quantitative decision metric. Finally, chapter 6 is dedicated to the development and validation of the quantitative decision metric.

Chapter 2

Review of Spatial Discretization Methods

It is the purpose of this work to compare the properties of various (promising) spatial discretization methods of the multi-dimensional S_N equations. This chapter introduces a general classification of spatial discretization methods especially stressing the importance of discontinuous finite element methods (DFEM) before laying out the general framework typically used for the derivation for these methods. Subsequently, methods traditionally used for the discretization of the S_N equations are reviewed along with accounts of their performance whenever available. Finally, some of the reviewed methods are re-derived as DFEM methods, thus reducing the difference between them to differences in the respective test and trial spaces.

2.1 General Classification of Spatial Discretization Methods

The ultimate goal of this work is to construct a decision metric associating features of a given test problem and the computed quantities of interest with the best performing spatial discretization scheme. To this end, it is useful to elaborate on the classification of spatial discretization schemes typically encountered in computational science. Moreover, terminology in most fields like computational fluid dynamics (CFD) closely follows the standard jargon, but computational neutron transport methods evolved without much communication with other fields, and thus utilize a slightly different terminology. This section contrasts three broad classes of spatial discretization methods, namely finite difference methods (FDM)[26], finite volume methods (FVM)[27], and finite element methods (FEM)[28]. It also classifies the type of schemes that are considered in the remainder of this work in the standard jargon as used in [10].

The FDM approximates the solution of the constituting PDEs by grid function values that are only defined at a finite number of points by replacing the partial derivatives present in the

PDE by finite differences. Since finite differences involve the grid function value at neighboring points the equation obtained at each grid point is coupled to the equations at a certain number of neighboring grid points. The approximation order of a finite difference scheme is determined by the employed finite differences: The more neighboring points are involved, the higher in general the order of accuracy[1], but the more coupling between the equations.

In this work, FDMs are not considered for two reasons: First, the S_N equations are an expression of neutron balance, but the FDM is in general not conservative because it approximates the neutron flux at grid points as opposed to over cell volumes. Second, the coupling between neighboring grid points necessitates solving a global matrix equation involving all grid function values if the finite differences involve downstream as well as upstream information. If only upstream values[2] are utilized, sweeping the mesh is still possible. However, depending on the size of the finite difference stencil, a memory overhead compared to more localized methods can be expected because the global system of equations has more non-zero off-diagonal terms. An additional problem when using wide stencils is how to generate grid function values outside of the domain which are necessary when evaluating the FDM equations for points close to the boundary. Further shortcomings of the FDM are that it cannot be extended to unstructured grids (such as tetrahedral grids) and that it may exhibit oscillations near sharp material discontinuities because of the near-discontinuous[29] underlying solution in their vicinity. For the reasons mentioned above, the FDM method is currently not used in neutron transport applications any more, but older attempts can be found in [20].

The FVM most often used in CFD decomposes the domain into homogeneous mesh cells, then integrates the system of conservation equations over the extent of each cell. Subsequently, Gauss' theorem is used and the volume integrals of the derivative terms are recast as integrals over the cell faces. Using the homogeneity of the cell, the volume and face integrals can then be rewritten as averages of the dependent variable over the volume and faces, respectively. In the framework of FVMs, the face-averaged fluxes that originate from applying Gauss' theorem are referred to as numerical fluxes.

The obtained balance-relation between the numerical fluxes and the average is exact, but it comprises more unknowns than equations, and hence requires closure. The closure is usually obtained via a reconstruction approach, which assumes that the true dependent variable has the shape given by some simple function, e.g. a polynomial of some order. Using the averages of the dependent variable in the neighboring cells, an interpolation formula can be devised that allows for the numerical fluxes on the edges to be computed from the cell averages living in the neighboring cells[27].

However, the interpolation formula also globally couples the averages in neighboring cells

[1]Given sufficient smoothness of the underlying exact solution.

[2]For example by using one-sided finite differences

in a manner very similar to FDM (with the exception of the first order step method), imposing the restriction that increasing the accuracy requires enlarging the stencil, thus causing more coupling between cells. Moreover, if the interpolation formula requires downstream values, then the full global system of equations has to be solved simultaneously, increasing the execution time tremendously. Finally, if sharp material discontinuities exist, then the solution might feature oscillations because the reconstruction spanning multiple cells assumes that the underlying solution does not stray too much from the assumed polynomial shape.

Common to all FEM schemes is that the solution of the PDE is approximated by a linear combination of functions belonging to some finite dimensional trial function space. The unknowns of the FEM computation are the coefficients of the linear combination of trial functions, also referred to as expansion coefficients. Several different approaches exist to derive an algebraic system of equations for the unknown expansion coefficients, but common to all of them is that the set of PDEs is replaced by an integral formulation of the problem, i.e. the set of equations of interest is replaced by some integral over the domain of interest: If the solution of the PDE minimizes a particular functional, then the flux expansion via the trial functions can be substituted into the functional, and setting the functional's derivatives with respect to the expansion coefficients to zero provides enough equations to determine all expansion coefficients (Ritz method). If such a functional does not exist, then the residual of the approximate solution can be required to be orthogonal to a set of test functions with respect to some inner product(weighted-residual method). Finally, the least squares FEM (LSFEM) requires the integral of the square of the residuals over the domain to be minimal.

The FEMs can further be divided into continuous (CFEM) and discontinuous methods (DFEM), with the difference between these two classes being whether the global approximate solution is continuous or not. Continuity in an FEM scheme is generally enforced by letting test and trial function spaces be supported on adjacent patches of cells such that flux values at the interfaces of cells are unique, i.e. regardless from which cell the interface point is approached, the same flux value is encountered.

The flux shape on the interfaces can be retrieved by the unique (polynomial) interpolation through the flux values on the interface. As the flux values on the interfaces are unique, the polynomial interpolation is also unique and therefore the flux is continuous pointwise on the interface. On the other hand DFEMs restrict the support for test and trial functions to a single cell such that flux values at the interfaces are local to one cell and therefore not unique[10]. The coupling across cell interfaces in DFEMs is achieved by imposing boundary conditions on the cells faces only in an integral (as opposed to pointwise) sense, which is very similar to the way FVMs impose cell boundary conditions.

From an algorithmic point of view, the major difference between CFEM and DFEM is that the former always features a globally coupled, albeit sparse, matrix, while DFEM's matrix

exhibits a block structure arising from the local character of the test and trial function spaces with very little interdependency between the blocks. As a consequence, CFEM necessitates the simultaneous solution of the global system of equations, but DFEM might allow a mesh sweep if information only propagates downstream. At each step of this mesh sweep, a local system of equations has to be solved whose size depends on the local expansion order but is typically much smaller than the global system of equations. The preferential propagation of information in the S_N equations is accounted for by using the numerical upstream flux which allows application of the mesh sweep, while for CFEMs the preferential direction cannot be accounted for, leading to stability problems.

In summary, the FDM, FVM, and CFEM in their typical form all exhibit undesirable properties that render them unfit for the solution of the S_N equations, while the broad class of DFEM is well suited for this purpose. Consistent with Ref. [10], the general scope of methods considered in this work is referred to as nodal methods, which is used synonymously with DFEM. Nodal methods are a class of methods that share the following properties:

- All function spaces are defined local to a mesh cell.
- Coupling between cells occurs only through their faces.
- Coupling between cells is only imposed in an integral sense.
- Increasing the order of the methods is achieved by increasing the **local** order of expansion.

In neutron transport theory, various spatial discretization methods have been derived using physical arguments, but the final methods still shared all properties of nodal methods. The term nodal method in neutron transport theory strictly applies to methods that use a set of spatial Legendre-Polynomial moments of the S_N transport equations augmented by closure relations obtained from transverse moments of the continuum transport equation as e.g. in [13]. Along the same line of thought are short characteristic schemes[30], which use the same set of moments of the S_N equations, but derive closure relations from approximate solutions obtained from the characteristic form of the transport equation.

Both the nodal and characteristic schemes are shown to resemble discontinuous Petrov-Galerkin FEMs (DPGFEM), and hence fall into the class of nodal methods as defined in this work([1] and [31]). Discontinuous Petrov-Galerkin FEMs are DFEM, but in contrast to discontinuous Galerkin methods, they utilize different test and trial spaces while DGFEM utilize identical test and trial spaces. Many similar schemes exist that utilize the same set of moment equations augmented by some approximating assumption about the flux shape across the cell, and all of these schemes belong to the class of nodal methods considered in this work.

2.2 Discontinuous Finite Element Framework

Throughout the remainder of this work, the discretization of the spatial variables via the discontinuous finite element framework is of special importance. Therefore, this section introduces notation and the weak and strong forms of the S_N transport equation that commonly serve as the starting point for the derivation of discontinuous finite element methods.

Let the domain $\mathcal{D}$ be decomposed into a set of conforming (no "hanging" nodes) cuboidal elements $\mathcal{Q}_{\vec{i}} = [x_{i-1}, x_i] \times [y_{j-1}, y_j] \times [z_{k-1}, z_k]$ with x_i, y_j and z_k indicating mesh cell boundaries such that $\mathcal{D} = \bigcup_{\vec{i}} \mathcal{Q}_{\vec{i}}$ and $\vec{i} = (i, j, k)^T$. Let the set of all faces in $\mathcal{D}$ be given by $\mathcal{E}$, and the set of all faces of $\mathcal{Q}_{\vec{i}}$ be denoted by

$$\mathcal{E}_{\vec{i}} = \left\{ \mathcal{E}^N_{\vec{i}}, \mathcal{E}^S_{\vec{i}}, \mathcal{E}^W_{\vec{i}}, \mathcal{E}^E_{\vec{i}}, \mathcal{E}^T_{\vec{i}}, \mathcal{E}^B_{\vec{i}} \right\},$$

where N, S, W, E, T, and B represent the north, south, west, east, top and bottom faces, respectively. Associated with each face is a unit outward normal vector $\hat{n}_F$ with $F = N, S, W, E, T, B$.

Deviating from standard notation, the E, N, and T faces are always outflow faces, while the W, S, and B faces are always inflow faces. Hence, if components of $\hat{\Omega}$ change sign, then for a given cell the denotation of the cell faces changes. The set of faces is then divided into inflow and outflow faces according to the sign of the inner product: $\hat{n}_F \cdot \hat{\Omega} < 0$ and $\hat{n}_F \cdot \hat{\Omega} > 0$ for inflow and outflow faces, respectively. We refer to the set of all inflow faces as $\mathcal{E}^I$, and to the set of all outflow faces as $\mathcal{E}^O$:

$$\begin{aligned} \mathcal{E}^O &= \{\mathcal{E}^E, \mathcal{E}^N, \mathcal{E}^T\} \\ \mathcal{E}^I &= \{\mathcal{E}^W, \mathcal{E}^S, \mathcal{E}^B\}. \end{aligned} \tag{2.1}$$

Further, we associate with each face an interior and exterior trace denoted by $\mathcal{E}^{F,+}$ and $\mathcal{E}^{F,-}$, respectively. Restriction of information from within the cell to any face is defined on the interior trace, while information from outside the cell is restricted to the exterior trace. Note that discontinuities typical for discontinuous finite element methods[10] originate from the pointwise difference of the exterior and interior flux traces, while for continuous FEM the interior and exterior traces are identical.

The mesh shall always be constructed to approximate the problem configuration such that the material properties, i.e. the total cross section $\sigma_t(\vec{r})$ and the scattering cross section $\sigma_s(\vec{r})$, are constant within the cell $\mathcal{Q}_{\vec{i}}$; then we denote the total and scattering cross sections within that cell by $\sigma_t^{\vec{i}}$ and $\sigma_s^{\vec{i}}$, respectively.

A local approximation of the angular flux $\psi_n^{h,\vec{i}}(\vec{r})$ on the element $\mathcal{Q}_{\vec{i}}$ is given by a linear

combination of trial functions $f_{n,l}^{\vec{i}}(\vec{r})$:

$$\vec{r} \in \mathcal{Q}_{\vec{i}} : \ \psi_n(\vec{r}) \approx \psi_n^{h,\vec{i}}(\vec{r}) = \sum_{l=1}^{L} a_{n,l}^{\vec{i}} f_{n,l}^{\vec{i}}(\vec{r}), \tag{2.2}$$

where the expansion coefficients $a_{n,l}^{\vec{i}}$ as well as the trial functions $f_{n,l}^{\vec{i}}(\vec{r})$ may depend on the Discrete Ordinate index n. The superscript h refers to the utilized mesh spacing $h = \max_{\vec{i}} (x_i - x_{i-1}, y_j - y_{j-1}, z_k - z_{k-1})$ and serves as a reminder that the superscripted quantity is an approximation of the exact solution. The global approximation of the angular flux can be retrieved from the local approximations as their direct sum:

$$\psi_n^h(\vec{r}) = \bigoplus_{\vec{i}} \psi_n^{h,\vec{i}}(\vec{r}). \tag{2.3}$$

Frequently, the restriction of the flux expansion on the cell faces will be needed, so for convenience we denote the restriction onto the interior trace as:

$$\psi_n^h(\vec{r})\Big|_{\mathcal{E}_{\vec{i}}^{F,+}} = \lim_{\epsilon \to 0} \psi_n^h(\vec{r} + s_F|\epsilon|\hat{\Omega}) \text{ if } \vec{r} \in \mathcal{E}_{\vec{i}}^F,$$

and onto the exterior trace as:

$$\psi_n^h(\vec{r})\Big|_{\mathcal{E}_{\vec{i}}^{F,-}} = \lim_{\epsilon \to 0} \psi_n^h(\vec{r} - s_F|\epsilon|\hat{\Omega}) \text{ if } \vec{r} \in \mathcal{E}_{\vec{i}}^F,$$

where

$$s_F = \begin{cases} 1 & \text{if } F \in \mathcal{E}^I \\ -1 & \text{if } F \in \mathcal{E}^O. \end{cases}.$$

Note, that for the definitions of the restriction operators, the global flux solution is used, which means that the restriction onto the exterior trace uses the flux solution in the appropriate adjacent cell.

The local space of test functions $\mathcal{V}^{\vec{i}} = \text{span}\{v_l^{\vec{i}}(\vec{r})\}_{l=1}^{L}$ with $L = \Lambda^3$ is defined such that $v_l^{\vec{i}}(\vec{r}) = 0$ if $\vec{r} \notin \mathcal{Q}_{\vec{i}}$, and as a direct consequence this implies

$$\int_{\mathcal{D}} dV v_l^{\vec{i}}(\vec{r}) G(\vec{r}) = \int_{\mathcal{Q}_{\vec{i}}} dV v_l^{\vec{i}}(\vec{r}) G(\vec{r}) \tag{2.4}$$

for any $G(\vec{r})$. As a short hand notation for the integrals on the left and right hand side we

define

$$\int_{\mathcal{D}} dV v_l^{\vec{i}}(\vec{r}) G(\vec{r}) = \left(v_r^{\vec{i}}(\vec{r}), G(\vec{r}) \right)_{\mathcal{D}} \tag{2.5}$$

$$\int_{\mathcal{Q}_{\vec{i}}} dV v_l^{\vec{i}}(\vec{r}) G(\vec{r}) = \left(v_r^{\vec{i}}(\vec{r}), G(\vec{r}) \right), \tag{2.6}$$

respectively, and similarly for integrals over the cell faces:

$$\int_{\mathcal{E}_{\vec{i}}^F} dS v_l^{\vec{i}}(\vec{r}) G(\vec{r}) = \langle v_r^{\vec{i}}(\vec{r}), G(\vec{r}) \rangle_F.$$

2.2.1 The Weak and Strong Form of the S_N Transport Equation

The derivation of the weak and strong form of the S_N equations largely follows Ref. [10], with the distinct difference that in this work the S_N equations are discussed while [10] illustrates the development of the weak and strong form on the basis of the linear one-dimensional transport equation.

Let the residual of the one-group S_N equations, Eqs. 1.5, be given by:

$$\mathcal{R}_n \left[G_n(\vec{r}) \right] = \hat{\Omega}_n \cdot \nabla G_n(\vec{r}) + \sigma_t(\vec{r}) G_n(\vec{r}) - \frac{\sigma_s(\vec{r})}{4\pi} \sum_{n=1}^{N} w_n G_n(\vec{r}) - \frac{q(\vec{r})}{4\pi}, \tag{2.7}$$

with the obvious property that:

$$\mathcal{R}_n \left[\psi_n(\vec{r}) \right] = 0.$$

In order to derive the weak and subsequently the strong form of the within-group S_N transport equation the local approximation of the angular flux in terms of the trial functions ψ_n^h is substituted into the expression for the residual Eq. 2.7. Note that now:

$$\mathcal{R}_n \left[\psi_n^h(\vec{r}) \right] \neq 0,$$

i.e. the S_N equations are not satisfied pointwise by $\psi_n^h(\vec{r})$.

However, for deriving a discretized system of equations the residual is required to be orthogonal to all members of the test space with respect to the inner product $(\cdot, \cdot)_{\mathcal{D}}$ s.t. by using Eq. 2.4 the following expression can be obtained:

$$\left(v_l^{\vec{i}}(\vec{r}), \hat{\Omega}_n \cdot \nabla \psi_n^{h,\vec{i}} + \sigma_t^{\vec{i}} \psi_n^{h,\vec{i}}(\vec{r}) - \frac{\sigma_s^{\vec{i}}}{4\pi} \phi_N^{h,\vec{i}}(\vec{r}) - \frac{q(\vec{r})}{4\pi} \right)_{\mathcal{D}} = 0 \text{ for all } l = 1, .., L. \tag{2.8}$$

Applying integration by parts to the gradient term results in

$$-\left(\psi_n^{h,\vec{i}},\hat{\Omega}_n\cdot\nabla v_l^{\vec{i}}\right)_{\mathcal{D}}+\left(v_l^{\vec{i}}(\vec{r}),\sigma_t^{\vec{i}}\psi_n^{h,\vec{i}}(\vec{r})-\frac{\sigma_s^{\vec{i}}}{4\pi}\phi_N^{h,\vec{i}}(\vec{r})-\frac{q(\vec{r})}{4\pi}\right)_{\mathcal{D}}=-\sum_F\left\langle v_l^{\vec{i}},\hat{n}_F^T\hat{\Omega}_n\mathcal{F}^*\right\rangle_F, \quad (2.9)$$

where $\mathcal{F}^*$ is the numerical flux on the cell faces. The numerical flux is instrumental in coupling the equations on $\mathcal{Q}_{\vec{i}}$ to the rest of the domain, i.e. it imposes cell boundary conditions and controls the flow of information for the discretization method. Equation 2.9 is referred to as the weak form of the S_N equations because it does not require the trial functions (and hence $\psi_n^{h,\vec{i}}$) to possess integrable first partial derivatives.

The strong form of the S_N equations can be obtained by applying integration by parts again leading to:

$$\left(v_l^{\vec{i}}(\vec{r}),\hat{\Omega}_n\cdot\nabla\psi_n^{h,\vec{i}}+\sigma_t^{\vec{i}}\psi_n^{h,\vec{i}}(\vec{r})-\frac{\sigma_s^{\vec{i}}}{4\pi}\phi_N^{h,\vec{i}}(\vec{r})-\frac{q(\vec{r})}{4\pi}\right)_{\mathcal{D}}=\sum_F\left\langle v_l^{\vec{i}},\hat{n}_F^T\hat{\Omega}_n\left(\psi_n^h\Big|_{\mathcal{E}_{\vec{i}}^{F,+}}-\mathcal{F}_F^*\right)\right\rangle_F, \quad (2.10)$$

which in contrast to the weak form requires the trial functions to possess integrable first partial derivatives. In this work we solely employ the numerical upstream flux given by:

$$\mathcal{F}^*=\begin{cases}\psi_n^h\big|_{\mathcal{E}_{\vec{i}}^{F,-}} & \text{if } \mathcal{E}_F\in\mathcal{E}^I\\ \psi_n^h\big|_{\mathcal{E}_{\vec{i}}^{F,+}} & \text{if } \mathcal{E}_F\in\mathcal{E}^O\end{cases}, \quad (2.11)$$

i.e. the numerical flux is equal to the cell's interior trace on all outflow faces, but equal to the appropriate upstream cell's flux on all inflow edges. Physically, the numerical upstream flux ensures propagation of information only in the direction of neutron travel.

Upon substitution of Eq. 2.11, the weak and strong form Eqs. 2.8 and 2.10 become

$$\begin{aligned}&-\left(\psi_n^{h,\vec{i}},\hat{\Omega}_n\cdot\nabla v_l^{\vec{i}}\right)+\left(v_l^{\vec{i}}(\vec{r}),\sigma_t^{\vec{i}}\psi_n^{h,\vec{i}}(\vec{r})-\frac{\sigma_s^{\vec{i}}}{4\pi}\phi_N^{h,\vec{i}}(\vec{r})-\frac{q(\vec{r})}{4\pi}\right)\\&=-\sum_{\mathcal{E}^O}\left\langle v_l^{\vec{i}},\hat{n}_F^T\hat{\Omega}_n\,\psi_n^h\Big|_{\mathcal{E}_{\vec{i}}^{F,+}}\right\rangle_F-\sum_{\mathcal{E}^I}\left\langle v_l^{\vec{i}},\hat{n}_F^T\hat{\Omega}_n\,\psi_n^h\Big|_{\mathcal{E}_{\vec{i}}^{F,-}}\right\rangle_F,\end{aligned} \quad (2.12)$$

and

$$\begin{aligned}&\left(v_l^{\vec{i}}(\vec{r}),\hat{\Omega}_n\cdot\nabla\psi_n^{h,\vec{i}}+\sigma_t^{\vec{i}}\psi_n^{h,\vec{i}}(\vec{r})-\frac{\sigma_s^{\vec{i}}}{4\pi}\phi_N^{h,\vec{i}}(\vec{r})-\frac{q(\vec{r})}{4\pi}\right)=\\&\qquad\sum_{\mathcal{E}^I}\left\langle v_l,\hat{n}_F^T\hat{\Omega}_n[[\psi_n^h]]_F\right\rangle_F,\end{aligned} \quad (2.13)$$

respectively, where we defined the jump operator $[[]]_F$ as the pointwise difference between the interior and exterior traces:

$$[[G(\vec{r})]]_F = G|_{\mathcal{E}^{F,+}} - G|_{\mathcal{E}^{F,-}} .$$

Most of the spatial discretization methods that are discussed in this work can be obtained by selecting appropriate test and trial function spaces which are substituted into the weak or strong form Eqs. 2.12 and 2.13, respectively, which leads to a local system of algebraic equations.

2.3 Review of Spatial Discretization Methods for the S_N Equations

In this section we review promising classes of spatial discretization schemes including diamond difference type methods, discontinuous Galerkin finite element type methods (DGFEM) and transverse moments based methods (TMB). For convenience let the spatial Legendre moment of the flux denoted by $\psi^{\vec{i}}_{n,\vec{m}}$ be defined as:

$$\begin{aligned} \psi^{\vec{i}}_{n,\vec{m}} &= M^{\vec{i}}_{\vec{m}} \{\psi_n(\vec{r})\} \\ M^{\vec{i}}_{\vec{m}} \{\cdot\} &= \frac{1}{V^{\vec{i}}} \int_{V^{\vec{i}}} dV p^{\vec{i}}_{\vec{m}}(\vec{r}) \cdot \ , \end{aligned} \tag{2.14}$$

where $\vec{m} = (m_x, m_y, m_z)^T$ denotes the order of the moments. Further, triple sums and products of Legendre polynomials are abbreviated by:

$$\begin{aligned} \sum_{\vec{m}=0}^{\Lambda} \cdot &= \sum_{m_x=0}^{\Lambda} \sum_{m_y=0}^{\Lambda} \sum_{m_z=0}^{\Lambda} \cdot \\ p^{\vec{i}}_{\vec{m}}(\vec{r}) &= p^i_{m_x}(x)\, p^j_{m_y}(y)\, p^k_{m_z}(z) , \end{aligned} \tag{2.15}$$

where $p^{i_s}_{l_s}(s)$ is the Legendre polynomial of order l_s, $s = x, y, z$ normalized on the interval $[s_{i_s-1}, s_{i_s}]$. See section A.1 for a precise definition.

2.3.1 Diamond Difference Type Methods

The diamond difference (DD) method is the most commonly known and used spatial discretization method for the S_N equations. There exist an extensive body of literature concerned with the Diamond Difference method in various dimensional Cartesian geometries (among others): [32], [33], and [34] for slab geometry, [9] and [18] for 2D Cartesian geometry, and [23] for 3D Cartesian geometry. In two-dimensional Cartesian geometry, Lathrop [9] derives the DD

method by first integrating the S_N equations over the extent of a single mesh cell, which yields a statement of conservation of particles within the mesh cell. For consistency with the proposition of this work, let us derive the balance equation in 3D Cartesian geometry, i.e. integrating Eq. 1.5 over the extent of cell $\vec{i}$:

$$\frac{\mu_n}{\Delta x_i}\left(\bar{\psi}_{n,E}^{h,\vec{i}} - \bar{\psi}_{n,W}^{h,\vec{i}}\right) + \frac{\eta_n}{\Delta y_j}\left(\bar{\psi}_{n,N}^{h,\vec{i}} - \bar{\psi}_{n,S}^{h,\vec{i}}\right) + \frac{\xi_n}{\Delta z_k}\left(\bar{\psi}_{n,T}^{h,\vec{i}} - \bar{\psi}_{n,B}^{h,\vec{i}}\right) + \sigma_t^{\vec{i}}\bar{\psi}_n^{h,\vec{i}} = \bar{S}^{\vec{i}}, \tag{2.16}$$

where we use the following definitions:

- The cell average angular flux $\bar{\psi}_n^{h,\vec{i}} = \frac{1}{V^{\vec{i}}}\left(1, \psi_n^{h,\vec{i}}\right) = M_0^{\vec{i}}\left\{\psi_n^{h,\vec{i}}\right\}$.
- The total cell average source $\bar{S}_n^{h,\vec{i}} = \frac{1}{V^{\vec{i}}}\left(1, S^{h,\vec{i}}(\vec{r})\right) = M_0^{\vec{i}}\left\{S_n^{h,\vec{i}}\right\}$ known from the previous inner iteration where $S^{\vec{i}}(\vec{r}) = \frac{q(\vec{r})}{4\pi} + \frac{\sigma_s^{\vec{i}}}{4\pi}\phi_N^{h,\vec{i}}$.
- The face average angular fluxes $\bar{\psi}_{n,F}^{h,\vec{i}} = \begin{cases} \frac{1}{A_F^{\vec{i}}}\left\langle 1, \psi_n^h\big|_{\mathcal{E}_{\vec{i}}^{F,-}}\right\rangle & \text{if } \mathcal{E}_F \in \mathcal{E}^I \\ \frac{1}{A_F^{\vec{i}}}\left\langle 1, \psi_n^h\big|_{\mathcal{E}_{\vec{i}}^{F,+}}\right\rangle & \text{if } \mathcal{E}_F \in \mathcal{E}^O \end{cases}$,
- The linear cell dimensions $\Delta x_i = x_i - x_{i-1}$, $\Delta y_j = y_j - y_{j-1}$ and $\Delta z_k = z_k - z_{k-1}$.

The cell balance equation is exact, i.e. it does not encompass any approximation. However, it also comprises four unknowns, namely the face average fluxes on the three outflow faces and the cell average angular flux, while only providing one equation to determine them.

In order to close the system of equations, the angular flux is assumed to be a linear function within the cell $\mathcal{Q}_{\vec{i}}$ [9][3]:

$$\psi_n^{h,\vec{i}}(\vec{r}) = \alpha_n + \beta_n x + \gamma_n y + \delta_n z. \tag{2.17}$$

Following [9], closure relations are determined via the interpolation problem:

$$\begin{aligned} \psi_n^{h,\vec{i}}\left(\frac{x_i + x_{i-1}}{2}, \frac{y_j + y_{j-1}}{2}, \frac{z_k + z_{k-1}}{2}\right) &= \bar{\psi}_n^{h,\vec{i}} \\ \psi_n^{h,\vec{i}}(\vec{r}_F) &= \bar{\psi}_{n,F}^{h,\vec{i}}, \end{aligned} \tag{2.18}$$

where $\vec{r}_F$ is the midpoint of the appropriate face, $\mathcal{E}_F$. Strictly speaking, the cell and face average fluxes are not point values, such that the above equalities hold only up to a second order error. However, given a smooth underlying solution, the DD method is second order accurate such that the error introduced in the interpolation problem is consistent with the overall discretization error. Utilizing Eq. 2.17 in Eqs. 2.18 yields, after some manipulation, the

[3]The reference makes the same assumption in 2D, i.e. the $\delta_n z$ term is not present.

Diamond Difference relations:

$$\bar{\psi}_n^{h,\vec{i}} = \frac{1}{2}\left(\bar{\psi}_{n,E}^{h,\vec{i}} + \bar{\psi}_{n,W}^{h,\vec{i}}\right) = \frac{1}{2}\left(\bar{\psi}_{n,N}^{h,\vec{i}} + \bar{\psi}_{n,S}^{h,\vec{i}}\right) = \frac{1}{2}\left(\bar{\psi}_{n,T}^{h,\vec{i}} + \bar{\psi}_{n,B}^{h,\vec{i}}\right) \tag{2.19}$$

Hebert[15], [16] extends the DD method to the arbitrary expansion order Λ, where Λ is the order up to which spatial Legendre moments of the source are retained:

$$S^{h,\vec{i}}(\vec{r}) = \sum_{\vec{m}=0}^{\Lambda} (2m_x+1)(2m_y+1)(2m_z+1)\, S_{\vec{m}}^{h,\vec{i}} p_{m_x}(x)\, p_{m_y}(y)\, p_{m_z}(z). \tag{2.20}$$

The $S_{\vec{m}}^{h,\vec{i}}$ are referred to as the Legendre source moments and can be obtained by:

$$S_{\vec{m}}^{h,\vec{i}} = M_{\vec{m}}^{\vec{i}}\left\{S^{h,\vec{i}}(\vec{r})\right\}, \tag{2.21}$$

Hebert's higher order Diamond Difference method (HODD) uses cell Legendre moments of the balance relation up to order Λ along with a sufficient number of auxiliary relations to close the system of equations in a manner very similar to the original DD equations. Following [13], the moments of the balance equations can be obtained by applying the operator $M_{\vec{m}}^{\vec{i}}$ to Eq. 1.5:

$$M_{\vec{m}}^{\vec{i}}\left\{\hat{\Omega}_n \cdot \nabla\psi_n + \sigma_t^{\vec{i}}\psi_n(\vec{r})\right\} = S_{\vec{m}}^{h,\vec{i}}. \tag{2.22}$$

After considerable manipulations, the $\vec{m}$-th order Legendre moment of the S_N equation can be obtained:

$$\begin{aligned}
&\frac{\mu_n}{\Delta x_i}\left(\psi_{E,\vec{m}^x}^h - (-1)^{m_x}\psi_{W,\vec{m}^x}^h - 2\sum_{l=0}^{\left[\frac{m_x-1}{2}\right]}(2m_x-4l-1)\,\psi_{\vec{m}-(2l+1)\hat{e}_x}^h\right)\\
+&\frac{\eta_n}{\Delta y_j}\left(\psi_{N,\vec{m}^y}^h - (-1)^{m_y}\psi_{S,\vec{m}^y}^h - 2\sum_{l=0}^{\left[\frac{m_y-1}{2}\right]}(2m_y-4l-1)\,\psi_{\vec{m}-(2l+1)\hat{e}_y}^h\right)\\
+&\frac{\xi_n}{\Delta z_k}\left(\psi_{T,\vec{m}^z}^h - (-1)^{m_z}\psi_{B,\vec{m}^z}^h - 2\sum_{l=0}^{\left[\frac{m_z-1}{2}\right]}(2m_z-4l-1)\,\psi_{\vec{m}-(2l+1)\hat{e}_z}^h\right)\\
&+\sigma_t\psi_{\vec{m}}^h = S_{\vec{m}}^h \text{ for } m_x, m_y, m_z = 0, 1, ..., \Lambda.
\end{aligned} \tag{2.23}$$

In Eq. 2.23 the operator $\left[\frac{m_x-1}{2}\right]$ denotes the truncated integer value of $(m_x-1)/2$. For the sake of lightening the notation, the discrete ordinates index n and the cell index $\vec{i}$ are dropped in Eq. 2.23 and for the remainder of the HODD discussion. Further, Eq. 2.23 utilizes the following definitions:

- The face Legendre order indices $\vec{m}^x = (m_y, m_z)^T$, $\vec{m}^y = (m_z, m_x)^T$, and $\vec{m}^z = (m_x, m_y)^T$.
- The cell face moments (using the east surface as example):

$$\psi^h_{E,\vec{m}^x} = \begin{cases} \frac{1}{A_E} \left\langle p_{m_y}(y) p_{m_z}(z),\, \psi^h_n\big|_{\mathcal{E}^{E,-}} \right\rangle & \text{if } \mathcal{E}_E \in \mathcal{E}^I \\ \frac{1}{A_E} \left\langle p_{m_y}(y) p_{m_z}(z),\, \psi^h_n\big|_{\mathcal{E}^{E,+}} \right\rangle & \text{if } \mathcal{E}_E \in \mathcal{E}^O \end{cases} .$$

Similar to the situation that occurred for the derivation of the DD equations, the set of moments of the balance relation contains more unknowns than equations, specifically $(\Lambda + 1)^3 + 3(\Lambda + 1)^2$ unknowns and only $(\Lambda + 1)^3$ equations. Hebert ([15], [16]) merely states the following auxiliary relations for the HODD method of order $\Lambda \in$ even:

$$\begin{aligned} \frac{1}{2}\left(\psi^h_{E,\vec{m}^x} + \psi^h_{W,\vec{m}^x}\right) &= \sum_{m_x=0,even}^{\Lambda} (2m_x + 1)\psi^h_{\vec{m}},\ \text{for } m_y, m_z = 0, 1, .., \Lambda, \\ \frac{1}{2}\left(\psi^h_{N,\vec{m}^y} + \psi^h_{S,\vec{m}^y}\right) &= \sum_{m_y=0,even}^{\Lambda} (2m_y + 1)\psi^h_{\vec{m}},\ \text{for } m_x, m_z = 0, 1, .., \Lambda, \\ \frac{1}{2}\left(\psi^h_{T,\vec{m}^z} + \psi^h_{B,\vec{m}^z}\right) &= \sum_{m_z=0,even}^{\Lambda} (2m_z + 1)\psi^h_{\vec{m}},\ \text{for } m_x, m_y = 0, 1, .., \Lambda, \end{aligned} \tag{2.24}$$

and $\Lambda \in$ odd:

$$\begin{aligned} \frac{1}{2}\left(\psi^h_{E,\vec{m}^x} - \psi^h_{W,\vec{m}^x}\right) &= \sum_{m_x=1,odd}^{\Lambda} (2m_x + 1)\psi^h_{\vec{m}},\ \text{for } m_y, m_z = 0, 1, .., \Lambda, \\ \frac{1}{2}\left(\psi^h_{N,\vec{m}^y} - \psi^h_{S,\vec{m}^y}\right) &= \sum_{m_y=1,odd}^{\Lambda} (2m_y + 1)\psi^h_{\vec{m}},\ \text{for } m_x, m_z = 0, 1, .., \Lambda, \\ \frac{1}{2}\left(\psi^h_{T,\vec{m}^z} - \psi^h_{B,\vec{m}^z}\right) &= \sum_{m_z=1,odd}^{\Lambda} (2m_z + 1)\psi^h_{\vec{m}},\ \text{for } m_x, m_y = 0, 1, .., \Lambda, \end{aligned} \tag{2.25}$$

but does not provide a rigorous derivation for their particular form. However, following the example in [9] for DD in 2D Cartesian geometry and [34] for an arbitrary order DD scheme in slab geometry, Hebert's extension can be derived by assuming the flux to have a polynomial

flux shape of order $\Lambda + 1$:

$$\psi^h(\vec{r}) = \sum_{\vec{m}=0}^{\Lambda} \alpha_{\vec{m}} p_{m_x}(x) p_{m_y}(y) p_{m_z}(z) + \sum_{\vec{m}^x=0}^{\Lambda} \alpha_{\Lambda+1,m_y,m_z} p_{\Lambda+1}(x) p_{m_y}(y) p_{m_z}(z) + \sum_{\vec{m}^y=0}^{\Lambda} \alpha_{m_x,\Lambda+1,m_z} p_{m_x}(x) p_{\Lambda+1}(y) p_{m_z}(z) + \sum_{\vec{m}^z=0}^{\Lambda} \alpha_{m_x,m_y,\Lambda+1} p_{m_x}(x) p_{m_y}(y) p_{\Lambda+1}(z). \quad (2.26)$$

For the purpose of deriving the auxiliary relations Eqs. 2.24 and 2.25, the assumed shape Eq. 2.26 is required to match the cell Legendre moments and the face Legendre moments on all faces. The complete proof is conducted in Sec. B.1. It is important to point out that the interpolation does not enforce pointwise flux continuity across the cells' interfaces. This is illustrated for the HODD-0, i.e. the DD, method: The flux on the faces is a linear function, but only the averages and not the slopes of the fluxes on the exterior and interior trace are forced to match during the interpolation procedure and thus pointwise continuity is not enforced. Even though the interpolation procedure does not enforce continuity between cells, it imposes a significant coupling of the inflow, outflow and nodal flux moments within each cell.

The DD method and its generalizations were subject to both theoretical and numerical investigations in slab and multi-dimensional Cartesian geometries. However, the results obtained for slab geometries do not carry over to multi-dimensional geometries because of the lack of smoothness of the exact solution in multi-dimensional Cartesian geometries.

The DD method in multi-dimensional Cartesian geometries is second order accurate if the underlying exact solution features bounded third partial derivatives[35]. However, as demonstrated in Ref. [20], realistic configurations provide, at most, bounded first partial derivatives such that the theoretical a-priori error estimate in [35] does not hold in practice. Later, Larsen[18] showed numerically for a simple test problem that the DD method in fact does not exhibit second order accuracy for a problem that features bounded first partial derivatives when measuring the error in a discrete 2-norm. However, integral quantities such as region-integrated fluxes/reaction rates or the multiplication factor converge with the theoretically predicted second order.

Building on Larsen's work, Azmy[21], and Duo, and Azmy([1], [22]) demonstrated on variations of Larsen's benchmark that DD, along with two other constant spatial approximations of the S_N equations (AHOTN-0 and AHOTC-0), exhibits observed orders of accuracy that (1) depend on the applied error norm and (2) on the smoothness of the underlying exact solution of the S_N problem. Moreover, in the case of a discontinuous angular flux, all three methods fail to converge cell-wise to the exact solution, i.e. some cells do not converge to the exact solution.

The early popularity of the DD method can be greatly attributed to the prospect of its second-order accuracy, even though only the average source (i.e. no higher order moments) is retained. While other schemes feature that same property (AHOTN-0 in multi-dimensional Cartesian geometries and both AHOTN-0 and the step characteristic method in slab geometry), it can be shown that all of them are asymptotically, i.e. in the limit of optically small mesh cells, equivalent to the DD method. In multi-dimensional geometries, the DD method is also inexpensive when compared with other constant approximation methods.

However, the downside of the DD method is that it can produce negative cell average angular fluxes (and potentially scalar fluxes) if the cell optical thickness $t_{n,x}^{\vec{i}} = \frac{\sigma_t^{\vec{i}} \Delta x_i}{2\mu_n}$ (in the x-dimension, analogous definitions apply for the y and z-dimension) exceeds a value of two[23]. While sometimes tolerable, negative fluxes can lead to instability/failure of acceleration procedures[9], or failure of subsequent models involving different physics that are driven by the neutron transport solution.

Reed[36] associates discretization methods that enforce flux continuity with poor behavior in optically thick regions, namely oscillations and occurrence of negative cell average fluxes. Even though he wrongly classifies the DD method as continuous by extending its properties from slab geometry to multidimensional geometries, it holds true for DD (and also for HODD) that more coupling in between cells is enforced (see subsection 2.4.1) than e.g. for discontinuous FEM methods, which leads to the general lack of positivity and robustness. While methods that enforce less rigidity in the cell-to-cell solution are less prone to negative solutions, they do not necessarily guarantee positive solutions: In fact, strictly positive schemes all feature an accuracy less than second order. In conclusion, the HODD method is expected to be more accurate than e.g. the DGFEM method given the same source expansion in smooth, well resolved regions, but to fail in optically unresolved, non-smooth, or strongly heterogeneous regions.

The relationship of the HODD method with the general class of Discontinuous FEM methods, i.e. nodal methods, is further discussed in section 2.4.

2.3.2 The Discontinuous Galerkin Finite Element Method (DGFEM)

The discontinuous Galerkin finite element method (DGFEM) uses identical polynomial test and trial function spaces that are typically substituted into the weak form and tested against all members of the test space to obtain a per-cell system of equations. Following [12] to obtain a compact expression of the DGFEM equations let the members of the test/trial space in cell $\vec{i}$ be collected in the vector $\vec{f}^{\,\vec{i}}$ and the set of unknown expansion coefficients be collected in $\vec{\psi}^{\vec{i}}$. In addition let the source be expanded in the set of trial functions such that the flux and source expansions within cell $\vec{i}$ are given by $\psi_n^{h,\vec{i}}(\vec{r}) = \left(\vec{f}^{\,\vec{i}}\right)^T \vec{\psi}_n^{h,\vec{i}}$ and $S^{h,\vec{i}}(\vec{r}) = \left(\vec{f}^{\,\vec{i}}\right)^T \vec{S}^{h,\vec{i}}$.

Subsequently the flux and source expansions are substituted into Eq. 2.12:

$$-\hat{\Omega}_n \underbrace{\left(\nabla\vec{f},\vec{f}^{\,T}\right)}_{\mathbf{D}}\vec{\psi}_n^{\,h,\vec{i}} + \underbrace{\left(\vec{f},\vec{f}^{\,T}\right)}_{\mathbf{M}}\left[\sigma_t\vec{\psi}_n^{\,h,\vec{i}} - \vec{S}^{h,\vec{i}}\right] + \sum_{\mathcal{E}^O}\hat{n}_F^T\hat{\Omega}_n \underbrace{\left\langle \vec{f}\left(\vec{r}_F\right), \vec{f}^{\,T}\left(\vec{r}_F\right)\right\rangle_F}_{\mathbf{E}_F}\vec{\psi}_n^{\,h,\vec{i}} =$$
$$-\sum_{\mathcal{E}^I}\hat{n}_F^T\hat{\Omega}_n \underbrace{\left\langle \vec{f}\left(\vec{r}_F\right), \left(\vec{f}^{\,\vec{i}'}\right)^T\left(\vec{r}_F\right)\right\rangle_F}_{\mathbf{E}_F}\vec{\psi}_n^{\,h,\vec{i}'}, \tag{2.27}$$

where the superscript $\vec{i}'$ denotes the appropriate upstream cell and $\mathbf{D}$, $\mathbf{M}$ and $\mathbf{E}_F$ are the stiffness, mass and edge matrices, respectively. The question at hand for deriving a DGFEM method is which polynomial function space to use. From a theoretical point of view, i.e. disregarding its numerical implementation, it is irrelevant which explicit basis functions are used to describe a particular function space: As long as two function spaces have an identical span, the results of the respective DGFEM computations are equivalent[10]. However, the user still has to select the order and the family of the function space: We adopt the notation where the order indicates the highest occurring power of the spatial variables x, y and z, while the family is classified by which polynomial cross-terms are retained for a given order[28]. The following discussion introduces two families of function spaces: the *complete* and the *Lagrange*[4] families.

The DGFEM method for discretizing the S_N equations was first suggested by Reed and Hill [36] for two-dimensional triangular cells using a basis of *Lagrange* polynomials: Each *Lagrange* basis function is associated with a support point at which its value is unity while it assumes a zero value at all other support points. The unknowns in Reed's methods are then the flux values at the support points and the method's order is related to the number of support points within a single cell. For the DGFEM scheme of order Λ Reed's scheme features $L = \frac{(\Lambda+1)(\Lambda+2)}{2}$ distinct support points:

$$\psi_n^{h,\vec{i}}(x,y) = \sum_{l=1}^{L}\psi_n^h\left(\vec{r}_l\right)d_l\left(\vec{r}\right) \tag{2.28}$$

where d_l is the *Lagrange* polynomial at support point $\vec{r}_l$ (see section A.2). As this work is concerned with Cartesian geometries we refer to [36] for more information on how to place support points within the cell. Reed[36] also gives an alternative monomial form of his Lagrange basis which shows that the highest mixed moment term (e.g. the $x{\cdot}y$ for $\Lambda = 1$) is not retained on triangular geometries. Later, Wang[12] derived a DGFEM method also for triangular geometry suitable for hp-refinement by using a hierarchical basis (i.e. his basis functions are neither monomials nor *Lagrange* polynomials). While the specifics of the derivation are not important for this work it is important that again the number of independent basis functions is $L =$

[4]Lagrange type does not imply that Lagrange polynomials are used.

$\frac{(\Lambda+1)(\Lambda+2)}{2}$. On three-dimensional Cartesian meshes Evans implements two families of DGFEM methods into the transport code DENOVO[24]. The first DGFEM methods which he refers to as linear discontinuous (LD) method uses the following approximation for the angular flux:

$$\psi_n^{\vec{i},h}(\vec{r}) = \sum_{m\leq 1} \psi_{\vec{m}}^{\vec{i},h} p_{\vec{m}}(\vec{r}), \tag{2.29}$$

where $m = m_x + m_y + m_z$. Reeds', Wang's and Evan's LD method are all examples of the *complete* DGFEM family[28].

In two-dimensional Cartesian geometry Gastaldo[37] derived a DGFEM scheme of arbitrary order Λ which retains all cross moments, i.e. for $\Lambda = 1$ the trial and test spaces would comprise the $x \cdot y$ term. Along the same line, yet limited to $\Lambda = 1$, is Evans' second DGFEM implementation in DENOVO which uses the following function space:

$$\psi_n^{\vec{i},h}(\vec{r}) = \sum_{m_x=0}^{1} \sum_{m_y=0}^{1} \sum_{m_z=0}^{1} \psi_{\vec{m}}^{\vec{i},h} p_{\vec{m}}(\vec{r}). \tag{2.30}$$

Evans refers to this DGFEM method as the tri-linear discontinuous method. Gastaldo's and Evan's TLD methods are examples of the *Lagrange* DGFEM family. Note, that the term *Lagrange* does not imply that *Lagrange* polynomials are employed; the name originates from the original construction of the *Lagrange* set: Use **Lagrange** polynomials to create a function space per dimension, i.e. in x, y and z direction. This is achieved by separately distributing the support points within the one-dimensional x, y and z ranges, then creating the 3D function space as the outer product of the resulting one-dimensional function spaces.

In summary two families of DGFEM function spaces are mostly used in discretizing the spatial variable in the S_N approximation of the transport equation: (1) the *complete* family and (2) the *Lagrange* family. When expressed in Legendre polynomials up to order Λ the approximation of the angular flux within a mesh cell for the *complete* and the *Lagrange* set can be expressed as follows:

$$\psi_n^{\vec{i},h}(\vec{r}) = \sum_{m\leq\Lambda} \psi_{\vec{m}}^{\vec{i},h} p_{\vec{m}}(\vec{r}), \tag{2.31}$$

and

$$\psi_n^{\vec{i},h}(\vec{r}) = \sum_{m_x=0}^{\Lambda} \sum_{m_y=0}^{\Lambda} \sum_{m_z=0}^{\Lambda} \psi_{\vec{m}}^{\vec{i},h} p_{\vec{m}}(\vec{r}), \tag{2.32}$$

respectively.

The following observations are now in order:

- For a given order Λ the *Lagrange* set has more unknowns per mesh cell. Therefore, it is expected to be more accurate but also more expensive. The question at hand is which of the two families features a superior computational efficiency.

- Assume that we formulate our function spaces such that we solve for point values of the flux, i.e. we use *Lagrange* polynomials as basis functions. Then, in two-dimensional triangular geometry and three-dimensional tetrahedral geometry the *complete* basis would require one flux value per corner point. The *Lagrange* basis would introduce more degrees of freedom that are not associated with the flux values in the corner points. In two-dimensional and three-dimensional Cartesian geometry the *Lagrange* family would result in one flux value per corner point. The *complete* basis would result in less degrees of freedom. For $\Lambda = 1$ for example, the *Lagrange* function space seems for more natural for Cartesian meshes, while the *complete* family appears to be a more natural choice for triangles/tetrahedra.

- Adams[7] finds that the linear order *Lagrange* spaces on Cartesian grids and the linear order *complete* sets on triangular and tetrahedral grids (but not the other way around) feature properties inherent in their test/trial spaces that allow them the resolution of the thick diffusion limit on the respective grids.

The idea of reducing the number of polynomial cross moments is neither new nor restricted to the DGFEM method, the same idea will return for the transverse moment based method. It seems beneficial to us to compare the efficiency of the *complete* and *Lagrange* families since it is not clear which will be more efficient for a fixed order Λ: While the *Lagrange* family is more accurate, the *complete* family is expected to execute faster.

In two-dimensional Cartesian geometry and in the absence of solution discontinuities Lesaint and Raviart[38] demonstrated that the order of accuracy of the *Lagrange* DGFEM set of order Λ is $\mathcal{O}\left(h^{\Lambda+1}\right)$ in a continuous two norm. Less important for this work, but necessary to appreciate the results of the following reference Richter[39] showed that the same order holds true in two-dimensional triangular geometry. Both references note though that the observed accuracy is limited by the regularity of the solution, i.e. the expected accuracy is $\mathcal{O}\left(h^{\min(\Lambda+1,r)}\right)$ with r being the regularity index of the solution. Wang and Ragusa[11] performed an extensive convergence study using *complete* sets of order Λ for $\Lambda = 1, .., 4$ in two-dimensional triangular geometry for test problems designed in the spirit of Duo and Azmy[22] whose exact solution features a limited degree of smoothness. The level of smoothness is referred to as C_0 and C_1 denoting configurations that feature discontinuous angular fluxes and discontinuous first partial derivatives of the angular fluxes, respectively. A more formal definition of the smoothness C_p is given in section 3.2. The purpose of their study is to show by numerical experiment that

the a-priori error estimates in [38] and [39] are observed. Their findings can be summarized as follows:

- If the mesh is aligned with the singular characteristic line (no cell is intersected by SC) and the problem is purely absorbing the theoretical order of accuracy is observed. This could be expected since the exact solution within each cell is smooth.

- Regardless of scattering, if the mesh is not aligned with the SC the observed orders of accuracy are 1/2 and 3/2 for a C_0 and C_1 problem, respectively. This is expected because r is 1/2 and 3/2 for the C_0 and C_1 problem, respectively.

- If the mesh is aligned, the medium scatters and the boundary conditions render the problem C_0 then the observed accuracy is 3/2. Wang did not explain this phenomenon. We propose the following explanation: The uncollided flux does not feature a discontinuity within any of the cells since the mesh is aligned with the SC. However, the first collided source creates a collided flux that features discontinuous partial derivatives in a manor very similar to Duo's second MMS test case in Table 3.2. Therefore, the flux within each cell does not feature continuous first partial derivatives and thus the observed accuracy is limited by order 3/2.

Wang also applied the DGFEM method to more realistic problems (EIR-2 benchmark, Takeda and C5G7) in [12] demonstrating that integral quantities such as the eigenvalue (Takeda) and region integrated flux (EIR-2) enjoy convergence close to (and approaching) their theoretically predicted rates; as a corollary the increased observed accuracy makes high-order methods attractive. In fact, Wang plots the error in the region integrated flux (eigenvalue) vs. the execution time for the EIR-2 (Takeda) problem, respectively, and demonstrates that the most efficient method is the DGFEM-4 method.

The DGFEM-1 method is much less prone to negative fluxes than the DD method, yet negative fluxes can arise[23]. This behavior may be attributed to DGFEM-1 imposing less restrictions on inter-cell continuity than DD. In fact, the whole family of DGFEM methods does not constrain the inter-cell jump at all while the HODD family enforces continuity in an integral sense. This might also lead to an improved behavior at material interfaces featuring vastly different total cross sections where near-discontinuities of the exact angular flux along the characteristic can occur. These near-discontinuities can lead to spurious oscillations in the computed solution.

The TLD (or *Lagrange* type of order 1) method also allows the resolution of the thick diffusion limit on multi-dimensional Cartesian meshes even though the limiting discretization of the Diffusion equation might not be good since it lacks robustness and accuracy (negative solutions, discontinuities, oscillations, poor approximations to boundary conditions) [40],[7].

Within both references Adams suggests lumping the mass, surface and stiffness matrices (fully lumped DGFEM method) which yields a better thick diffusion limit.

In conclusion, the DGFEM method is robust (allowing for large inter-cell jumps), allows the resolution of the diffusion limit in certain cases and is simple since it uses polynomials for the interpolation thus yielding a potential advantage regarding execution time when compared to non-polynomial methods. However, it remains to be determined if it is less accurate than the HODD method of the same order.

2.3.3 Simple Corner Balance Method

The simple-corner balance (SCB) method on hexahedral grids introduced by Adams in his seminal work in [40] and [7] is a relative of the *Lagrange* type DGFEM of order one, i.e. the TLD method. It can be derived from the pertaining DGFEM equations by a process referred to as lumping applied to all mass, stiffness and face matrices. The process of lumping reduces the accuracy of the method but increases its robustness in the diffusion limit, a feature that ranks higher in the radiative transfer community than accuracy[8].

The process of lumping can only be applied to the particular form of the TLD equations that is obtained when a cardinal set of basis functions is utilized, i.e. using the Lagrange interpolatory functions defined in section A.2. The eight corner points of the hexahedral mesh cells are used as the support points for the Lagrange interpolants such that the eight basis functions are given by:

$$d_{i_x,i_y,i_z} = \frac{x - x_{i'_x}}{x_{i'_x} - x_{i_x}} \frac{y - y_{i'_y}}{y_{i'_y} - y_{i_y}} \frac{z - z_{i'_z}}{z_{i'_z} - z_{i_z}}, \tag{2.33}$$

where

$$\begin{aligned} i_x &= i \text{ or } i-1 \\ i_y &= j \text{ or } j-1 \\ i_z &= k \text{ or } k-1 \end{aligned} \tag{2.34}$$

and i'_k is the respective other choice.

In the case the SCB the components of the vector $\vec{\psi}_n^{\,h,\vec{i}}$ are the point flux values at the corners of the hexahedron. We shall order the corners according to the numbering scheme in Fig. 2.1. Substituting d_{i_x,i_y,i_z} into the weak form Eq. 2.9 results in the *Lagrange* type DGFEM

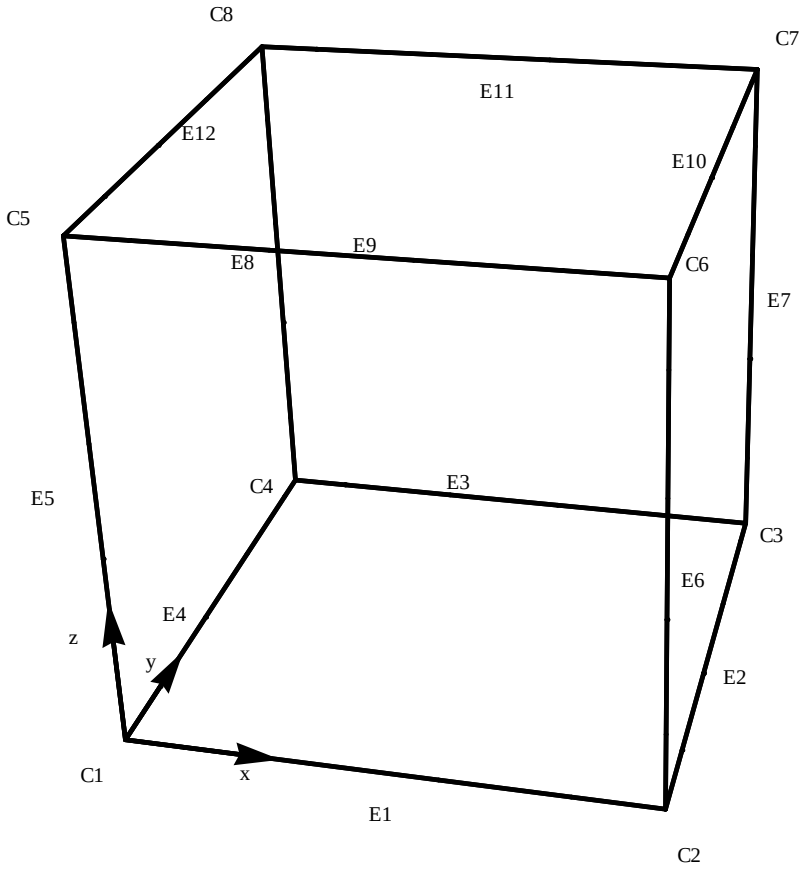


Figure 2.1: Numbering of the corners and edges of a mesh cell.

of order one to obtain a linear system of equations analogous to equation 2.27:

$$\begin{aligned} & - \mu_n \mathbf{D}_x \vec{\psi}_n^{\,h,\vec{i}} - \eta_n \mathbf{D}_y \vec{\psi}_n^{\,h,\vec{i}} - \xi_n \mathbf{D}_z \vec{\psi}_n^{\,h,\vec{i}} + \mu_n \mathbf{E}_E \vec{\psi}_n^{\,h,\vec{i}} + \eta_n \mathbf{E}_N \vec{\psi}_n^{\,h,\vec{i}} + \xi_n \mathbf{E}_T \vec{\psi}_n^{\,h,\vec{i}} + \sigma_t \mathbf{M} \vec{\psi}_n^{\,h,\vec{i}} \\ & = \mathbf{M} \vec{S}_n^{\,h,\vec{i}} + \mu_n \mathbf{E}_W \vec{\psi}_n^{\,h,\vec{i}'} + \eta_n \mathbf{E}_S \vec{\psi}_n^{\,h,\vec{i}''} + \xi_n \mathbf{E}_B \vec{\psi}_n^{\,h,\vec{i}'''}, \end{aligned} \tag{2.35}$$

where $\mathbf{M}$, $\mathbf{D}_x$, and $\mathbf{E}_E$, etc. are given explicitly in the Mathematica notebooks in section B.2 and $\vec{i}' = \vec{i} - \mathrm{sg}(\mu_n)\hat{e}_x$, $\vec{i}'' = \vec{i} - \mathrm{sg}(\eta_n)\hat{e}_y$ and $\vec{i}''' = \vec{i} - \mathrm{sg}(\xi_n)\hat{e}_z$. The SCB equations are obtained from Eq. 2.35 by lumping the mass, stiffness and face matrices.

Mass Lumping

Instead of computing the mass matrix as prescribed in Eq. 2.27 it is approximated by first evaluating the flux expansion at the support point of the r-th weight function:

$$(\mathbf{M})_{r,c} = (f_r, f_c) \rightarrow f_c(\vec{r}_r)(f_r, 1) = \delta_{r,c}(f_r, 1) = (\mathbf{M}_L)_{r,c}, \tag{2.36}$$

where $\delta_{r,c}$ is the Kronecker delta. Lumping the mass matrix amounts to summing all the contributions on the off-diagonal into the diagonal elements. The lumped mass matrix for the discussed TLD scheme is listed in the form of a Mathematica notebook in section B.2.

Lumping of Surface Terms

The lumping of the surface terms proceeds similarly as the mass matrix lumping by evaluating the flux expansion over face F at the support point of the r-th weight function:

$$(\mathbf{E}_F)_{r,c} = \langle f_r(\vec{r}_F), f_c(\vec{r}_F)\rangle_F \rightarrow \delta_{r,c}\langle f_r, 1\rangle_F = (\mathbf{E}_{F,L})_{r,c}. \tag{2.37}$$

The lumped east face matrix for the discussed TLD scheme is listed in the form of a Mathematica notebook in section B.2.

Lumping of Stiffness Terms

The lumping of the stiffness term differs from the straight-forward lumping process applied to face and mass matrices. First, integration by parts is used to derive the following identity:

$$-\mu_{k,n}(\mathbf{D}_k + \mathbf{E}_{k^+} - \mathbf{E}_{k^-}) = \mu_{k,n}\left(\vec{f}, \nabla_k \vec{f}^T\right). \tag{2.38}$$

Then, instead of performing the integration in Eq. 2.38, the lumping procedure evaluates the gradient term at the support points of the r-th basis function $\vec{r}_r$:

$$\left(\vec{f}, \nabla_k \vec{f}^T\right) \rightarrow \nabla_k \vec{f}(\vec{r}_r)\left(\vec{f}, 1\right). \tag{2.39}$$

Finally, Eqs. 2.38 and 2.39 are used to compute the lumped stiffness term:

$$\mathbf{D}_{k,L} = \mathbf{E}_{k^+,L} - \mathbf{E}_{k^-,L} - \left(\vec{f}, \nabla_k \vec{f}^T\right)_L. \tag{2.40}$$

As an example the lumped stiffness matrix for the x-direction is listed in section B.2.

SCB Equations

The SCB equations are obtained as the fully lumped version of the TLD equations. The method's name *Simple Corner Balance* is attributed to the common interpretation of the resulting equations as holding a balance over one of the eight subvolumes associated with the cell's corners. The unknown corner flux values are interpreted as the volume averages within these subvolumes. For the first corner the equations for $\hat{\Omega}_n$ in the first angular quadrant are:

$$\mu_n \frac{\Delta y_j \Delta z_k}{4} \left(\frac{\psi_{n,1}^{h,\vec{i}} + \psi_{n,2}^{h,\vec{i}}}{2} - \psi_{n,2}^{h,\vec{i}-\hat{e}_x} \right) + \eta_n \frac{\Delta x_i \Delta z_k}{4} \left(\frac{\psi_{n,1}^{h,\vec{i}} + \psi_{n,4}^{h,\vec{i}}}{2} - \psi_{n,4}^{h,\vec{i}-\hat{e}_y} \right) + \xi_n \frac{\Delta x_i \Delta y_j}{4} \left(\frac{\psi_{n,1}^{h,\vec{i}} + \psi_{n,5}^{h,\vec{i}}}{2} - \psi_{n,5}^{h,\vec{i}-\hat{e}_z} \right) + \sigma_t \frac{V^{\vec{i}}}{8} \psi_{n,1}^{h,\vec{i}} = \frac{V^{\vec{i}}}{8} S_{n,1}^{h,\vec{i}}, \quad (2.41)$$

where $\psi_{n,l}^{h,\vec{i}} = \left(\vec{\psi}_n^{h,\vec{i}} \right)_l$. In general, the SCB equations can be cast into the form given by [40]:

$$\mu_n \frac{\Delta y_j \Delta z_k}{4} \left(\frac{\psi_{n,l}^{h,\vec{i}} + \psi_{n,l_z}^{h,\vec{i}}}{2} - \psi_{n,l_z}^{h,\vec{i}-s_\mu \hat{e}_x} \right) + \eta_n \frac{\Delta x_i \Delta z_k}{4} \left(\frac{\psi_{n,l}^{h,\vec{i}} + \psi_{n,l_y}^{h,\vec{i}}}{2} - \psi_{n,l_y}^{h,\vec{i}-s_\eta \hat{e}_y} \right) + \xi_n \frac{\Delta x_i \Delta y_j}{4} \left(\frac{\psi_{n,l}^{h,\vec{i}} + \psi_{n,l_x}^{h,\vec{i}}}{2} - \psi_{n,l_z}^{h,\vec{i}-s_\xi \hat{e}_z} \right) + \sigma_t \frac{V^{\vec{i}}}{8} \psi_{n,l}^{h,\vec{i}} = \frac{V^{\vec{i}}}{8} S_{n,l}^{h,\vec{i}}, \quad (2.42)$$

where $l = 1, ..., 8$ and l_x, l_y and l_z is the number of the neighboring corner along the x, y and z directions, respectively, for example if $l = 4$ then $l_x = 3$, $l_y = 1$ and $l_z = 8$. The signs of the direction cosines μ_n, η_n and ξ_n are denoted by s_μ, s_η and s_ξ, respectively.

To the author's knowledge little is known about the accuracy and efficiency of the *SCB* method when compared to the DGFEM methods that it is related to. This might originate in the fact, that SCB was developed primarily with highly diffusive problems in mind, where it is uncommon to have spatially resolved cells in all energy groups and domain regions. The notion of accuracy as understood in the realm of error estimation and analysis requires at least that cells are reasonably resolved which is almost never satisfied for radiative transfer problems. Within this work, a comparison of the SCB and other selected methods will show how much accuracy is lost in the lumping process.

2.3.4 Transverse Moment Type Methods

The transverse moment based (TMB) methods are understood in this work in the spirit of Ref. [13] and [14]. In neutron transport theory this type of method is traditionally labeled nodal method; in fact the TMB methods are nodal methods as defined in this work, but in contrast

to the traditional nomenclature our definition is more general: All TMB methods are nodal methods but not all nodal methods are based on transverse averaging.

TMB methods can be derived for arbitrary expansion orders with the order Λ typically but not always referring to the source expansion as already discussed in Eq. 2.20. Similar to the DD-type methods TMB methods constitute the per mesh-cell system of equations from the spatial Legendre moments of the transport equations Eq. 2.23 augmented by closure/auxiliary relations derived via the transverse moment procedure, followed by an approximate direction-by-direction analytical solution of the resulting 1D transport equation.

TMB methods were originally devised to enable accurate solutions on coarse spatial meshes [14] thus potentially increasing solution efficiency. Common methods at the time such as DD failed miserably because of negative solutions or lack of accuracy on coarse meshes. The idea of TMB methods is to develop closure relations to Eqs. 2.23 that resemble the physics of the transport process closely and the tool to achieve this goal is the transverse moment formalism. The development of the TMB methods is now divided into the AHOTN[13] methods and two additional linear TMB approximations, namely the linear nodal(LN) and the linear-linear (LL) methods[14]. Similar to the discussion of the HODD method the h and $\vec{i}$ superscripts of ψ are omitted.

AHOTN

The arbitrarily high-order transport method of the nodal type (AHOTN) is comprehensively developed in [13] leading to a very compact weighted diamond difference (WDD) representation of the per-cell set of equations. For three-dimensional Cartesian geometry the S_N equations Eq. 1.5 are multiplied by $p_{m_y}(y)p_{m_z}(z)$ and then integrated in the y and z directions over the extent of a single mesh cell, i.e. the operator

$$M^{\vec{i}}_{\vec{m}^x}\{\cdot\} = \frac{1}{\Delta y_j \Delta z_k}\int_{y_{j-1}}^{y_j} dy\; p_{m_y}(y)\int_{z_{k-1}}^{z_k} dz\; p_{m_z}(z) \tag{2.43}$$

is applied to the S_N equations Eq. 1.5. This yields the transverse moments equations for the x-direction:

$$\begin{aligned}
&\mu_n \frac{d}{dx}\psi_{\vec{m}^x}(x) + \sigma_t \psi_{\vec{m}^x}(x) = S_{\vec{m}^x}(x) \\
- \quad &\frac{\eta_n}{\Delta y_j} \underbrace{\left[\psi_{N,m_z}(x) - (-1)^{m_y}\, \psi_{S,m_z}(x) - 2 \sum_{l=0}^{\frac{m_y-1}{2}} (2m_y - 4l - 1)\, \psi_{\vec{m}^x-(2l+1)\hat{e}_2}(x) \right]}_{\chi_{m_y,y}(x)} \\
- \quad &\frac{\xi_n}{\Delta z_k} \underbrace{\left[\psi_{T,m_y}(x) - (-1)^{m_z}\, \psi_{B,m_y}(x) - 2 \sum_{l=0}^{\frac{m_z-1}{2}} (2m_z - 4l - 1)\, \psi_{\vec{m}^x-(2l+1)\hat{e}_3}(x) \right]}_{\chi_{m_z,z}(x)} \\
&\text{for } m_y = 0, ..., \Lambda,\ m_z = 0, ..., \Lambda,
\end{aligned} \tag{2.44}$$

where $\psi_{F,m}$ is a transverse face moment defined by (using $\psi_{N,m_z}(x)$ as example):

$$\psi_{N,m_z}(x) = \frac{1}{\Delta z_k} \int_{z_{k-1}}^{z_k} dz\ p_{m_z}(z) \psi\left(x, y_j, z\right),$$

$\psi_{\vec{m}^x}(x)$ is the transverse nodal moment defined by:

$$\psi_{\vec{m}^x}(x) = \frac{1}{\Delta z_k \Delta y_j} \int_{y_{j-1}}^{y_j} dy \int_{z_{k-1}}^{z_k} dz\ p_{m_z}(z) p_{m_y}(y) \psi\left(\vec{r}\right).$$

and $\chi_{m_y,y}$ and $\chi_{m_z,z}$ are the transverse leakage terms in the y and z directions, respectively. Totally equivalent to Eq. 2.44 transverse integrated transport equations can be derived for the other two directions. Note that so far no approximation has been introduced: Eq. 2.44 is exact but it constitutes a system of ordinary differential equation in the x variable that is not closed since (1) it contains the transverse nodal moments and the transverse face moments on the outflow edges as unknowns and (2) we have not specified how to evaluate the partial source moments given the limited number of available flux moments. We proceed by formally solving Eq. 2.44 using the integrating factor $\frac{1}{\mu_n} \exp\left(\frac{\sigma_t x}{\mu_n}\right)$ and integrating over the extent of the mesh cell in the x-direction:

$$\begin{aligned}
&\frac{\mu_n}{\Delta x_i} \left[\exp\left(\frac{\sigma_t x_i}{\mu_n}\right) \psi_{E,\vec{m}^x} - \exp\left(\frac{\sigma_t x_{i-1}}{\mu_n}\right) \psi_{W,\vec{m}^x} \right] \\
= \quad &\frac{1}{\Delta x_i} \int_{x_{i-1}}^{x_i} dx\ \exp\left(\frac{\sigma_t x}{\mu_n}\right) S_{\vec{m}^x}(x) \\
- \quad &\frac{1}{\Delta x_i} \int_{x_{i-1}}^{x_i} dx\ \exp\left(\frac{\sigma_t x}{\mu_n}\right) \left(\frac{\eta_n}{\Delta y_j} \chi_{m_y,y}(x) + \frac{\xi_n}{\Delta z_k} \chi_{m_z,z}(x) \right).
\end{aligned} \tag{2.45}$$

Similar to Eq. 2.44 Eq. 2.45 is under-determined. Closure is provided by expanding the exponential in the integrals into a finite series of Legendre polynomials[13]:

$$\exp\left(\frac{\sigma_t x}{\mu_n}\right) = \sum_{m_x=0}^{\Lambda} e_{n,m_x} p_{m_x}(x).$$

Substituting the above approximation into Eq. 2.45 results in the following expression:

$$\begin{aligned}
&\frac{\mu_n}{\Delta x_i}\left[\exp\left(\frac{\sigma_t x_i}{\mu_n}\right)\psi^h_{E,\vec{m}^x} - \exp\left(\frac{\sigma_t x_{i-1}}{\mu_n}\right)\psi^h_{W,\vec{m}^x}\right] = \sum_{m_x=0}^{\Lambda} e_{n,m_x} S_{\vec{m}}(x) \\
- &\left\{\sum_{m_x=0}^{\Lambda} e_{n,m_x}\frac{\eta_n}{\Delta y_j}\left[\psi^h_{N,\vec{m}^y} - (-1)^{m_y}\psi^h_{S,\vec{m}^y} - 2\sum_{l=0}^{\frac{m_y-1}{2}}(2m_y-4l-1)\,\psi^h_{\vec{m}-(2l+1)\hat{e}_2}\right]\right. \\
+ &\left.\sum_{m_x=0}^{\Lambda} e_{n,m_x}\frac{\xi_n}{\Delta z_k}\left[\psi^h_{T,\vec{m}^z} - (-1)^{m_z}\psi^h_{B,\vec{m}^z} - 2\sum_{l=0}^{\frac{m_z-1}{2}}(2m_z-4l-1)\,\psi^h_{\vec{m}-(2l+1)\hat{e}_3}\right]\right\} \quad (2.46)
\end{aligned}$$

The last term (in parenthesis) on the right hand side of Eq. 2.45 can be manipulated by using Eqs. 2.23 yielding:

$$\begin{aligned}
&\frac{\mu_n}{\Delta x_i}\left[\exp\left(\frac{\sigma_t x_i}{\mu_n}\right) - \sum_{m_x=0}^{\Lambda} e_{n,m_x}\right]\psi^h_{E,\vec{m}^x} - \left[\exp\left(\frac{\sigma_t x_{i-1}}{\mu_n}\right) - \sum_{m_x=0}^{\Lambda}(-1)^{m_x} e_{n,m_x}\right]\psi^h_{W,\vec{m}^x} \\
- &\sum_{m_x=0}^{\Lambda} e_{n,m_x}\left[\sigma_t\psi^h_{\vec{m}} - \frac{2\mu_n}{\Delta x_i}\sum_{l=0}^{\frac{m_x-1}{2}}(2m_x-4l-1)\,\psi^h_{\vec{m}-(2l+1)\hat{e}_1}\right] = 0. \quad (2.47)
\end{aligned}$$

After several straight forward but lengthy manipulations the WDD form of the AHOTN closure relations can be obtained:

$$\begin{aligned}
&\frac{1+\alpha_{n,x}}{2}\psi^h_{E,\vec{m}^x} + \frac{1-\alpha_{n,x}}{2}\psi^h_{W,\vec{m}^x} = \sum_{m_x=0,even}^{\Lambda}(2m_x+1)\psi^h_{\vec{m}} + \sum_{m_x=0,odd}^{\Lambda}(2m_x+1)\alpha_{n,x}\psi^h_{\vec{m}}, \\
&\text{for } m_y, m_z = 0, 1, .., \Lambda, \quad (2.48)
\end{aligned}$$

where the spatial weight $\alpha_{n,x}$ is given by:

$$\alpha_{n,x} = \frac{\frac{2\mu_n}{\sigma_t\Delta x_i}\left[\cosh\frac{\sigma_t\Delta x_i}{2\mu_n} - \sum_{m_x=0,even}^{\Lambda} e_{n,m_x}\right]}{\frac{2\mu_n}{\sigma_t\Delta x_i}\left[\sinh\frac{\sigma_t\Delta x_i}{2\mu_n} - \sum_{m_x=1,odd}^{\Lambda} e_{n,m_x}\right]}. \quad (2.49)$$

Analogous WDD closure relations exist for the y and z directions. The per-cell system of equations for the AHOTN method is comprised of the spatial moments of the transport equation Eq. 2.23 along with the auxiliary relations Eq. 2.48 for all three dimensions x, y and z. Two things should be noted at this point: (1) the transport physics incorporated via the transverse averaging formalism gets lumped into the spatial weights, otherwise the AHOTN system of equations resembles the standard WDD equations and (2) in the limit of infinitely small cells the AHOTN method becomes identical to the HODD method (pertinent proof section B.4). Thus, on sufficiently fine meshes the solutions obtained with the AHOTN method are indistinguishable from the HODD computed results and consequently the difference of the two methods vanishes and the observed order of accuracy are thus identical.

As the AHOTN method can be conveniently cast into a WDD form with all the AHOTN specifics lumped into the spatial weights, a standard WDD solver can be used to solve the per-cell AHOTN system of equations. Typically, the WDD relations Eq. 2.48 are solved for the outflow face moments and substituted into the nodal balance relations Eq. 2.23 which are then solved for the $(\Lambda+1)^3$ unknown nodal flux moments (NEFD algorithm), [13].

The specific value of the weights computed for a certain optical thickness is the only difference between AHOTN and an arbitrary WDD scheme. Therefore, the spatial weights and their evaluation play a pivotal role in the AHOTN method. For AHOTN there exists one distinct weight per angular direction and spatial dimension. The numerical evaluation of Eq. 2.49 can be performed via a recursion, but Zamonsky[41] found that the spatial weights suffer from round-off instability for optically thin cells and large Λ. He resolves this problem by using asymptotic expansions of the spatial weights for optically thin cells:

$$\alpha_n = \begin{cases} \frac{\vec{t_n^i}}{2\Lambda+3} & \Lambda \text{ even} \\ \frac{2\Lambda+3}{\vec{t_n^i}} & \Lambda \text{ odd} \end{cases} \tag{2.50}$$

An open question regarding the computation of the spatial weights is how much computational effort, compared to the WDD solution, is necessary to compute the spatial weights in a stable and accurate manner.

The AHOTN method in 2D Cartesian geometry is tested in [13] to determine which expansion order results in the most efficient algorithm with the result that the efficiency increases monotonically up to order five. However, Ref. [13] does not examine the performance for a strongly heterogeneous test problem where it is impossible to utilize large cells thus placing higher-order methods at a relative advantage.

Later Azmy[21] and Duo and Azmy[22] used the AHOTN-0,1,2 method to solve variations of Larsen's problem (see also section 2.3.1) and showed by numerical experiment that AHOTN-0 is asymptotically equivalent to DD thus featuring the same observed order of accuracy. A

comparison of the magnitude of the error on coarser, non-asymptotic meshes between AHOTN-0 and DD is neither performed in [21] nor [22] even though the accuracy on realistic, non-asymptotic meshes is more important in practice than the behavior in the asymptotic regime.

The Linear Nodal and Linear-Linear Methods

For practical applications it is expected that a linear TMB approximation is a good compromise for achieving high accuracy at a reasonably short execution time, thus resulting in improved computational efficiency. In Ref. [14] two methods, the linear nodal (LN) and the linear-linear (LL) method, are compared to the AHOTN-1 method; both methods are linear TMB methods but retain less fidelity than the AHOTN-1 method for the purpose of reducing the execution time and required memory.

The LN and LL methods both utilize moments of the balance equations, Eq. 2.23, satisfying $m_x + m_y + m_z \leq 1$ augmented by three WDD equations per dimensions. Thus, the full set of LN and LL equations comprises four balance relations and twelve WDD equations. Within this subsection the WDD equations for the LL and LN method are derived in three-dimensional Cartesian geometry. In particular, the WDD equations for the x-direction will be developed for each of these two methods, but equivalent equations can be derived using the exact same procedure for the y and z-direction.

Applying the operator Eq. 2.43 to the transport equation leads to an ordinary differential equation for $\psi^{h,\vec{i}}_{\vec{m}^x}(x)$:

$$\mu_n \frac{d\psi^{h,\vec{i}}_{\vec{m}^x}}{dx} + \sigma_t \psi^{h,\vec{i}}_{\vec{m}^x}(x) = S^{\vec{i}}_{\vec{m}^x} - \chi^y_{m_y,m_z}(x) - \chi^z_{m_z,m_y}(x), \tag{2.51}$$

where $\chi^k_{m_k,m_p}(x)$ with $k = y$ or $k = z$ is given by:

$$\chi^k_{m_k,m_p}(x) = \frac{\mu_{n,k}}{\Delta_k}\left[\psi^{h,+k}_{m_p} - (-1)^{m_k}\psi^{h,-k}_{m_p} - 2\sum_{l=0}^{[(m_k-1)/2]}(2m_k - 4l - 1)\psi^{h,\vec{i}}_{\vec{m}^x-(2l+1)\hat{e}_k}\right], \tag{2.52}$$

and $+k = N,\ T$ and $-k = S,\ B$. In addition, the indices m_k and m_p are defined as follows: if $k = y$ then $m_k = m_y$ and $m_p = m_z$, while for $k = z$ we have $m_k = m_z$ and $m_p = m_y$. For making Eq. 2.51 amenable to a solution $\chi^k_{m_k,m_p}(x)$'s dependence on x is approximated by:

$$\begin{aligned}
\chi^k_{0,m_p}(x) &\approx \frac{\mu_{n,k}}{\Delta_k}\left[\psi^{h,+k}_{0,m_p} - \psi^{h,-k}_{0,m_p}\right] + \frac{3\mu_{n,k}}{\Delta_k}\left[\psi^{h,+k}_{1,m_p} - \psi^{h,-k}_{1,m_p}\right]p_1(x) \\
\chi^k_{1,m_p}(x) &\approx \frac{\mu_{n,k}}{\Delta_k}\left[\psi^{h,+k}_{0,m_p} + \psi^{h,-k}_{0,m_p} - 2\bar{\psi}^{h,\vec{i}}\right] \\
&+ \lambda\frac{3\mu_{n,k}}{\Delta_k}\left[\psi^{h,+k}_{1,m_p} + \psi^{h,-k}_{1,m_p} - 2\psi^{h,\vec{i}}_{1,0,0}\right]p_1(x),
\end{aligned} \tag{2.53}$$

where $\lambda = 0$ for the LN method and $\lambda = 1$ for the LL method. Further, the source is approximated by:

$$S^{\vec{i}}_{\vec{m}^x} \approx S^{\vec{i}}_{0,\vec{m}^x} + 3\nu S^{\vec{i}}_{1,\vec{m}^x} p_1(x), \tag{2.54}$$

where $\nu = 1$ except if $m_y = m_z = 1$, then $\nu = 0$. Solving the ordinary differential equation Eq. 2.51 using $\exp\left(\frac{x\sigma_t}{\mu_n}\right)$ as integrating factor leads to:

$$\begin{aligned}\exp\left(t^{\vec{i}}_{n,x}\right)\psi^{h,\vec{i}}_{\vec{m}^x,E} \quad - \quad & \exp\left(-t^{\vec{i}}_{n,x}\right)\psi^{h,\vec{i}}_{\vec{m}^x,W} = 2\sinh\left(t^{\vec{i}}_{n,x}\right)\left\{\frac{S^{\vec{i}}_{0,\vec{m}^x}}{\sigma_t} - \frac{\kappa^y_{0,m_y,m_z}}{2t^{\vec{i}}_{n,y}} - \frac{\kappa^z_{0,m_z,m_y}}{2t^{\vec{i}}_{n,z}}\right\} \\ + \quad & 2\left[\cosh\left(t^{\vec{i}}_{n,x}\right) - \frac{\sinh\left(t^{\vec{i}}_{n,x}\right)}{t^{\vec{i}}_{n,x}}\right]\left\{\nu\frac{S^{\vec{i}}_{1,\vec{m}^x}}{\sigma_t} - \frac{\kappa^y_{1,m_y,m_z}}{2t^{\vec{i}}_{n,y}} - \frac{\kappa^z_{1,m_z,m_y}}{2t^{\vec{i}}_{n,z}}\right\},\end{aligned} \tag{2.55}$$

where:

$$\kappa^k_{m_x,m_k,m_p} = \psi^{h,+k}_{\vec{m}^k} - (-1)^{m_k}\psi^{h,-k}_{\vec{m}^k} - 2\sum_{l=0}^{[(m_k-1)/2]}(2m_k - 4l - 1)\psi^{h,\vec{i}}_{\vec{m}-(2l+1)\hat{e}_k}. \tag{2.56}$$

Using Eq. 2.23, the source and transverse leakage terms in Eq. 2.55 can be replaced by volume flux moments and face flux moments on the East and West faces:

$$\begin{aligned}&m_y = m_z = 0: \\ &\frac{1+\alpha_{n,0,x}}{2}\,\psi^{h,E}_{\vec{m}^x} + \frac{1-\alpha_{n,0,x}}{2}\,\psi^{h,W}_{\vec{m}^x} = \bar{\psi}^{h,\vec{i}} + 3\,\alpha_{n,0,x}\,\psi^{h,\vec{i}}_{1,0,0} \\ &m_k = 1,\ m_p = 0: \\ &\frac{1+\alpha_{n,1,x}}{2}\,\psi^{h,E}_{\vec{m}^x} + \frac{1-\alpha_{n,1,x}}{2}\,\psi^{h,W}_{\vec{m}^x} = \psi^{h,\vec{i}}_{0,\vec{m}^x} - 3\lambda\frac{\alpha_{n,1,x}}{t^{\vec{i}}_{n,k}}\left(\psi^{h,+k}_{1,0} + \psi^{h,-k}_{1,0} - 2\psi^{h,\vec{i}}_{1,0,0}\right).\end{aligned} \tag{2.57}$$

The spatial weight $\alpha_{n,l,x}$ is thereby defined as:

$$\alpha_{n,l,x} \quad = \quad \frac{\left[\coth t_{n,x} - \frac{1}{t_{n,x}}\right]}{1 - \frac{\nu_l}{t_{n,x}}\left[\coth t_{n,x} - \frac{1}{t_{n,x}}\right]}, \tag{2.58}$$

where $\nu_0 = 1$ and $\nu_1 = 0$.

For two-dimensional Cartesian geometry a wealth of literature exists comparing the LL, LN and AHOTN-1 methods. Reference [14] shows that the first two methods feature two distinct weights per dimension, per cell and per discrete ordinates while AHOTN-1 only requires the

computation of a single spatial weight. The difference between the LN and LL method is that the LL method retains the bilinear leakage component while the LN neglects it. From an algorithmic (i.e. solution of the local equations within the mesh sweep) point of view the LN provides the least coupling among the set of equations while the LL has stronger coupling in the WDD relations than both LN and AHOTN-1 and AHOTN-1 has stronger coupling than LL and LN in the nodal balance equations. Both LL and LN methods are expected to be less accurate than the AHOTN-1 method since more approximations are made but the hope is that they execute faster leading to superior efficiency. Results in [14] comparing the computational performance of the three methods show that for three test problems the computed results are very close. Execution times vary somewhat with the general conclusion that LN executes fastest, AHOTN-1 is the runner-up and the LL method is slowest. As there is no rigorous computation of the discretization error performed in [14] it is not possible to conclusively decide which method is the most efficient.

For two-dimensional x-y geometry Walters[42] compares computationally the accuracy of the DD, LN, LL and two DGFEM methods, namely the linear discontinuous method and what he refers to as quadratic discontinuous method for a well-logging problem using under-resolved coarse meshes. The quadratic discontinuous method uses polynomial trial functions up to order two in x and y directions but neglects all mixed cross moments (even the $x \cdot y$ term). Walters finds that the LL method is the most accurate of the participating methods followed by the LN method. The FEM methods are found to be of intermediate accuracy while the DD methods, due to the large optical cell thickness, yields unacceptable results. Walters does not rigorously compute the discretization error and also does not measure the execution time of the participating methods, hence no measure of efficiency can be deduced from the stated results. Preceding Ref. [42] Walters and O'Dell[43] presented a comparison of the DD, LN and linear discontinuous method for the ZPPR-7A critical assembly mock up (x-y reactor physics k-eigenvalue problem). The data provided comprises errors and execution times such that the efficiency can be inferred. In general, LD executes in about 66% of the execution time necessary for LN to converge, but LN is more accurate. From the given results, it appears that LN is more efficient on coarse meshes while LD is more efficient on fine meshes.

2.4 Nodal Finite Element Framework

The DGFEM method naturally is a nodal methods as it satisfies all requirements stated in section 2.1 and we readily have the appropriate function spaces available to derive these methods; in fact we started with the function spaces and derived the methods substituting the function spaces into the appropriate weak form. However, more traditional methods that even pre-date the development of a sound theory on nodal methods, can be shown to satisfy all requirements

stated in section 2.1 and therefore are also nodal methods. Within this section the AHOTN and HODD are shown to be nodal methods: while the necessary analysis for the AHOTN method was performed in [1], the findings regarding the HODD method are new. It is not obvious that HODD and AHOTN belong to the class of nodal methods because it is not straightforward to determine appropriate test and trial spaces that when substituted into the strong or weak form return the same set of equations previously derived for each of these methods using physical arguments. However, apart from missing the appropriate function spaces both HODD and AHOTN satisfy the following properties:

- The systems of equations that are solved are local to each mesh cell.
- Communication between mesh cells only occurs via the mesh cells' faces.
- Pointwise continuity of the flux is not enforced.

In the following test and trial functions are derived that yield the HODD and AHOTN equations when substituted into the strong form of the transport equation. This completes the demonstration that HODD and AHOTN are in fact nodal methods. The test and trial spaces can be used as a post-process to reconstruct the flux within each mesh cell for providing accurate interpolation formulae across thick mesh cells.

2.4.1 HODD as Petrov-Galerkin FEM

The HODD method can be derived as a discontinuous Petrov-Galerkin FEM (DPGFEM)[5] by choosing suitable test and trial spaces and utilizing them in the strong form of the S_N transport equation Eq. 2.13. For the following derivation of the HODD method as DPGFEM the flux within a cell is approximated as in Eq. 2.26 and the test space is simply selected to be the space of all Legendre polynomials of order $\vec{m} \leq \Lambda$:

$$\mathcal{V} = \{p_{\vec{m}}, m_x, m_y, m_z = 0, ..., \Lambda\}\,. \tag{2.59}$$

For simplicity let all direction cosines be positive such that the unknown cell quantities in the traditional balance/auxiliary system of equations are the cell Legendre moments and the face Legendre moments on the east, north and top faces. The generic expansion coefficients $\vec{\alpha}$ can

[5] In contrast to DGFEM Petrov-Galerkin FEM do not utilize identical test and trial functions.

be related to the nodal and face flux moments by using their respective definition:

$$\begin{aligned}
\psi^h_{\vec{m}} &= M_{\vec{m}} \left\{ \psi^h \left(\vec{r} \right) \right\} \\
\psi^h_{E,\vec{m}^x} &= \frac{1}{A_E} \left\langle p_{m_y}(y) p_{m_z}(z), \psi^h \left(\vec{r}_E \right) \right\rangle_E \\
\psi^h_{N,\vec{m}^y} &= \frac{1}{A_N} \left\langle p_{m_x}(x) p_{m_z}(z), \psi^h \left(\vec{r}_N \right) \right\rangle_N \\
\psi^h_{T,\vec{m}^z} &= \frac{1}{A_T} \left\langle p_{m_x}(x) p_{m_y}(y), \psi^h \left(\vec{r}_T \right) \right\rangle_T .
\end{aligned} \tag{2.60}$$

It is easy to show that the following linear combination of trial functions:

$$\begin{aligned}
\psi^h \left(\vec{r} \right) &= \sum_{\vec{m}=0}^{\Lambda} \left(2m_x + 1 \right) \left(2m_y + 1 \right) \left(2m_z + 1 \right) \psi^h_{\vec{m}} p_{\vec{m}}(\vec{r}) \\
&+ \sum_{\vec{m}^x=0}^{\Lambda} \left(2m_y + 1 \right) \left(2m_z + 1 \right) \left[\psi^h_{E,\vec{m}^x} - \sum_{m_x}^{\Lambda} \left(2m_x + 1 \right) \psi^h_{\vec{m}} \right] p_{\Lambda+1}(x) p_{m_y}(y) p_{m_z}(z) \\
&+ \sum_{\vec{m}^y=0}^{\Lambda} \left(2m_x + 1 \right) \left(2m_z + 1 \right) \left[\psi^h_{N,\vec{m}^y} - \sum_{m_y}^{\Lambda} \left(2m_y + 1 \right) \psi^h_{\vec{m}} \right] p_{m_x}(x) p_{\Lambda+1}(y) p_{m_z}(z) \\
&+ \sum_{\vec{m}^z=0}^{\Lambda} \left(2m_x + 1 \right) \left(2m_y + 1 \right) \left[\psi^h_{T,\vec{m}^z} - \sum_{m_z}^{\Lambda} \left(2m_z + 1 \right) \psi^h_{\vec{m}} \right] p_{m_x}(x) p_{m_y}(y) p_{\Lambda+1}(z),
\end{aligned} \tag{2.61}$$

satisfies Eq. 2.60. In order to provide more conditions to make the approximation of Eq. 2.61 amenable to solution we constrain the approximate angular flux to be continuous at the incoming edge in an **integral** sense. To this end let the flux on the exterior trace of the incoming faces be expanded in Legendre polynomials (e.g. for the west face):

$$\psi^h_W \left(\vec{r}_W \right) = \sum_{\vec{m}^x=0}^{\Lambda} \left(2m_y + 1 \right) \left(2m_z + 1 \right) \psi_{W,\vec{m}^x} p_{m_y}(y) p_{m_z}(z). \tag{2.62}$$

Then for each inflow face we require that (stating as an example only the west face again) the difference of the interior and the exterior trace is orthogonal to the test space:

$$\left\langle p_{m_y} p_{m_z}, \psi^h \left(\vec{r}_W \right) - \psi^h_W \left(\vec{r}_W \right) \right\rangle_W = \left\langle p_{m_y} p_{m_z}, \left[\left[\psi^h \left(\vec{r} \right) \right] \right]_W \right\rangle_W = 0, \text{ for } m_y, m_z = 0, ..., \Lambda. \tag{2.63}$$

This condition is essentially the same as used in the derivation of the HODD equations via the interpolation problem discussed in section 2.3.1. Evaluation of the difference of the exterior

and interior trace $\psi^h(\vec{r}_W) - \psi^h_W(\vec{r}_W)$ gives:

$$\begin{aligned}\psi^h(\vec{r}_W) - \psi^h_W(\vec{r}_W) &= \sum_{\vec{m}^x=0}^{\Lambda} (2m_y+1)(2m_z+1)\, p_{m_y}(y) p_{m_z}(z) \left[(-1)^{\Lambda+1}\psi^h_{E,\vec{m}^x} - \psi_{W,\vec{m}^x}\right] \\ &+ \sum_{\vec{m}=0}^{\Lambda} (2m_x+1)(2m_y+1)(2m_z+1)\,\psi^h_{\vec{m}} \left[(-1)^{m_x} - (-1)^{\Lambda+1}\right] \\ &+ \text{other terms,} \end{aligned} \tag{2.64}$$

where "other terms" collects terms that comprise either $p_{\Lambda+1}(y)$ or $p_{\Lambda+1}(z)$ and thus are orthogonal to the test functions on the west face. The pointwise difference of the interior and exterior trace is then substituted into Eq. 2.63 giving:

$$\left[(-1)^{\Lambda+1}\psi^h_{E,\vec{m}^x} - \psi^h_{W,\vec{m}^x}\right] + \sum_{m_x=0}^{\Lambda} (2m_x+1)\,\psi^h_{\vec{m}} \left[(-1)^{m_x} - (-1)^{\Lambda+1}\right] = 0, \text{ for } \vec{m}^x = 0..,\Lambda. \tag{2.65}$$

and after some manipulation:

$$\left[\psi^h_{E,\vec{m}^x} - (-1)^{\Lambda+1}\psi^h_{W,\vec{m}^x}\right] = \sum_{m_x=0}^{\Lambda} (2m_x+1)\,\psi^h_{\vec{m}}\,(-1)^{\Lambda+1}\left[(-1)^{\Lambda+1} - (-1)^{m_x}\right], \text{ for } \vec{m}^x = 0..,\Lambda$$

$$\left[\psi^h_{E,\vec{m}^x} - (-1)^{\Lambda+1}\psi^h_{W,\vec{m}^x}\right] = \sum_{m_x=0}^{\Lambda} (2m_x+1)\,\psi^h_{\vec{m}}\left[1 + (-1)^{\Lambda+m_x}\right], \text{ for } \vec{m}^x = 0..,\Lambda,$$

which is identical to the HODD auxiliary relation in the x-direction. Since there are a total of $3(\Lambda+1)^2$ of the constraints Eq. 2.63 the remaining number of unknowns is $(\Lambda+1)^3$. To obtain equations for these unknowns the strong formulation Eq. 2.13 is used in conjunction with the test space $\mathcal{V}$. However, as the term coupling in the difference between interior and exterior trace on the right hand side of Eq. 2.13 is naturally satisfied by ψ^h due to the constraints Eq. 2.63 we only need to require:

$$M_{\vec{m}}\left\{\mu_n\frac{\partial\psi^h}{\partial x} + \eta_n\frac{\partial\psi^h}{\partial y} + \xi_n\frac{\partial\psi^h}{\partial z} + \sigma_t\psi^h(\vec{r}) = S^h(\vec{r})\right\}, \text{ for } \vec{m} = 0...,\Lambda, \tag{2.66}$$

which when evaluated gives the Legendre moments of the balance relations Eq. 2.23. Therefore, the system of equations resulting from utilizing Eq. 2.61 in the strong form of the S_N equations under the constraint of integral continuity is identical to the standard HODD set of balance and auxiliary relations. While the flux is not necessarily continuous across cells, the integral continuity requirement imposes a great deal of rigidity.

2.4.2 AHOTN as DPGFEM Method

Duo[44], [1] found that, similar to the HODD method, the AHOTN method can be derived as a DPGFEM projection using the trial space:

$$\begin{aligned}
\psi^h(x,y,z) &= \sum_{m_x=-1}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=0}^{\Lambda} a_{\vec{m}}\zeta_{m_x}(x)\,p_{m_y}(y)\,p_{m_z}(z) \\
&+ \sum_{m_x=0}^{\Lambda}\sum_{m_y=-1}^{\Lambda}\sum_{m_z=0}^{\Lambda} b_{\vec{m}}\zeta_{m_y}(y)\,p_{m_x}(x)\,p_{m_z}(z) \\
&+ \sum_{m_x=0}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=-1}^{\Lambda} c_{\vec{m}}\zeta_{m_z}(z)\,p_{m_x}(x)\,p_{m_y}(y) \\
&+ \sum_{m_x=0}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=0}^{\Lambda} d_{\vec{m}}p_{m_x}(x)\,p_{m_y}(y)\,p_{m_z}(z),
\end{aligned} \tag{2.67}$$

with the unknown expansion coefficients $a_{\vec{m}}$, $b_{\vec{m}}$, $c_{\vec{m}}$ and $d_{\vec{m}}$ and the function $\zeta_k(x)^{\vec{i}}$ defined as:

$$\begin{aligned}
\xi_{-1}^{\vec{i}}(x) &= e^{-t_n^{\vec{i}}(\hat{x}+1)} \\
\xi_{\lambda}^{\vec{i}}(x) &= t_n^{\vec{i}}\int_{-1}^{\hat{x}} e^{-t_n^{\vec{i}}(\hat{x}-s)}P_\lambda(s)\,ds \\
\hat{x} &= \operatorname{sign}(\mu_n)\,2\frac{x-\frac{x_i+x_{i-1}}{2}}{\Delta x_i}.
\end{aligned} \tag{2.68}$$

The expansion coefficients are further constrained by:

$$\begin{aligned}
M_{\vec{m}}\left\{\sum_{m_x=-1}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=0}^{\Lambda} a_{\vec{m}}\zeta_{m_x}(x)\,p_{m_y}(y)\,p_{m_z}(z)\right\} &= \frac{d_{\vec{m}}}{(2m_x+1)(2m_y+1)(2m_z+1)} \\
M_{\vec{m}}\left\{\sum_{m_x=0}^{\Lambda}\sum_{m_y=-1}^{\Lambda}\sum_{m_z=0}^{\Lambda} b_{\vec{m}}\zeta_{m_y}(y)\,p_{m_x}(x)\,p_{m_z}(z)\right\} &= \frac{d_{\vec{m}}}{(2m_x+1)(2m_y+1)(2m_z+1)} \\
M_{\vec{m}}\left\{\sum_{m_x=0}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=-1}^{\Lambda} c_{\vec{m}}\zeta_{m_z}(z)\,p_{m_x}(x)\,p_{m_y}(y)\right\} &= \frac{d_{\vec{m}}}{(2m_x+1)(2m_y+1)(2m_z+1)},
\end{aligned} \tag{2.69}$$

such that there are in total $(\Lambda+1)^3+3(\Lambda+1)^2$ independent expansion coefficients. The test space is identical to the HODD test space outlined in section 2.4.1. Equivalent to the derivation of the HODD method as DPGFEM the trial space is constrained to satisfy continuity in an integral sense on the inflow edges which yields the $3(\Lambda+1)^2$ WDD relations. Substituting

the trial functions into the strong form Eq. 2.13 and noting that the term accounting for the difference of exterior and interior traces on the inflow edges vanishes for the constrained trial space then gives the nodal balance relations Eq. 2.23.

2.5 Summary of Spatial Discretization Schemes

Three different classes of spatial discretization methods for the transport equation were reviewed: HODD, DGFEM, and TMB methods, each of which features several sub-classes. A fourth class of methods, namely the short characteristics methods, is not mentioned because its extension to three-dimensional Cartesian geometry is complicated, thereby undermining its potential for an efficient algorithm. All described methods are nodal methods as defined in this work and we stated test and trial function spaces for each method, thus reducing the difference between at least the HODD, DGFEM, and the AHOTN methods to the difference in their respective function spaces.

While an extensive body of literature about spatial discretization methods exists, a comprehensive comparison across methods for multi-dimensional Cartesian geometry does not. For slab geometry, Alcouffe et al.[32] performed a comprehensive comparison of methods, but these results cannot be extended to multi-dimensional S_N problems due to the substantially different properties of the underlying exact solutions that fundamentally influence the behavior of numerical schemes to compute approximations to these exact solutions.

Chapter 3

Test Problem Specification

This chapter introduces three test problems that are instrumental in obtaining performance data from the selected spatial discretization methods for the final goal of creating a data-based fitness function measuring the applicability of spatial discretization methods for certain classes of problems. In section 3.1 properties of the exact solution of the S_N equations are reviewed setting the stage for the description of the developed Method of Manufactured solution test suite in section 3.2 which is used for measuring method's accuracy and execution time. The MMS test suite was implemented in the code MMS3D. Subsequently, in section 3.4 Lathrop's test problem is described which is used for measuring method's resilience against negative fluxes. Finally, in section 3.6 a simple test problem is described testing whether a method possesses the thick diffusion limit or not.

3.1 Review of the Exact S_N Solution

In this section we review properties of the underlying exact solution of the S_N equations in multi-dimensional geometry and describe methods for obtaining the exact, or near-exact, solution for simplified S_N transport problems. This is crucial for the remainder of the described work because quantification of the spatial discretization error requires knowledge of the underlying exact solution, or some very accurate approximation thereof. The review within this section is a prelude to the description of the MMS test problem in section 3.2, where the basic concepts introduced within this section are utilized in creating viable test problems that allow for an accurate quantification of the spatial discretization error.

3.1.1 Smoothness of the S_N Exact Solution

In this section properties of the exact solution of the S_N equations are reviewed following the discussion in [1] and [20] for two-dimensional Cartesian geometries which are subsequently

extended to three-dimensional Cartesian geometries. For the discussion in the two spatial dimensions x and y, let us simplify the S_N equations, Eqs. 1.5, by assuming that the total cross section σ_t is a positive constant, and let the medium be non-scattering $\sigma_s = 0$ such that the S_N equations simplify to:

$$\begin{aligned} \hat{\Omega}_n \cdot \nabla \psi_n + \sigma_t(\vec{r})\psi_n(\vec{r}) &= \frac{q(\vec{r})}{4\pi} \text{ for } n = 1,..,N \text{ and } \vec{r} \in \mathcal{D} \\ \psi_n(\vec{r}) &= \psi_B(\vec{r}, \hat{\Omega}_n) \text{ for } n = 1,..,N \text{ and } \vec{r} \in \partial\mathcal{D} \text{ and } \hat{n} \cdot \hat{\Omega} < 0. \end{aligned} \quad (3.1)$$

The set of S_N equations is now decoupled and can be considered on a direction-by-direction basis.

Without loss of generality, let the angular cosines μ_n and η_n be larger than zero, i.e. only angular cosines in the first quadrant are considered. The conclusions of this discussion can easily be extended to the three other quadrants by applying appropriate transformations. Assume further that the imposed external source and the boundary conditions given on the west and south edge, $\psi_{B,W}(y)$ and $\psi_{B,S}(x)$, are smooth, i.e. all partial derivatives are continuous. Then, an analytical solution of the S_N equations for the non-scattering case can be obtained by transforming Eq. 3.1 into the characteristic form

$$\frac{d\psi_n}{ds} + \sigma_t \psi_n \left(x_0 + \mu_n s, y_0 + \eta_n s\right) = \frac{1}{4\pi} q \left(x_0 + \mu_n s, y_0 + \eta_n s\right), \quad (3.2)$$

where $\vec{r}_0 = (x_0, y_0)^T$ is a point on the west or south boundary of the domain.

As illustrated in Fig. 3.1, the solution of the ordinary differential equation 3.2 along the characteristic has to recognize that the angular flux at any point in the domain depends on the boundary condition at $\vec{r}_0$ and the source along the characteristic up to the field point (x, y). For instance the angular flux at $P2$ depends on the value of $\psi_{B,W}(\vec{r}_{P1})$ and the source along the red line, while the angular flux at $P4$ depends on $\psi_{B,S}(\vec{r}_{P3})$ and the source along the green line. Consequently, the domain is divided into two segments, W and S, that are illuminated by the west and south boundary, respectively. The line of demarcation is called the singular characteristic (SC), i.e. the characteristic that emanates from the lower left corner of the domain $\vec{r} = (0, 0)$. The solution within its respective segment can be obtained as:

$$\psi_n(x,y) = \begin{cases} \psi_{B,W}\left(y - \frac{\eta_n}{\mu_n}x\right) e^{-\frac{\sigma_t}{\mu_n}x} + \int_0^{x/\mu_n} ds\ e^{-\sigma_t s} \frac{q(x-\mu_n s, y-\eta_n s)}{4\pi} & y < \frac{\eta_n}{\mu_n}x \\ \psi_{B,S}\left(x - \frac{\mu_n}{\eta_n}y\right) e^{-\frac{\sigma_t}{\eta_n}y} + \int_0^{y/\eta_n} ds\ e^{-\sigma_t s} \frac{q(x-\mu_n s, y-\eta_n s)}{4\pi} & y > \frac{\eta_n}{\mu_n}x \end{cases}. \quad (3.3)$$

While the solution within each segment is clearly smooth, the global solution might exhibit irregularity across the SC. As an example, let $\psi_{B,S} = 0$, $\psi_{B,W} = 1$, and $q(x, y) = 0$. Then the angular flux in the segment illuminated by the south boundary is zero, in contrast to the

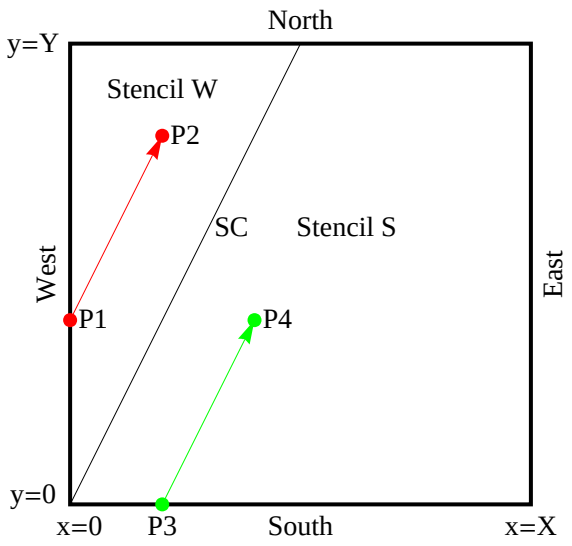


Figure 3.1: Separation of the domain $\mathcal{D}$ by the singular characteristic (SC) into the W and S segments illuminated by an incoming flux on the west and south boundary, respectively.

solution in the W segment, which is non-zero; thus the angular flux is not continuous across the SC.

In the remainder of this work the degree of smoothness of the S_N equations' exact solution is denoted by C_p where p takes integer values $p = 0, 1, ...$ Thereby, p is set equal to the lowest partial derivative order α defined by

$$D^{\alpha}\psi_n = \frac{\partial^{\alpha}\psi_n}{\partial x^{\alpha_x}\partial y^{\alpha_y}\partial z^{\alpha_z}}, \text{ with } \alpha = \alpha_x + \alpha_y + \alpha_z \tag{3.4}$$

such that at least one $D^{\alpha}\psi_n$ is discontinuous across the SC:

$$p = \min \alpha \text{ s.t. } [[D^{\alpha}\psi_n]]_{SC} \neq 0 \text{ for some } \alpha = \alpha_x + \alpha_y + \alpha_z. \tag{3.5}$$

In Eq. 3.5, the jump across the singular characteristic $[[g(x,y)]]_{SC}$ for the generic function $g(x,y)$ is given by

$$[[g]]_{SC} = \lim_{\epsilon \to 0} \left(g\left(x, \frac{\eta_n}{\mu_n}x + \epsilon\right) - g\left(x, \frac{\eta_n}{\mu_n}x - \epsilon\right)\right). \tag{3.6}$$

The first few orders of smoothness C_0, C_1 and C_2 represent a discontinuous angular flux, discontinuous first partial derivatives, and discontinuous second partial derivatives, respectively. The smoothness of the angular flux is determined by the choice of the boundary conditions and

the distributed source.

The consideration for the two-dimensional case is now extended to three spatial dimensions. As depicted in Fig. 3.2 the spatial domain is now separated into three segments, depending on the boundary face from which a point is illuminated. In the remainder of this work, the following naming convention of the boundary faces is used. Without loss of generality, let the domain be given by:

$$\mathcal{D} = [0, X] \times [0, Y] \times [0, Z], \tag{3.7}$$

then the north(N), south(S), east(E), west(W), top(T), and bottom(B) boundary faces are characterized as follows:

$$\begin{aligned}
\text{north: } y &= Y,\ x, z \in [0, X] \times [0, Z] \\
\text{south: } y &= 0,\ x, z \in [0, X] \times [0, Z] \\
\text{east: } x &= X,\ y, z \in [0, Y] \times [0, Z] \\
\text{west: } x &= 0,\ y, z \in [0, Y] \times [0, Z] \\
\text{top: } z &= Z,\ x, y \in [0, X] \times [0, Y] \\
\text{bottom: } z &= 0,\ x, y \in [0, X] \times [0, Y].
\end{aligned} \tag{3.8}$$

Analogous to the two-dimensional case, consider a discrete ordinate with angular cosines in the first octant, i.e. $\mu_n > 0$, $\eta_n > 0$, and $\xi_n > 0$, such that the bottom, west, and south boundary faces are inflow boundaries. The planes of demarcation between the three segments are referred to as the singular planes (SPs) E_x (red), E_y (green), and E_z (blue) that are given by the following parametric forms:

$$\begin{aligned}
E_x &: \ \vec{r} = \lambda\hat{\Omega} + \zeta\hat{e}_x \text{ for } \lambda, \zeta > 0 \\
E_y &: \ \vec{r} = \lambda\hat{\Omega} + \zeta\hat{e}_y \text{ for } \lambda, \zeta > 0 \\
E_z &: \ \vec{r} = \lambda\hat{\Omega} + \zeta\hat{e}_z \text{ for } \lambda, \zeta > 0,
\end{aligned} \tag{3.9}$$

where λ and ζ are the independent characteristic variables.

The E_x plane is the demarcation between the segments illuminated from the bottom and the segment illuminated from the south, while the E_y plane segregates the bottom illuminated segment and the west illuminated segment, and finally E_z separates the west and south illuminated segments. All three SPs intersect in the SC line S given by the parametric form:

$$S : \ \vec{r} = \lambda\hat{\Omega} \text{ for } \lambda > 0. \tag{3.10}$$

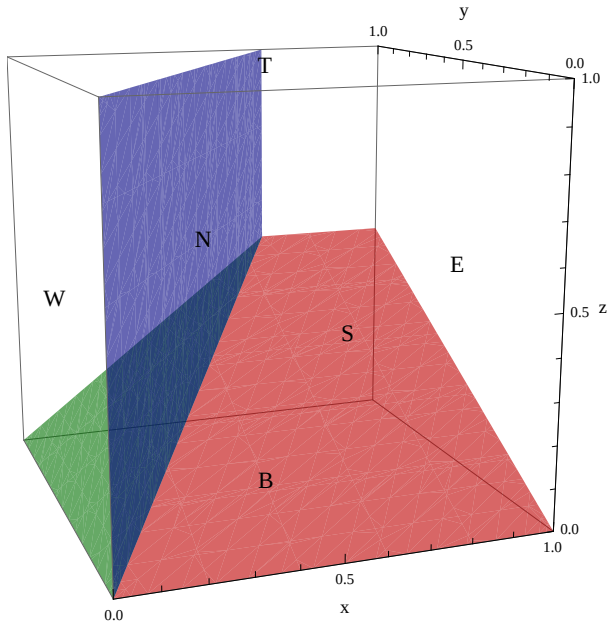


Figure 3.2: Separation of the three-dimensional domain $\mathcal{D}$ by the singular characteristic and the singular planes into three segments illuminated by the west, bottom, and south boundary.

In analogy to the two-dimensional case, the solution of the S_N equations in three-dimensional Cartesian geometry can be determined analytically to be

$$\psi_n(\vec{r}) =$$

$$\begin{cases} \psi_{B,W}\left(y-\frac{\eta_n}{\mu_n}x, z-\frac{\xi_n}{\mu_n}x\right) e^{-\frac{\sigma_t}{\mu_n}x} + \int_0^{x/\mu_n} ds\; e^{-\sigma_t s}\frac{q(\vec{r}-s\hat{\Omega})}{4\pi} & \mathrm{Or}_y(\vec{r})>0,\ \mathrm{Or}_z(\vec{r})<0 \\ \psi_{B,S}\left(z-\frac{\xi_n}{\eta_n}y, x-\frac{\mu_n}{\eta_n}y\right) e^{-\frac{\sigma_t}{\eta_n}y} + \int_0^{y/\eta_n} ds\; e^{-\sigma_t s}\frac{q(\vec{r}-s\hat{\Omega})}{4\pi} & \mathrm{Or}_x(\vec{r})<0,\ \mathrm{Or}_z(\vec{r})>0 \\ \psi_{B,B}\left(x-\frac{\mu_n}{\xi_n}z, y-\frac{\eta_n}{\xi_n}z\right) e^{-\frac{\sigma_t}{\xi_n}z} + \int_0^{z/\xi_n} ds\; e^{-\sigma_t s}\frac{q(\vec{r}-s\hat{\Omega})}{4\pi} & \mathrm{Or}_x(\vec{r})>0,\ \mathrm{Or}_y(\vec{r})<0, \end{cases} \tag{3.11}$$

where the orientation with respect to a singular plane OR_k for $k = x, y, z$ is given by the following expressions:

$$\begin{aligned} \mathrm{Or}_x(\vec{r}) &= \vec{r}^{\,T}\left(\hat{\Omega}\times\hat{e}_1\right) \\ \mathrm{Or}_y(\vec{r}) &= \vec{r}^{\,T}\left(\hat{\Omega}\times\hat{e}_2\right) \\ \mathrm{Or}_z(\vec{r}) &= \vec{r}^{\,T}\left(\hat{\Omega}\times\hat{e}_3\right). \end{aligned} \tag{3.12}$$

The boundary conditions $\psi_{B,W}(y,z)$, $\psi_{B,S}(z,x)$, and $\psi_{B,B}(x,y)$ given on the west, south, and bottom inflow boundary are now functions of the two spatial variables that are not fixed on a boundary face; the respective variables can be inferred from Eq. 3.8.

The solution within each of the three segments is again smooth, but it can exhibit irregularities across the SPs and the SC. For the discussion of the order of smoothness, let the jump across the SPs E_x, E_y and E_z be denoted by:

$$\begin{aligned}
[[g(\vec{r})]]_{E_x} &= \lim_{\epsilon\to 0}\left\{g\left(\lambda\hat{\Omega}+\zeta\hat{e}_1+|\epsilon|\left(\hat{\Omega}\times\hat{e}_1\right)\right)-g\left(\lambda\hat{\Omega}+\zeta\hat{e}_1-|\epsilon|\left(\hat{\Omega}\times\hat{e}_1\right)\right)\right\}\\
[[g(\vec{r})]]_{E_y} &= \lim_{\epsilon\to 0}\left\{g\left(\lambda\hat{\Omega}+\zeta\hat{e}_2+|\epsilon|\left(\hat{\Omega}\times\hat{e}_2\right)\right)-g\left(\lambda\hat{\Omega}+\zeta\hat{e}_2-|\epsilon|\left(\hat{\Omega}\times\hat{e}_2\right)\right)\right\}\\
[[g(\vec{r})]]_{E_z} &= \lim_{\epsilon\to 0}\left\{g\left(\lambda\hat{\Omega}+\zeta\hat{e}_3+|\epsilon|\left(\hat{\Omega}\times\hat{e}_3\right)\right)-g\left(\lambda\hat{\Omega}+\zeta\hat{e}_3-|\epsilon|\left(\hat{\Omega}\times\hat{e}_3\right)\right)\right\}. \qquad (3.13)
\end{aligned}$$

Then, in analogy to the two-dimensional case, we refer to a solution in three-dimensional Cartesian geometry to be smooth to order C_p with $p = 0, 1, 2, ...$ if at least one partial derivative of order p is discontinuous across one of the singular planes:

$$p = \min\alpha \text{ s.t. } [[D^{\alpha}\psi_n]]_{E_k} \neq 0 \text{ for } k = x, y \text{ or } z \text{ and some } \alpha = \alpha_x + \alpha_y + \alpha_z. \qquad (3.14)$$

The smoothness of the exact solution can be controlled by imposing boundary conditions that satisfy certain criteria on the edges common to two inflow faces. In section 3.2 boundary conditions will be developed that render the solution C_p with $p = 0, 1, .., \infty$ smooth.

3.1.2 Obtaining Reference Solutions

A main theme throughout the described work is the estimation of the spatial discretization error associated with an approximate solution of the S_N equations. The discretization error is used for comparing the accuracy of various spatial discretization schemes among each other and for computing their respective rate of convergence. In order to be able to compute the discretization error exactly it is necessary to secure knowledge of the exact solution of the S_N equations. However, in reality it might suffice to obtain an approximation of the true solution that is much more accurate than the solutions whose discretization error we want to compute. In the following, three methods are reviewed that are commonly used to compute exact solutions or highly accurate approximations thereof for the problem of interest: Method of Exact Solution (MES), fine mesh reference solutions and the Method of Manufactured Solution (MMS).

Method of Exact Solutions

For the MES the system of PDEs under consideration (the S_N equations in our case) is solved analytically for a given configuration (domain, boundary conditions, sources) such that a closed

form solution is obtained. The desirable feature of the MES is that the reference solution is exact (at least theoretically) and is known everywhere in phase space. However, MES suffers from two major drawbacks[45]:

- The set of problems that possess an analytical solution is very limited. In general configurations allowing an analytical solution involve simplifications regarding the dimensionality of the problem ((semi)-infinite or symmetric problems), the physics of the problem (e.g. inviscid flow, [46]) or the decoupling of the equations among others. Thus, the exact solution does not exercise the full range of capabilities implemented in a code and therefore error estimation is not possible in all of the code's applicable regimes.

- The obtained exact solutions are often given in terms of special mathematical functions (exponential integrals, gamma functions, inverse Laplace transforms etc.) and are therefore often difficult to implement. Consequently, the numerical values resulting from the evaluation of the analytical expressions might be inaccurate thus invalidating the primary advantage of the MES.

In particular the MES found several applications for S_N problems in one-dimensional slab geometry and two-dimensional Cartesian geometry. Because of vastly different properties of the exact solution as well as the mechanics to obtain it, work done in slab-geometry and two-dimensional Cartesian geometry has barely any overlap. Exact solutions for slab-geometry transport problems can be obtained in the presence of scattering[4], while that is impossible for two-dimensional Cartesian problems. In addition, the exact angular flux in slab-geometry transport problems is smooth, i.e. it possesses an infinite number of bounded partial derivatives, because the S_N equations in one-dimensional slab geometry are merely "a coupled system of first-order ordinary differential equations with constant coefficients"[47]; in marked contrast the underlying exact angular flux for two-dimensional transport problems usually features limited smoothness, [20]. For this reason, we skip the discussion of slab geometry problems and only discuss two-dimensional Cartesian geometry in this work.

Two-dimensional Cartesian transport problems allow exact solution of the S_N equations only in the absence of scattering. The solution process was outlined in subsection 3.1.1 and the solution for general boundary conditions and sources is given in Eq. 3.3. The first application of Eq. 3.3 was in [20] where it was used to demonstrate the generally limited smoothness of the exact angular flux solution. Later, Larsen[18] used a yet more simplified configuration featuring unit incoming angular fluxes on the west and south boundary edges and a vanishing source for demonstrating the reduction of the convergence order of the Diamond Difference scheme due to the limited smoothness of the underlying exact solution. In this configuration the solution simplifies to:

$$\psi(x,y) = e^{-\sigma \min\left(\frac{x}{\mu},\frac{y}{\eta}\right)}.$$

Table 3.1: Boundary Conditions and the resulting smoothness used in published variations of Larsen's benchmark.

Boundary Conditions	Smoothness	Reference
$\psi_L = 0$, $\psi_B = 1$ $\psi_L = 1$, $\psi_B = 0$	C_0	[21], [22], [1]
$\psi_L = 1$, $\psi_B = 1$	C_1	[18], [21], [22], [1]
$\psi_L = y^2$, $\psi_B = x^2$	C_2	[1]
$\psi_L = y^2$, $\psi_B = x^2$, $\eta = \mu$	C_3	[1]

This test case shall be referred to as Larsen's benchmark in the remainder of this work. Note, that [18] uses the solution only along a single discrete ordinate such that μ and η bear no indices. This is inconsequential for non-scattering media since the various discrete ordinates' equations are decoupled.
Based on [18], Azmy[21], Duo and Azmy[22] and Duo[1] created variations of Larsen's benchmark with different levels of smoothness of the underlying solution for the purpose of performing error analysis on various spatial discretization schemes all featuring a constant representation of the source. Within [21] and [22] three variations were considered: Zero incoming flux on the west and unit incoming flux on the south boundary, unit incoming flux on the west and zero incoming flux on the south boundary and finally Larsen's original setup featuring unit incoming flux on both west and south boundary. Later, Duo[1] expanded on the earlier variations by prescribing boundary conditions varying like x^2 (y^2) along the south edge (west edge); using these boundary conditions the smoothness of the exact solution can be increased to C_2 and C_3, respectively. For a comprehensive presentation of all previously reported variations of Larsen's benchmark consult Table 3.1. The exact solutions employed in Refs. [18], [21], [22] and [1] are extremely valuable (and in fact represents the starting point for the development of the MMS test suite in this work) but they do not not allow for scattering and are therefore not general enough for the purpose of this work. In addition, the setups in the above references do not necessitate the computation of higher order spatial flux moments, only the average angular flux needs to be computed. Therefore, it is necessary to extend the formalism in these references to suit the requirements of this work.

Fine Mesh Reference Solution

The idea behind using a fine mesh reference solution is to solve the underlying system of equations on a very fine mesh and then utilize this solution to compute the discretization error

on coarser meshes featuring a much larger error, e.g. [11]. The underlying assumption of this method is that the discretization error in the reference solution is too small to pollute the computation of the discretization error on the coarse mesh. However, as demonstrated in [1], the discretization error in the reference solution invalidates the computed discretization error for mesh refinement levels that are too close to the reference solution's mesh. Reference [1] states that convergence of the fine mesh reference solution to the exact angular flux cannot be guaranteed "unless the method is fully consistent and the asymptotic regime has been reached"[1] thus challenging the rigor behind using fine mesh reference solutions. If for example the discretization error measured in the $\mathcal{L}_\infty$ norm is desired and the exact angular flux is not continuous, the computed error using a fine mesh reference solution is meaningless because the reference solution does not converge to the exact solution in the $\mathcal{L}_\infty$ norm. Even if this pathological example is avoided, the limited smoothness of the S_N solution causes the reference solution to suffer from a limited rate of convergence so the error even on fine meshes can be inaccurate[48].

Despite these concerns, fine mesh reference solutions can be successfully used for the computation of discretization errors if the reference solutions' discretization error is negligible, e.g. by using an adaptive mesh refinement procedure (AMR)[11]. However, this requires access to a robust AMR procedure which offsets the main advantage of the fine mesh reference solution, namely its simplicity.

Method of Manufactured Solutions

The first application (to our knowledge) of the MMS within the realm of radiative transport theory dates back to 1971 by Lingus[19]. However, the application is much more limited than the general treatment in [45] and [46] which are not specific to neutron/radiative transport theory. Following the path laid out in [45] the PDE is given as some (differential) operator (also containing boundary conditions) $\mathbf{D}$ operating on the solution vector $\vec{u}$ to yield some source vector $\vec{g}$. The forward way of solving this problem is to find $\vec{u}$ to some given $\vec{g}$, while for the purpose of the MMS we select $\vec{u} = \vec{u}_M$ and compute the corresponding source vector $\vec{g}_M$ by:

$$\vec{g}_M = \mathbf{D}\vec{u}_M.$$

If we solve the problem using the numerical method under scrutiny, with an appropriate discretization of the source vector $\vec{g}_M^h$, we obtain an approximation $\vec{u}^h$ of the true solution:

$$\mathbf{D}^h\vec{u}^h = \vec{g}_M^h \;\Rightarrow \vec{u}^h = \left(\mathbf{D}^h\right)^{-1}\vec{g}_M^h,$$

and since we know the solution $\vec{u}_M$ we can compute the discretization error by $\vec{u}^h - \vec{u}_M$.

Within the realm of neutron/radiative transport theory the MMS was mostly used for computer code verification [49], [50] and [51]. In Ref. [49] Pautz develops MMS for the verification of the ATTILA code. In marked contrast to the approach adopted in this work, Pautz' manufactured solutions are not restricted to the one-group S_N equations. They are not only an MMS in the spatial variables but rather depend on energy, direction and space. Thus, his MMS exercises the energy, angular and spatial discretization and consequently any computed discretization error is, in general, a combination of the error from the energy, angular and spatial discretization. However, he designes several variants of his MMS such that a subset of the errors vanishes; some test cases are even designed so as to be solved exactly by the selected discretization schemes. Later Pautz used the MMS to verify the SCEPTRE code [50], this time only verifying that the one-group problem is solved correctly, i.e. the manufactured solutions are independent of energy. The first manufactured solution is non-zero only along a given discrete ordinate and varies like a polynomial in space. It is exactly solved by the utilized FEM method if (1) the trial functions' span encompasses the utilized polynomial and (2) the discrete ordinate along which the angular flux is non-zero is part of the quadrature rule. Subsequently, Pautz uses functions that vary exponentially in space and linearly/quadratically in angle; for these manufactured solutions the S_N/FEM solution will comprise a discretization error. Finally, Drumm [51] uses a manufactured solution that is very similar to the second type used in [50], yet not identical. He uses these manufactured solutions to investigate order of convergence anomalies occurring for second order S_N methods.

The three main differences between the MMS that is developed in this work and the MMS used in Refs. [49], [50] and [51] is:

1. We adopt the S_N approximation within the manufactured solution such that when comparing numerical and reference solutions only the spatial discretization error is quantified.

2. The developed manufactured solutions are potentially non-smooth across the singular characteristic (SC) or singular planes (SPs), while Pautz' and Drumm's manufactured solutions are all smooth, i.e. they possess an infinite number of bounded partial derivatives.

3. Pautz' manufactured solutions usually contain negative sources while ours typically feature a strictly positive source but can be tuned to comprise negative sources as well.

The approach used throughout this work to develop the MMS test suite is strongly rooted in the works of Larsen, Azmy and Duo described in subsection 3.1.2. In fact, Duo [17], [1] originally suggested the method to derive manufactured solutions that is adopted in this work and implemented it for a small number of test cases in 2D. The ultimate goal is to manufacture

Table 3.2: Variants of the MMS benchmark suggested by Duo[1]. Note, Duo uses a different normalization of the angular weights so he lacks the $1/4\pi$ factor.

Smoothness	Boundary Conditions	Auxiliary Source Q
C_0	$\psi(0,y) = \psi_W$ $\psi(x,0) = \psi_S$	$\frac{Q}{4\pi} > \frac{w_1 \sigma_s \max(\psi_W, \psi_S)}{1-\frac{\sigma_s}{\sigma_t}}$
C_1	$\psi(0,y) = 0$ $\psi(x,0) = 0$	$\frac{Q}{4\pi} = 1$

a solution for the S_N equations (compare Eq. 1.5):

$$\hat{\Omega}_n \cdot \nabla\psi(\vec{r},\hat{\Omega}_n) + \sigma(\vec{r})\psi(\vec{r},\hat{\Omega}_n) = \frac{q(\vec{r},\hat{\Omega}_n)}{4\pi} + \frac{1}{4\pi}\sum_{n=1}^{N} w_n \sigma_s(\vec{r})\psi(\vec{r},\hat{\Omega}_n), \tag{3.15}$$

with boundary conditions given on the inflow boundaries:

$$\psi(\vec{r},\hat{\Omega}_n) = \psi_B(\vec{r}). \tag{3.16}$$

To this end, an auxiliary, non-scattering problem featuring a constant distributed source and the same boundary conditions as the original problem is solved analytically for the exact angular flux $\psi(\vec{r},\hat{\Omega}_n)$:

$$\hat{\Omega}_n \cdot \nabla\psi(\vec{r},\hat{\Omega}_n) + \sigma\psi(\vec{r},\hat{\Omega}_n) = \frac{Q}{4\pi}. \tag{3.17}$$

Then the source of the S_N transport problem is computed according to the MMS procedure:

$$\begin{aligned} q(\vec{r},\hat{\Omega}_n) &= 4\pi\left(\hat{\Omega}_n \cdot \nabla\psi(\vec{r},\hat{\Omega}_n) + \sigma(\vec{r})\psi(\vec{r},\hat{\Omega}_n)\right) - \sum_{n=1}^{N} w_n \sigma_s(\vec{r})\psi(\vec{r},\hat{\Omega}_n) \\ &= Q - \sum_{n=1}^{N} w_n \sigma_s(\vec{r})\psi(\vec{r},\hat{\Omega}_n). \end{aligned} \tag{3.18}$$

The analytical solution of Eq. 3.17 subject to boundary conditions Eq. 3.16 is then a solution of the original S_N problem Eq. 3.15 subject to boundary conditions given by Eq. 3.16 and a distributed source defined by Eq. 3.18. Duo proposes and implements two variations of this MMS with parameters given in Table 3.2.

Summary

The properties of the S_N equations' exact solution in slab geometry on the one hand and multi-dimensional geometry on the other hand are vastly different. Therefore, research in slab geometry regarding spatial discretization schemes cannot be extended to multi-dimensional geometries. In multi-dimensional geometries analytical reference solutions exist only in the absence of scattering leaving only the MMS as a reasonable option for securing knowledge of the underlying exact solution thereby enabling an accurate computation of the numerical solution's error. Finally, comparing Duo's [17] and Pautz'[50] manufactured solutions, consistency with the physical meaning of the source (at least for error estimation) demands positive sources and in addition error estimation for S_N transport problems requires the solution to exhibit limited smoothness otherwise the results will not reflect reality. Therefore, Duo's approach seems the most promising for securing knowledge of the exact solution of the S_N equations in realistic configurations; for the purpose of this work its extension to three-dimensional Cartesian geometry and computation of arbitrary polynomial order moments of the source, boundary conditions, and exact solution is necessary.

3.2 Method of Manufactured Solutions Test Suite

In this section the Method of Manufactured Solutions test suite in three-dimensional geometry is introduced before it is used for comparing the accuracy of the selected spatial discretization methods. The MMS test suite is based on the exact solution of the non-scattering S_N equations stated in Eq. 3.11; in this section it will be shown that by choosing appropriate boundary conditions the smoothness of the exact solution can be controlled. In the remainder of the section the implementation of the MMS suite is discussed including the tracking of the singular characteristic line (SC) and singular planes (SPs) and the computation of the exact angular and scalar fluxes and sources Legendre moments.

3.2.1 Smoothness of the constructed Solution

Using Eq. 3.11 as a starting point we simplify the construction of the test suite by assuming that the source in the domain is uniform and equal to $Q/4\pi$. Also, recall that the medium is homogeneous $\sigma_t(\vec{r}) = \sigma_t$. Then the exact solution of the non-scattering S_N equations is:

$$\psi_n(\vec{r}) = \begin{cases} \psi_{B,[W,E]}\left(\bar{y} - \left|\frac{\eta_n}{\mu_n}\right|\bar{x}, \bar{z} - \left|\frac{\xi_n}{\mu_n}\right|\bar{x}\right) e^{-\frac{\sigma_t}{|\mu_n|}\bar{x}} + \frac{Q}{4\pi\sigma_t}\left(1 - e^{-\frac{\sigma_t}{|\mu_n|}\bar{x}}\right) & \mathrm{Or}_y(\vec{r}) > 0,\ \mathrm{Or}_z(\vec{r}) < 0 \\ \psi_{B,[S,N]}\left(\bar{z} - \left|\frac{\xi_n}{\eta_n}\right|\bar{y}, \bar{x} - \left|\frac{\mu_n}{\eta_n}\right|\bar{y}\right) e^{-\frac{\sigma_t}{|\eta_n|}\bar{y}} + \frac{Q}{4\pi\sigma_t}\left(1 - e^{-\frac{\sigma_t}{|\eta_n|}\bar{y}}\right) & \mathrm{Or}_x(\vec{r}) < 0,\ \mathrm{Or}_z(\vec{r}) > 0 \\ \psi_{B,[B,T]}\left(\bar{x} - \left|\frac{\mu_n}{\xi_n}\right|\bar{z}, \bar{y} - \left|\frac{\eta_n}{\xi_n}\right|\bar{z}\right) e^{-\frac{\sigma_t}{|\xi_n|}\bar{z}} + \frac{Q}{4\pi\sigma_t}\left(1 - e^{-\frac{\sigma_t}{|\xi_n|}\bar{z}}\right) & \mathrm{Or}_x(\vec{r}) > 0,\ \mathrm{Or}_y(\vec{r}) < 0 \end{cases} \tag{3.19}$$

where the subscripts $[W,E]$, $[S,N]$ and $[B,T]$ indicate that depending on the sign of the direction cosines μ_n, η_n and ξ_n, respectively, the appropriate (inflow) face from each of the three sets of parallel faces is selected. The three coordinates $\bar{x}$, $\bar{y}$ and $\bar{z}$ extend the solution of the non-scattering S_N equations to all eight angular octants and are given by:

$$\begin{aligned} \bar{x} &= \frac{1 - (\text{sign}\ )\,\mu_n}{2} X + \text{sign}\ (\mu_n)\, x \\ \bar{y} &= \frac{1 - (\text{sign}\ )\,\eta_n}{2} Y + \text{sign}\ (\eta_n)\, y \\ \bar{z} &= \frac{1 - (\text{sign}\ )\,\xi_n}{2} Z + \text{sign}\ (\xi_n)\, z. \end{aligned} \tag{3.20}$$

For later reference it should be noted that Eq. 3.19 is the solution of the following S_N transport problem:

$$\begin{aligned}
\hat{\Omega}_n \cdot \nabla \psi_n + \sigma_t \psi_n \left(\vec{r}\right) &= \frac{Q}{4\pi} \\
\psi_n(\vec{r}_{[W,E]}) &= \psi_{B,[W,E]}(y,z) \\
\psi_n(\vec{r}_{[S,N]}) &= \psi_{B,[S,N]}(z,x) \\
\psi_n(\vec{r}_{[B,T]}) &= \psi_{B,[B,T]}(x,y),
\end{aligned} \tag{3.21}$$

where $\psi_{B,[W,E]}$, $\psi_{B,[S,N]}$ and $\psi_{B,[B,T]}$ are fixed boundary conditions. The choice of boundary conditions determines the smoothness of the exact solution defined via Eqs. 3.13 and 3.14 of the test problem. For an easier implementation of the MMS test suite we restrict all boundary conditions to be polynomials in the two degrees of freedom on the respective face, e.g. y and z on the W and E face, which still allows an arbitrary degree of smoothness p with the exception of the $p = \infty$ case which will be treated independently. Thus, let the boundary conditions for the west/east surface be given by:

$$\psi_{B,[W,E]} = \sum_{l_y=0}^{L_y} \sum_{l_z=0}^{L_z} a_{l_y,l_z}^{[W,E]} \bar{y}^{l_y} \bar{z}^{l_z}, \tag{3.22}$$

and let analogous expressions hold on the other two inflow faces. Boundary conditions that render the solution C_0 to C_3 are compiled in Table 3.3. Using the general expression for the boundary conditions on the west/east faces, Eq. 3.22, the exact angular flux in the solution segment illuminated by the west/east faces is given by

$$\psi_n \left(\vec{r}\right) = \left(\sum_{l_y=0}^{L_y} \sum_{l_z=0}^{L_z} a_{l_y,l_z}^{[W,E]} \left(\bar{y} - \left| \frac{\eta_n}{\mu_n} \right| \bar{x} \right)^{l_y} \left(\bar{z} - \left| \frac{\xi_n}{\mu_n} \right| \bar{x} \right)^{l_z} - \frac{Q}{4\pi\sigma_t} \right) e^{-\frac{\sigma_t}{|\mu_n|}\bar{x}} + \frac{Q}{4\pi\sigma_t}. \tag{3.23}$$

along with equivalent expressions for the other two segments. For the $p = \infty$ case the boundary conditions are chosen such that the flux within each segment becomes:

$$\psi_n \left(\vec{r}\right) = C e^{-\sigma_t \left(\frac{\bar{x}}{|\mu_n|} + \frac{\bar{y}}{|\eta_n|} - \frac{\bar{z}}{|\xi_n|} \right)} + \frac{Q}{4\pi\sigma_t}, \tag{3.24}$$

which can be achieved by using the boundary conditions listed in Table 3.3. Note that the exact angular flux expression for $p = \infty$ features a positive argument in the exponential for the z-dimension such that ψ_n grows exponentially with increasing $\bar{z}$. This is computationally undesirable but cannot be avoided because no boundary conditions exist that render the argument of the exponential in Eq. 3.24 to be $\sigma_t \left(\frac{\bar{x}}{|\mu_n|} + \frac{\bar{y}}{|\eta_n|} + \frac{\bar{z}}{|\xi_n|} \right)$. Even though selecting the

Table 3.3: Expressions for the boundary conditions and conditions that must be satisfied by the user-selected coefficients to ensure positivity of the distributed source for the MMS in three-dimensional Cartesian geometry that render the solution C_0 through C_3 and C_∞.

p	$\psi_{[W,E]}$	$\psi_{[S,N]}$	$\psi_{[B,T]}$
C_0	$a_{0,0}^{[W,E]}$	$a_{0,0}^{[S,N]}$ $a_{0,0}^{[W,E]} \neq a_{0,0}^{[S,N]} \neq a_{0,0}^{[B,T]} \neq \frac{Q}{\sigma_t}$	$a_{0,0}^{[B,T]}$
C_1	$a_{0,0}^{[W,E]}$	$a_{0,0}^{[S,N]}$ $a_{0,0}^{[W,E]} = a_{0,0}^{[S,N]} = a_{0,0}^{[B,T]} \neq \frac{Q}{\sigma_t}$	$a_{0,0}^{[B,T]}$
C_2	$\frac{Q}{4\pi\sigma_t} + a_{2,2}^{[W,E]}\bar{y}^2\bar{z}^2$	$\frac{Q}{4\pi\sigma_t} + a_{2,2}^{[S,N]}\bar{z}^2\bar{x}^2$	$\frac{Q}{4\pi\sigma_t} + a_{2,2}^{[B,T]}\bar{x}^2\bar{y}^2$
C_3	$\frac{Q}{4\pi\sigma_t} + a_{3,3}^{[W,E]}\bar{y}^3\bar{z}^3$	$\frac{Q}{4\pi\sigma_t} + a_{3,3}^{[S,N]}\bar{z}^3\bar{x}^3$	$\frac{Q}{4\pi\sigma_t} + a_{3,3}^{[B,T]}\bar{x}^3\bar{y}^3$
C_∞	$\frac{Q}{4\pi\sigma_t} + ae^{-\frac{\sigma_t}{\lvert\mu_n\rvert}\bar{x}+\frac{\sigma_t}{\lvert\xi_n\rvert}\bar{z}}$	$\frac{Q}{4\pi\sigma_t} + ae^{-\frac{\sigma_t}{\lvert\eta_n\rvert}\bar{y}+\frac{\sigma_t}{\lvert\xi_n\rvert}\bar{z}}$	$\frac{Q}{4\pi\sigma_t} + ae^{-\frac{\sigma_t}{\lvert\mu_n\rvert}\bar{x}+\frac{\sigma_t}{\lvert\eta_n\rvert}\bar{y}}$

appropriate boundary conditions allows for controlling the smoothness of the underlying exact solution it is noteworthy that only the C_0 (constant but different inflow flux on each face e.g. shielding problem with shadowing) and C_1 (vacuum boundary condition and distributed source, e.g. reactor physics like problems) cases are realistic; the $C_p,\ 1 < p < \infty$ cases are only interesting from an academic point of view and the $p = \infty$ case is interesting for computer code verification.

3.2.2 Construction of a Manufactured Solution with Scattering

The angular fluxes Eqs. 3.23 and 3.24 are solutions to the non-scattering S_N transport problem Eq. 3.21. In order to construct manufactured solutions to the general S_N equations, Eq. 1.5 is solved for the distributed source $q\left(\vec{r}\right)$:

$$q(\vec{r}) = \hat{\Omega}_n \cdot \nabla\psi_n + \sigma_t(\vec{r})\psi_n(\vec{r}) - \frac{1}{4\pi}\sigma_s(\vec{r})\phi_N(\vec{r}) \tag{3.25}$$

Following the standard MMS formalism the exact solution is now selected and substituted into Eq. 3.25. Since our manufactured solutions satisfy $Q = \nabla\psi_n + \sigma_t(\vec{r})\psi_n(\vec{r})$ we obtain the following prescription for the distributed source q:

$$q(\vec{r}) = Q - \sigma_s(\vec{r})\phi_N(\vec{r}). \tag{3.26}$$

The manufactured solution to the S_N problem **with** scattering can now be stated as follows: The angular flux Eq. 3.23 ($p < \infty$) or 3.24 ($p = \infty$) is the solution of problem Eq. 1.5 if the source is computed by Eq. 3.26. The distributed source can, under certain circumstances, become negative (see Eq. 3.26) which is inconsistent with the physical meaning of the source and might lead to problems with using such sources in a computer code if the input of the code is restricted to positive sources. Therefore, it should be an objective to select manufactured solutions such that the source is always positive. For the purpose of this work "positive" sources are understood to be positive everywhere $q(\vec{r}) > 0\ \vec{r} \in \mathcal{D}$ as opposed to requiring that the cell-averaged sources are positive $M_0^{\vec{i}}\{q\} > 0,\ \vec{i} \in \mathcal{D}$. Only the latter condition is usually checked for by computer codes; in fact pointwise positivity is a sufficient but not a necessary condition for positive cell-averaged sources. However, if a given point source is negative for some $\vec{r} \in \mathcal{D}$ but does not feature negative cell-averaged sources on a given mesh, then sufficient mesh refinement will lead to negative cell-averaged sources because negative "patches" are successively isolated until they are not offset by positive source regions in their vicinity.

3.2.3 Implementation of the MMS Test Suite

In order to generate data that can be used for computing the discretization error of spatial discretization methods of the S_N equations we compute cell Legendre moments of the angular and scalar flux given by:

$$\begin{aligned} \psi_{n,\vec{m}}^{\vec{i}} &= M_{\vec{m}}^{\vec{i}}\{\psi_n(\vec{r})\} \\ \phi_{\vec{m}}^{\vec{i}} &= M_{\vec{m}}^{\vec{i}}\{\phi(\vec{r})\} = \sum_{n=1}^{N} w_n \psi_{n,\vec{m}}^{\vec{i}} \\ q_{\vec{m}}^{\vec{i}} &= M_{\vec{m}}^{\vec{i}}\{q(\vec{r})\} = Q\delta_{\vec{m},\vec{0}} - \sigma_s \phi_{\vec{m}}^{\vec{i}}. \end{aligned} \tag{3.27}$$

However, this is not as easy as it might seem at first glance because a cell can be intersected by the SC **or** one of the SPs **or** can be completely within a single segment. In order to determine which cells are intersected by the SC (type I) and which by the SPk (k=x,y,z) (type II) a tracking procedure is devised, which returns a list of cells for each intersection type along with a list of points within this cell that delimit the segments each of which is illuminated by a different domain boundary, i.e. the convex hull[1]. The convex hull is then used for tessellating the cell into a set of tetrahedra (type I) or triangular prisms (type II intersections) to facilitate the integration procedure which effects the operations denoted via $M_{\vec{m}}^{\vec{i}}\{\cdot\}$. Note, that for the C_∞ case no tracking or tessellation needs to be performed because the flux follows the same mathematical expressions regardless of which boundary face a point is illuminated by.

[1]The convex hull is the minimal convex set of points containing the tetrahedron.

Tracking the SC and the SPs

The tracking procedure is broken up into two subtasks: First the intersection of the SC (see red solid line in Fig. 3.3) with all mesh cells within the domain is computed followed by the computation of the intersections for the SPs using tracking information from the projection of the SC onto the boundary face as depicted in Fig. 3.3 (dashed red line). The intersections of the SP is inferred from the tracking data of the SC and its projection.

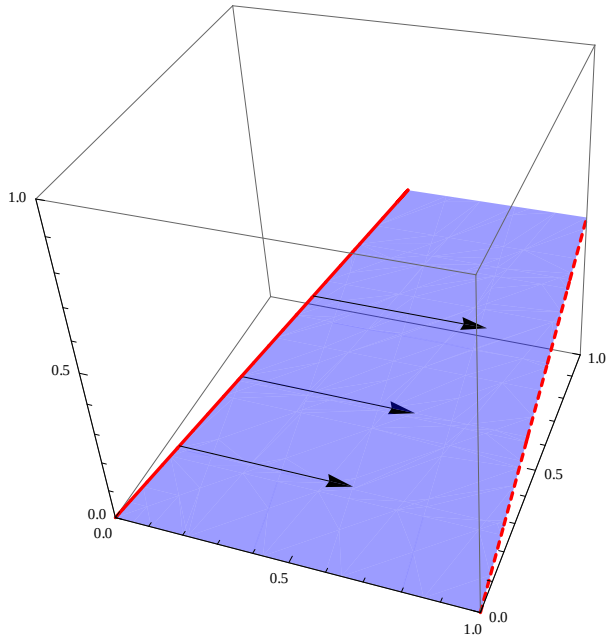


Figure 3.3: Schematic illustration of the tracking procedure. First the intersection of the SC (red solid line, along $\hat{\Omega}_n$) with all cells is computed followed by tracking the intersection of the projection of the SC onto the far y-z plane (dashed red line). The intersections of the SP (light blue) with the mesh cells is then inferred from this data. The direction of particle motion $\hat{\Omega}_n$ is a unit vector along the SC (red solid line).

Tracking the SC

Computing the intersection of the singular characteristic line with every face for every computational cell is computationally intractable since it scales with the total number of cells. It is computationally more efficient to follow the SC through the domain, i.e. to start at the corner cell where the SC originates, then compute the face (or in the degenerate case the edge or corner point) where the SC leaves the mesh cell and increment the cell index depending on

which face/edge/point is intersected. This basic step is repeated for each mesh cell that the SC intersects. Thus, the execution time of the tracking does not scale with the total number of cells but rather with their cubic root. The algorithm used within the implementation of the MMS test suite is detailed in algorithm 1. Note that particular care is taken to identify degenerate intersections, i.e. the SC intersects with the corner point that borders three outflow faces or one of the edges that border two outflow faces, by assuming that those lower dimensional entities have a finite thickness related to the machine precision ϵ_{mach}. In loose terms, one could think to these as "fat" points and edges. From the SC tracking we obtain a list of cells that are intersected by the SC along with the points where the SC enters and leaves the respective cell.

Tracking the SPs

The first step for obtaining the tracking information for the singular planes is to track the projection of the SC along the x, y and z-direction onto the domain's far face for the SPx, SPy and SPz planes, respectively. For a single SP this projection is depicted in Fig. 3.3 as a dashed red line. The utilized algorithm 2 is a two-dimensional equivalent of algorithm 1. Algorithm 2 references the two coordinates u and v and the two direction cosines μ_u and μ_v which are related to the standard coordinates x,y and z as follows:

$$\begin{aligned}
\text{SPx:} \quad & u \leftarrow y,\ v \leftarrow z,\ \mu_u \leftarrow \eta_n\ \mu_v \leftarrow \xi_n \\
\text{SPy:} \quad & u \leftarrow z,\ v \leftarrow x,\ \mu_u \leftarrow \xi_n\ \mu_v \leftarrow \mu_n \\
\text{SPz:} \quad & u \leftarrow x,\ v \leftarrow y,\ \mu_u \leftarrow \mu_n\ \mu_v \leftarrow \eta_n.
\end{aligned}$$

After executing algorithm 2 the indices of cells crossed by the projected SC in the u-v plane as well as the coordinates where the projected SC enters and leaves a two-dimensional cell are known. For a given pair of indices $i_u,\ i_v$ (varying orthogonal to the projection) cells that are intersected by the SP satisfy the following condition for the index i_w:

$$i_w^* = \begin{cases} \max\left(\{i_w | i_u, i_v\}\right) & \text{if } \mu_w > 0 \\ \min\left(\{i_w | i_u, i_v\}\right) & \text{if } \mu_w < 0 \end{cases}$$

$$\begin{aligned}
\mu_w > 0 :\ & i_w^* < i_w < I_w \\
\mu_w < 0 :\ & 1 \leq i_w < i_w^*.
\end{aligned} \tag{3.28}$$

In Eq. 3.28 $\{i_w | i_u, i_v\}$ (read i_w given i_u and i_v) is the subset of all cell indices assembled in the SC list featuring a particular value for indices i_u and i_v. Going back to Fig. 3.3 these cells are

Algorithm 1 SC Tracking

Define : $s_\mu = \text{sign } (\mu_n)$ and $s_\eta = \text{sign } (\eta_n)$ and $s_\xi = \text{sign } (\xi_n)$
Set $i = 1 + \frac{1-s_\mu}{2}(I-1)$ and $j = 1 + \frac{1-s_\eta}{2}(J-1)$ and $k = 1 + \frac{1-s_\xi}{2}(K-1)$ and
$\vec{r}_{out} = \left(\frac{1+s_\mu}{2}x_0 + \frac{1-s_\mu}{2}X, \frac{1+s_\eta}{2}y_0 + \frac{1-s_\eta}{2}Y, \frac{1+s_\xi}{2}z_0 + \frac{1-s_\xi}{2}Z\right)$ and
$\vec{r}_s = \vec{r}_{out}$
while $1 \le i \le I$ and $1 \le j \le J$ and $1 \le k \le K$ **do**
  $\vec{r}_{in} \leftarrow \vec{r}_{out}$
  $\vec{c} = \left(\frac{1+s_\mu}{2}x_i + \frac{1-s_\mu}{2}x_{i-1}, \frac{1+s_\eta}{2}y_j + \frac{1-s_\eta}{2}y_{j-1}, \frac{1+s_\xi}{2}z_k + \frac{1-s_\xi}{2}z_{k-1}\right)$
  $\kappa = \|\vec{r}_s - \vec{c}\|$ and $\kappa_x = \frac{c_x - x_s}{\mu_n}$ and $\kappa_y = \frac{c_y - y_s}{\eta_n}$ and $\kappa_z = \frac{c_z - z_s}{\xi_n}$
  if $|\kappa_x - \kappa_y| < \kappa\epsilon_{mach}$ and $|\kappa_x - \kappa_z| < \kappa\epsilon_{mach}$ and $|\kappa_y - \kappa_z| < \kappa\epsilon_{mach}$ **then**
    Intersection with corner.
    $i \leftarrow i + s_\mu$, $j \leftarrow j + s_\eta$, $k \leftarrow k + s_\xi$, $\vec{r}_{out} \leftarrow \vec{c}$
  else
    $x_a = x_s + \kappa_x\mu_n$ and $y_a = y_s + \kappa_y\eta_n$ and $z_a = z_s + \kappa_z\xi_n$
    if $|\kappa_x - \kappa_y| < \kappa\epsilon_{mach}$ and $z_{k-1} < z_a < z_k$ **then**
      Intersection with edge along z-axis.
      $i \leftarrow i + s_\mu$, $j \leftarrow j + s_\eta$, $\vec{r}_{out} \leftarrow (c_x, c_y, z_a)$
    else if $|\kappa_x - \kappa_z| < \kappa\epsilon_{mach}$ and $y_{j-1} < y_a < y_j$ **then**
      Intersection with edge along y-axis.
      $i \leftarrow i + s_\mu$, $k \leftarrow k + s_\xi$, $\vec{r}_{out} \leftarrow (c_x, y_a, c_z)$
    else if $|\kappa_y - \kappa_z| < \kappa\epsilon_{mach}$ and $x_{i-1} < x_a < x_i$ **then**
      Intersection with edge along x-axis.
      $j \leftarrow j + s_\eta$, $k \leftarrow k + s_\xi$, $\vec{r}_{out} \leftarrow (x_a, c_y, c_z)$
    else
      $\vec{r}_1 = \vec{r_s} + \kappa_x\hat{\Omega}_n$, $\vec{r}_2 = \vec{r_s} + \kappa_y\hat{\Omega}_n$, $\vec{r}_3 = \vec{r_s} + \kappa_z\hat{\Omega}_n$
      if $y_{j-1} < y_1 < y_j$ and $z_{k-1} < z_1 < z_k$ **then**
        Intersection with West/East face.
        $i \leftarrow i + s_\mu$, $\vec{r}_{out} \leftarrow (c_x, y_1, z_1)$
      else if $x_{i-1} < x_2 < x_i$ and $z_{k-1} < z_2 < z_k$ **then**
        Intersection with South/North face.
        $j \leftarrow j + s_\eta$, $\vec{r}_{out} \leftarrow (x_2, c_y, z_2)$
      else if $y_{j-1} < y_3 < y_j$ and $z_{k-1} < z_3 < z_k$ **then**
        Intersection with Bottom/Top face.
        $k \leftarrow k + s_\xi$, $\vec{r}_{out} \leftarrow (x_3, y_3, c_z)$
      else
        Stop execution and report error.
      end if
    end if
  end if
  Add cell intersection to SC list along with $\vec{r}_{in}$ and $\vec{r}_{out}$.
end while

denoted by the black arrows indicating that one starts from the singular characteristic and tags all cells as intersected by the SP that are in-between the two red lines. Note, that depending on the sign of the direction cosine μ_w the projection might potentially be in the opposite direction but the same logic still applies. The quantities that we are ultimately interested in are the intersection points of the SPs with the edges of type II cells since they will be used to separate the flux segment within the tessellation procedure; these intersection points can be inferred by projecting back the $\vec{r}_{in}$ and $\vec{r}_{out}$ points (red points) obtained in algorithm 2 along the black arrows depicted in Fig. 3.4 to obtain the green points.

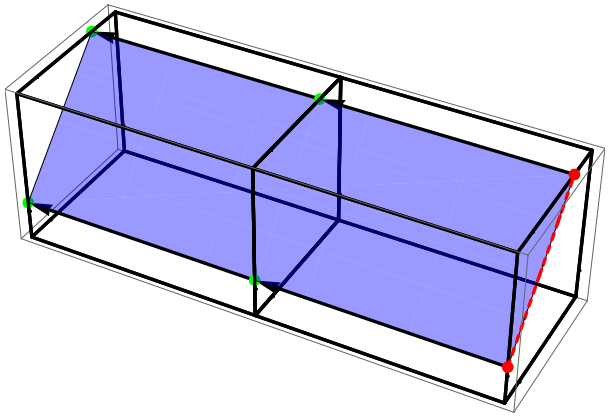


Figure 3.4: Schematic illustration of two mesh cells that are intersected by an SP. The projection of the SC onto the domain's face is depicted as dashed red line while the SP is blue. The arrows indicate the procedure by which the intersections of the SP with the cell edges are inferred from the two-dimensional tracking algorithm 2.

Tessellation of Type I and II Cells

For cells intersected either by one of the SPs or the SC it is necessary to determine the convex hull of the two or three segments, respectively, which are then tessellated using the subroutines from the Geompack90 package[52]. The difference between the type I and II intersections is that the former segments are tessellated into tetrahedrons while the latter ones are tessellated into prisms (triangular base). Since the intersections of the SPs are obtained by extruding the projection of the SC back towards the SC it is always possible to tessellate type II cells into prisms which enables a much faster integration algorithm.

Algorithm 2 Tracking of the projected SC

Define : $s_u = \text{sign } \mu_u$ and $s_v = \text{sign } \mu_v$ and $m = \frac{\mu_v}{\mu_u}$ and $d = -\frac{1-s_u}{2} mU + \frac{1-s_v}{2} V$
Set $i_u = 1 + \frac{1-s_u}{2}(I_u - 1)$ and $i_v = 1 + \frac{1-s_v}{2}(I_v - 1)$ and
$\vec{r}_{out} = \left(\frac{1+s_u}{2} u_0 + \frac{1-s_u}{2} U, \frac{1+s_v}{2} v_0 + \frac{1-s_v}{2} V\right)$ and
$\vec{r}_s = \vec{r}_{out}$
while $1 \leq i_u \leq I_u$ and $1 \leq i_v \leq I_v$ **do**
  $\vec{c} = \left(\frac{1+s_u}{2} u_{i_u} + \frac{1-s_u}{2} u_{i_u - 1}, \frac{1+s_v}{2} v_{i_v} + \frac{1-s_v}{2} v_{i_v - 1}\right)$
  $\kappa = \|\vec{r}_s - \vec{c}\|$
  if $|mc_u + d - c_v| < \kappa \epsilon_{mach}$ **then**
    Intersection with corner.
    $i_u \leftarrow i_u + s_u$, $i_v \leftarrow i_v + s_v$, $\vec{r}_{out} \leftarrow \vec{c}$
  else
    if $s_v |mc_u + d - c_v| < 0$ **then**
      Intersection with edge along v-dimension.
      $i_u \leftarrow i + s_u$, $\vec{r}_{out} \leftarrow (c_x, mc_u + d)$
    else
      Intersection with edge along y-axis.
      $i_v \leftarrow i_v + s_v$, $\vec{r}_{out} \leftarrow \left(\frac{v-d}{m}, c_y\right)$
    end if
  end if
  Add cell intersection to pSC list along with $\vec{r}_{in}$ and $\vec{r}_{out}$.
end while

Tessellating type I cells

Let the set of points given by the union of the mesh cell's corner points, the point where the SC enters and leaves the mesh cell and all viable intersections of the SPs with the cell edges be denoted by $\mathcal{P}$ (see illustration Fig. 3.5). The set of viable intersection points of the SPs with the edges is determined by intersecting the three SPs with all edges whose direction vector is not contained within the plane. Using the numbering scheme layed out in Fig. 2.1 the SPx is intersected with the E2, E4, E5, E6, E7, E8, E10 and E12 edges, while the SPy is intersected with the E1, E3, E5, E6, E7, E8, E9 and E11 and the SPz is intersected with the E1, E2, E3, E4, E9, E10, E11 and E12 edges. For determining the intersection point the parametric forms of the edges and the SP are equated:

$$\begin{aligned} \vec{r}_s + \beta\hat{\Omega}_n + \gamma\hat{e}_P &= \vec{r}_c + \alpha\hat{e}_E \\ \left[-\hat{e}_E, \hat{\Omega}_n, \hat{e}_E\right] \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} &= \vec{r}_c - \vec{r}_s, \end{aligned} \tag{3.29}$$

where $\vec{r}_s$ is defined in algorithm 1, $\vec{r}_c$ is the appropriate corner point for the respective edge, $\hat{e}_P$ is the unit vector along x,y or z defining the SP and $\hat{e}_E$ is the unit vector along the edge. Then an intersection with an edge is viable if:

$$\begin{aligned} 0 \leq \alpha \leq 1 \\ \beta \geq 0 \\ \gamma \geq 0 \end{aligned} \tag{3.30}$$

After removing duplicate points from the set $\mathcal{P}$[2], the set $\mathcal{P}$ is split into three subsets $\mathcal{P}_{[W,E]}$, $\mathcal{P}_{[S,N]}$ and $\mathcal{P}_{[B,T]}$ containing points that border the segments illuminated by the west/east, south/north and bottom/top domain boundaries, respectively, using the following rules

- The points $\vec{r}_{in}$ and $\vec{r}_{out}$ belong to all three segments.
- The viable intersection points of the SPx and the edges belong to $\mathcal{P}_{[S,N]}$ and $\mathcal{P}_{[B,T]}$ segments.
- The viable intersection points of the SPy and the edges belong to $\mathcal{P}_{[W,E]}$ and $\mathcal{P}_{[B,T]}$ segments.
- The viable intersection points of the SPz and the edges belong to $\mathcal{P}_{[W,E]}$ and $\mathcal{P}_{[N,S]}$ segments.

[2]Points are considered to be identical if their distance in some norm is smaller than a specified threshold

- Corner points belong to a single segment that can be determined by computing the orientations with respect to the SPs OR_x, OR_y and OR_z.

The sets $\mathcal{P}_{[W,E]}$, $\mathcal{P}_{[S,N]}$ and $\mathcal{P}_{[B,T]}$ are the convex hulls of their respective segment. They are stored in a list and a tessellation subroutine contained in the Geompack90 package[52] is used to obtain a tessellation comprising them.

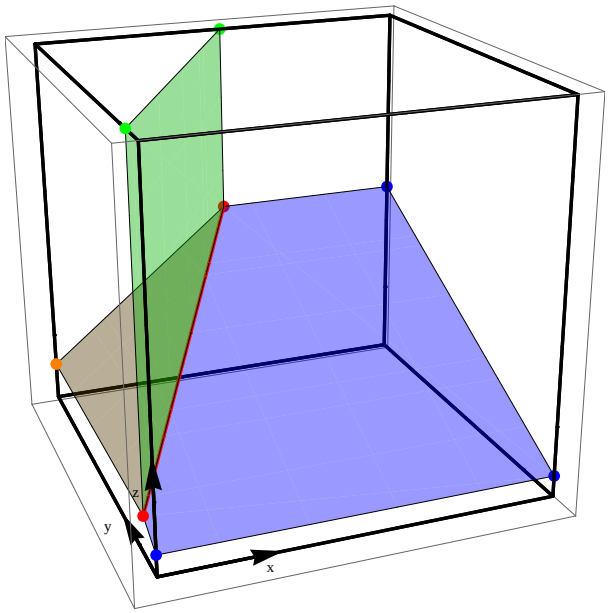


Figure 3.5: Illustration of the set of points used for the tessellation of type I intersected cells. The SC is red, the SPx is blue, the SPy is brown and the SPz is green. The set $\mathcal{P}$ contains all corners points, the points at which the SC enters and leaves the cell (red markers) and the intersections of the singular planes with appropriate edges (blue, green and orange markers).

Tessellating type II cells

Since the tessellation of the type II intersected cells is just the extrusion of a two-dimensional tessellation, i.e. a triangulation, we only discuss here how to triangulate the projection of the cell along the direction of extrusion. The set of points $\mathcal{P}_{2D}$ contains the corner points of the rectangle (projection of the cuboid onto the domain's face) and the intersection points of the projection of the SC with the edges of the rectangle. In the degenerate case that an intersection point and one of the rectangle's corners coincide, i.e. the distance is smaller than the set threshold, the corner point is removed from $\mathcal{P}_{2D}$. Then the segments are separated into the

subsets $\mathcal{P}_{2D,u}$ and $\mathcal{P}_{2D,v}$ which denote the convex hulls of the segments illuminated by the edge along the u and v variable, respectively, using the following rules:

- The intersection with the edges belong to both segments.
- For the corner points (we denote by $\vec{r}_c = (u_c, v_c)$, otherwise notation from algorithm 2):

$$\begin{aligned} \text{If} \quad & v_c - m u_c - d < 0 \text{ then } \mathcal{P}_{2D,u} \\ \text{Else} \quad & \mathcal{P}_{2D,v}. \end{aligned} \tag{3.31}$$

The two segments can then be triangulated using subroutines from the Geompack90 package and subsequently the prisms are obtained by extruding the triangles through all cells contained in the list pSC (algorithm 2) for a given pair of indices (i_u, i_v).

Integration Algorithm

From the tracking and tessellation procedures we have (1) a decomposition of all cells intersected by the SC into tetrahedra such that each tetrahedron is completely within a single segment (2) a decomposition of all cells intersected by the SPs into triangular prisms each of which is completely within a single segment; and (3) cells that are exclusively in one of the three segments. Thus, the angular flux is smooth within each tetrahedron or prism such that we can perform integrations over their respective volume. In general we are interested in the spatial Legendre moments of the angular flux within the mesh cells which can be expressed as the sum of integrals over the tetrahedral/prismatic subvolumes $V_s,\ s = 1, ..., S$:

$$\psi_{\vec{m}}^{\vec{i}} = \frac{1}{V^{\vec{i}}} \iiint\limits_{V^{\vec{i}}} dV p_{\vec{m}}\left(\vec{r}\right) \psi_n\left(\vec{r}\right) = \frac{1}{V^{\vec{i}}} \sum_{s=1}^{S} \iiint\limits_{V_s} dV p_{\vec{m}}\left(\vec{r}\right) \psi_{n,s}\left(\vec{r}\right), \tag{3.32}$$

where V_s is the volume of subvolume s and $\psi_{n,s}$ is the smooth angular flux within subvolume s, i.e. s stands for $[W, E]$, $[S, N]$ or $[B, T]$. For the sake of a more compact notation for the remainder of this chapter let the (u, v) set of spatial coordinates be augmented by w and μ_w which are associated to the standard set of coordinates and direction cosines by:

$$\begin{aligned} [W, E]: &\ w \leftarrow x,\ \mu_w = \mu_n \\ [S, N]: &\ w \leftarrow y,\ \mu_w = \eta_n \\ [B, T]: &\ w \leftarrow z,\ \mu_w = \xi_n. \end{aligned}$$

Then the integral Eq. 3.32 can be rewritten as:

$$
\begin{aligned}
\psi_{\vec{m}}^{\vec{i}} &= \frac{1}{V^{\vec{i}}} \sum_{s=1}^{S} \iiint_{V_s} dV p_{m_u}(u)\, p_{m_v}(v)\, p_{m_w}(w) \\
\times \quad & \left[\left(\sum_{l_u=0}^{L_u} \sum_{l_v=0}^{L_v} a_{l_u,l_v}^{s} \left(\bar{u} - \left| \frac{\mu_u}{\mu_w} \right| \bar{w} \right)^{l_u} \left(\bar{v} - \left| \frac{\mu_v}{\mu_w} \right| \bar{w} \right)^{l_v} - \frac{Q}{4\pi\sigma_t} \right) e^{-\frac{\sigma_t}{|\mu_w|}\bar{w}} + \frac{Q}{4\pi\sigma_t} \right] \\
= \quad & \frac{Q}{4\pi\sigma_t} \delta_{m_u,0} \delta_{m_v,0} \delta_{m_w,0} \\
+ \quad & \frac{1}{V^{\vec{i}}} \sum_{s=1}^{S} \iiint_{V_s} dV p_{m_u}(u)\, p_{m_v}(v)\, p_{m_w}(w) \\
\times \quad & \left[\left(\sum_{l_u=0}^{L_u} \sum_{l_v=0}^{L_v} a_{l_u,l_v}^{s} \left(\bar{u} - \left| \frac{\mu_u}{\mu_w} \right| \bar{w} \right)^{l_u} \left(\bar{v} - \left| \frac{\mu_v}{\mu_w} \right| \bar{w} \right)^{l_v} - \frac{Q}{4\pi\sigma_t} \right) e^{-\frac{\sigma_t}{|\mu_w|}\bar{w}} \right]
\end{aligned}
\tag{3.33}
$$

The remaining expression of the segment angular flux in the square brackets can symbolically be multiplied out which results in the following expression:

$$
\begin{aligned}
\psi_{\vec{m}}^{\vec{i}} &= \frac{Q}{4\pi\sigma_t} \delta_{m_u,0} \delta_{m_v,0} \delta_{m_w,0} \\
+ \quad & \frac{1}{V^{\vec{i}}} \sum_{s=1}^{S} \sum_{l_u=0}^{L_u} \sum_{l_v=0}^{L_v} \sum_{l_w=0}^{L_u+L_v} \left[\hat{c}_{l_u,l_v,l_w}^{s} \iiint_{V_s} dV p_{m_u}(u)\, p_{m_v}(v)\, p_{m_w}(w) \right. \\
\times \quad & \left. u^{l_u} v^{l_v} w^{l_w} \exp\left(-\frac{\sigma_t}{|\mu_w|} \frac{1 - \operatorname{sign}(\mu_w)}{2} W \right) \exp\left(-\frac{\sigma_t}{|\mu_w|} \operatorname{sign}(\mu_w)\, w \right) \right],
\end{aligned}
\tag{3.34}
$$

where the coefficient $\hat{c}_{l_u,l_v,l_w}^{s}$ can be obtained using the semi-symbolic polynomial multiplication algorithms outlined in section C.1. We now explain how to compute the volume integral in Eq. 3.34 for the three types of cells (1) not intersected by SC or SPs, (2) intersected by a single SP and (3) intersected by the SC.

Cells not intersected by SC or SPs

For cells not intersected by the SC or any of the SPs the mesh cell does not need to be divided into subvolumes such that $S = 1$ and the integral over the subvolume in Eq. 3.34 can be written

as:

$$\begin{aligned}
&\iiint_{V_s} dV p_{m_u}(u)\, p_{m_v}(v)\, p_{m_w}(w) u^{l_u} v^{l_v} w^{l_w} \exp\left(-\frac{\sigma_t}{|\mu_w|}\frac{1-\mathrm{sign}(\mu_w)}{2}W\right) \\
\times\quad & \exp\left(-\frac{\sigma_t}{|\mu_w|}\mathrm{sign}(\mu_w)\, w\right) \\
=\quad & \left[\int_{u_{i_u-1}}^{u_{i_u}} du\; p_{m_u}(u) u^{l_u}\right]\left[\int_{v_{i_v-1}}^{v_{i_v}} dv\; p_{m_v}(v) v^{l_v}\right] \\
\times\quad & \left[\exp\left(-\frac{\sigma_t}{|\mu_w|}\frac{1-\mathrm{sign}(\mu_w)}{2}W\right)\int_{w_{i_w-1}}^{w_{i_w}} dw\; p_{m_w}(w) w^{l_w} \exp\left(-\frac{\sigma_t}{|\mu_w|}\mathrm{sign}(\mu_w)\, w\right)\right]
\end{aligned} \tag{3.35}$$

For the evaluation of Eq. 3.35 integrals of the form:

$$e^b \int_{\theta_{i_\theta-1}}^{\theta_{i_\theta}} d\theta p_{m_\theta}(\theta)\, \theta^{l_\theta} e^{a\theta} \tag{3.36}$$

have to be evaluated. Note, that the case $a = b = 0$ is permitted so the algorithm that computes Eq. 3.36 needs to accommodate for this special case. The computation of the integral Eq. 3.36 is described in detail in section C.2. Thus, for cells not intersected by SC or SPs the evaluation of the spatial flux moments reduces to computing a $2(\Lambda+1)(L_u+L_v)$ one-dimensional integrals per mesh cell per discrete ordinate. For performance purposes all one-dimensional integrals for a single discrete ordinate can be pre-computed, saved for further use as required by Eqs. 3.34 and 3.35. For the C_∞ case the flux in all three segments is identical such that neither tracking nor tessellation need to be performed and only integrals of the form given in Eq. 3.36 need to be evaluated.

Cells intersected by single SPs

Cells that are intersected by a single SP feature a distinguished direction along the extrusion. It is easy to show that the direction of extrusion is never along the w-dimension so it must be along either the u or the v directions. For the further development of the integration routines let us assume that the direction of extrusion is along the u-dimension such that the

integral in Eq. 3.34 can be written as:

$$\begin{aligned}
& \iiint_{V_s} dV p_{m_u}(u)\, p_{m_v}(v)\, p_{m_w}(w) u^{l_u} v^{l_v} w^{l_w} \exp\left(-\frac{\sigma_t}{|\mu_w|}\frac{1-\mathrm{sign}(\mu_w)}{2} W\right) \\
\times \quad & \exp\left(-\frac{\sigma_t}{|\mu_w|}\mathrm{sign}(\mu_w)\, w\right) \\
= \quad & \left[\int_{u_{i_u-1}}^{u_{i_u}} du\; p_{m_u}(u) u^{l_u}\right] \left[\exp\left(-\frac{\sigma_t}{|\mu_w|}\frac{1-\mathrm{sign}(\mu_w)}{2} W\right)\right. \\
& \left.\iint_{A_s} dA p_{m_v}(v)\, p_{m_w}(w) v^{l_v} w^{l_w} \exp\left(-\frac{\sigma_t}{|\mu_w|}\mathrm{sign}(\mu_w)\, w\right)\right]
\end{aligned} \tag{3.37}$$

The one-dimensional integral in the u-dimension in Eq. 3.37 is of the form of the integral given in Eq. 3.36 such that only the double integrals over the triangle A_s needs to be discussed here which are of the form:

$$e^b \iint_{A_s} dA p_{m_\omega}(\omega)\, p_{m_\theta}(\theta) \omega^{l_\omega} \theta^{l_\theta} \exp(a\theta) \tag{3.38}$$

The utilized integration algorithm for integrals of the form Eq. 3.38 is presented in section C.3. For an efficient evaluation of the integral over the subvolume s (Eq. 3.37) the integral along the direction of extrusion is pre-computed and saved while the integrals over the triangles $s \in 1, ..., S$ are computed on the fly.

Cells intersected by SC

The integration algorithm for cells that are intersected by the SC cannot take advantage of the separability of the integral comprising Eq. 3.34. Therefore, integrals of the form:

$$e^b \iiint_{V_s} dV p_{m_\nu}(\nu)\, p_{m_\omega}(\omega)\, p_{m_\theta}(\theta) \nu^{l_\nu} \omega^{l_\omega} \theta^{l_\theta} e^{a\theta} \tag{3.39}$$

have to be evaluated. The pertinent algorithm is described in section C.4.

3.3 Error Norms

Throughout this work the spatial discretization error in the numerical solution of the S_N equations has to be quantified. The angular pointwise error $\epsilon_n(\vec{r})$ and the scalar pointwise error $\epsilon(\vec{r})$ are defined as the difference in the corresponding flux of the exact and spatially discrete

solutions of the S_N transport equations:

$$\begin{aligned} \epsilon_n(\vec{r}) &= \psi_n(\vec{r}) - \psi_n^h(\vec{r}) \\ \epsilon(\vec{r}) &= \phi_N(\vec{r}) - \phi_N^h(\vec{r}), \end{aligned} \tag{3.40}$$

The pointwise error is usually measured in some norm representing its magnitude over the whole phase space. The most natural choice of norms is to use a continuous $\mathcal{L}_p$ norm applied to the angular error[1]:

$$\|\epsilon_n\|_{c,\psi,p} = \left(\sum_{n=1}^{N} w_n \int_{\mathcal{D}} dV \, |\epsilon_n|^p \right)^{1/p}, \tag{3.41}$$

or scalar pointwise error[11]:

$$\|\epsilon\|_{c,\phi,p} = \left(\int_{\mathcal{D}} dV \, |\epsilon|^p \right)^{1/p}. \tag{3.42}$$

Often, it is more convenient to compute the error using a discrete version of the continuous error norms Eqs. 3.41 and 3.42 given by:

$$\begin{aligned} \|\epsilon_n\|_{d,\psi,p} &= \left(\sum_{n=1}^{N} w_n \sum_{\vec{i}} V_{\vec{i}} \left| \overline{\epsilon}_n^{\vec{i}} \right|^p \right)^{1/p} \\ \|\epsilon_n\|_{d,\phi,p} &= \left(\sum_{\vec{i}} V_{\vec{i}} \left| \overline{\epsilon}^{\vec{i}} \right|^p \right)^{1/p}, \end{aligned} \tag{3.43}$$

where $V_{\vec{i}}$ is the volume of the mesh cell $\Omega'_{\vec{i}}$ and $\overline{\epsilon}_n$ and $\overline{\epsilon}$ are the pointwise errors averaged over this volume:

$$\begin{aligned} \overline{\epsilon}_n^{\vec{i}} &= \frac{1}{V^{\vec{i}}} \left(1, \epsilon_n(\vec{r})\right) \\ \overline{\epsilon}^{\vec{i}} &= \frac{1}{V^{\vec{i}}} \left(1, \epsilon(\vec{r})\right). \end{aligned}$$

An important difference between the discrete and continuous error norms is that the latter allow cancellation across the mesh cells $\Omega_{\vec{i}}$ because the averaging is performed before taking the absolute values. It is then possible that the error $\epsilon_n \neq 0$ at least for some $\vec{r} \in \mathcal{D}$ is such that $\overline{\epsilon}_n^{\vec{i}} = 0$ for all $\vec{i}$, i.e. positive and negative contributions of the error cancel each other out. Therefore the discrete error norms are really semi-norms. For a certain class of problems almost exact cancellation of errors was observed in [44] and later attributed to a boundary layer that forms for configurations with large total cross sections[53].

Another difference is that the discrete norms are tied to the grid on which the spatial

averaging is performed. Within this work ψ_n and ϕ are known analytically and ψ_n^h and ϕ^h are known on the mesh characterized by mesh spacing h. Therefore, the analytical functions are simply averaged on the h mesh and the differences $\vec{\epsilon_n^i}$ and $\vec{\epsilon^i}$ are easy to compute. If the reference solution is obtained as a very fine mesh reference solution then either prolongation of the numerical solution onto the fine mesh or restriction of the reference solution onto the computational h mesh is necessary to compute $\vec{\epsilon_n^i}$ or $\vec{\epsilon^i}$.

Typical examples of the $\mathcal{L}_p$ norms are the continuous and discrete infinity norms which could be referred to as the maximum pointwise and maximum cell-wise errors, respectively. Thus, convergence as $h \to 0$ in these norms is sometimes referred to as pointwise and cellwise convergence, respectively. Another choice that will be utilized within this work is the two-norm which in contrast to the infinity norm may converge in C_0 cases where pointwise and cellwise norms do not converge. For the $\mathcal{L}_2$ it is easy to show that the discrete $\mathcal{L}_2$ norm is a truncated version of its continuous counterpart. Thereby, the truncation is performed over the summation of the modal expansion coefficients of the numerical solution within a cell. For further details and a proof confer to section A.3.

In more practical applications, the user is often interested in the accuracy of a flux or reaction rate within a certain subset of the domain $\mathcal{D}_s$, e.g. the fission rate integrated over a fuel rod:

$$\|\phi_N - \phi_N^h\|_s = \left| \int\limits_{\mathcal{D}_s} dV \phi_N - \int\limits_{\mathcal{D}_s} dV \phi_N^h \right| = \left| \int\limits_{\mathcal{D}_s} dV \epsilon\left(\vec{r}\right) \right|. \tag{3.44}$$

The difference with respect to the $\mathcal{L}_p$ error norms is that (1) before taking absolute values the exact and computed reaction rates are integrated over the subregion and the difference is computed and (2) within a mesh refinement study the subset of the domain is resolved by an increasing number of cells. In contrast, for the error norms Eq. 3.41 through 3.43 the exact and approximated fluxes are computed on a common mesh and the difference is computed for each mesh cell.

Error norms such as Eq. 3.44 are often referred to as integral error norms, e.g. [18]. They are found in [18] to converge with the theoretically predicted convergence order of two (for DD), while an error norm similar to Eq. 3.43 converges with a reduced rate because of the typical non-smoothness of the exact solution.

The choice of a particular error norm is application dependent; it should reflect features of the quantity that the user is interested in as the ultimate goal of solving the transport equation. If the user is interested in point values of the flux for example in a shielding application, a pointwise or cellwise $\mathcal{L}_\infty$ norm might be appropriate. If a region-averaged fission rate is desired, an integral norm or an $\mathcal{L}_2$ norm might be good choices.

In this work we focus on volumetric quantities when it comes to the computation of accuracy.

However, a user could demand the accuracy of the computed current across a surface instead of the accuracy of the flux. The same discussion that applies to volumes applies to face based quantities as well, i.e. their accuracy can be measured in $\mathcal{L}_p$ and/or in a region-averaged norm. In [30] Azmy shows that depending on whether fluxes (volume based) or currents (face based) are compared and their accuracy is compared, the conclusion on which is the more accurate discretization scheme, in Ref. [30] AHOTN or AHOTC, might change: AHOTN computes more accurate cell fluxes, but AHOTC provides more accurate currents.

3.4 Lathrop's Test Problem

The second test problem utilized within this work is a variation of Lathrop's test problem first published in [9] and later modified by Azmy[30]. Both references employ Lathrop's test problem for two-dimensional Cartesian geometry, while in this work the problem is extended to three spatial dimensions by replicating the characteristics of the problems along x and y axes to the z axis. The variant of Lathrop's problem used within this work is a simple cuboid-in-cuboid configuration depicted in Fig. 3.6.

The problem consists of two regions I and II, where region I contains an external distributed source and region II is source-free. Each region features a homogeneous material composition, but the materials in regions I and II may differ, and vacuum boundary conditions apply on the external faces of region II. The flux in region II is driven by the leakage out of the source region I, and thus decays exponentially towards the boundary of the domain. For the case of homogeneous materials throughout the whole domain and four distinct total cross sections, the center-line scalar flux along the x-axis is plotted in Fig. 3.7. The scattering ratio for all selected total cross sections is set to $c = 0.1$, and the solution is obtained using linear discontinuous spatial differencing, 120^3 cells ($\Delta x = \Delta y = \Delta z = 0.05$ cm), and an S_8 level symmetric quadrature. A uniform source with a cell-averaged source strength of unity is located in region I. As seen from Fig. 3.7, increasing the cross section leads to a more rapid drop of the flux right across the boundary between region I and II.

Within the framework of this work, we are interested in the resilience of spatial discretization methods to producing negative fluxes from non-negative incoming fluxes or distributed sources, and therefore the optical thickness of the spatial cells is selected to be large by setting the total cross section to large values. This means the flux is attenuated rapidly when crossing from region I to II. Negative fluxes tend to occur in optically thick, source-free regions. Thus, the described setup mimics a situation where negative fluxes are likely to occur.

Another important parameter for Lathrop's test problem is the scattering ratio in region II, because the presence of scattering reduces the likelihood of negative fluxes. Within a single source iteration, the source term is fixed such that external and scattering source are indistinguishable. Negative fluxes are a local phenomenon, i.e. they are not a deficiency of the iteration process, of the discretization in angle[3], nor are they related to the spatial mesh as a whole. They are solely attributed to the local, within-cell solution uniquely determined by the selection of the spatial discretization method. Therefore, increasing the scattering ratio is effectively equivalent to having an external source in region II that reduces the risk of negative fluxes. In section 5.2, the mechanism ensuring that sources, external fixed sources as well as

[3]Note, that in the presence of ray effects, negative fluxes can occur as a result of the hill-and-dale pattern of the flux solution. This is a non-local effect.

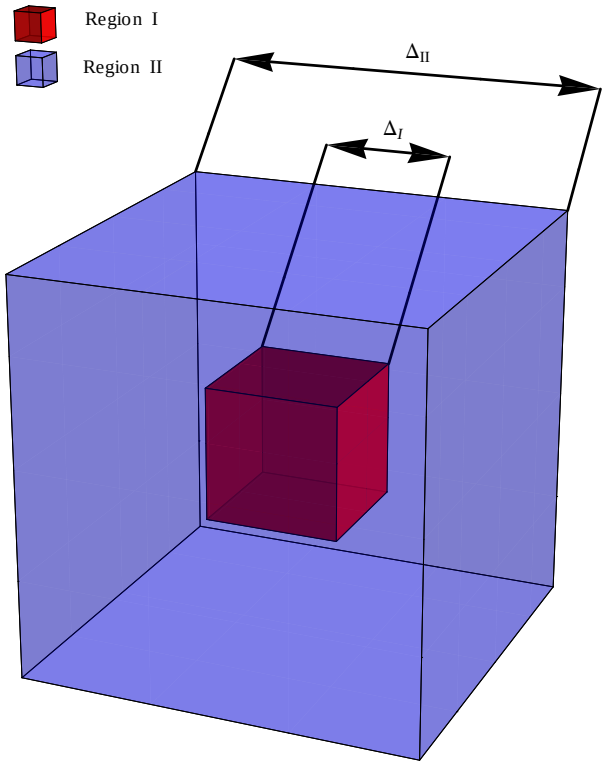


Figure 3.6: Illustration of Lathrop's test problem for three-dimensional geometry. Region I (red) contains the external source while region II (blue) does not feature an external source. Therefore, the flux in region II is driven by the leakage from region I. Vacuum boundary conditions apply on all external faces

scattering sources, always reduce the likelihood of negative fluxes will be explained in detail.

For the three-dimensional extension of Lathrop's test problem employed in this work, the material is selected to be homogeneous in regions I and II. The domain's physical size is fixed at $\Delta_I = 2$ cm and $\Delta_{II} = 6$ cm, and the domain's optical thickness is controlled by setting the total cross section to the desired value. Three values of the total cross section are utilized: 2, 4, and 16 cm^{-1}, as listed in Table 3.4, and denoted by descriptors I, II, and III, respectively. Further, three scattering ratios, also listed in Table 3.4, are employed denoted by descriptors

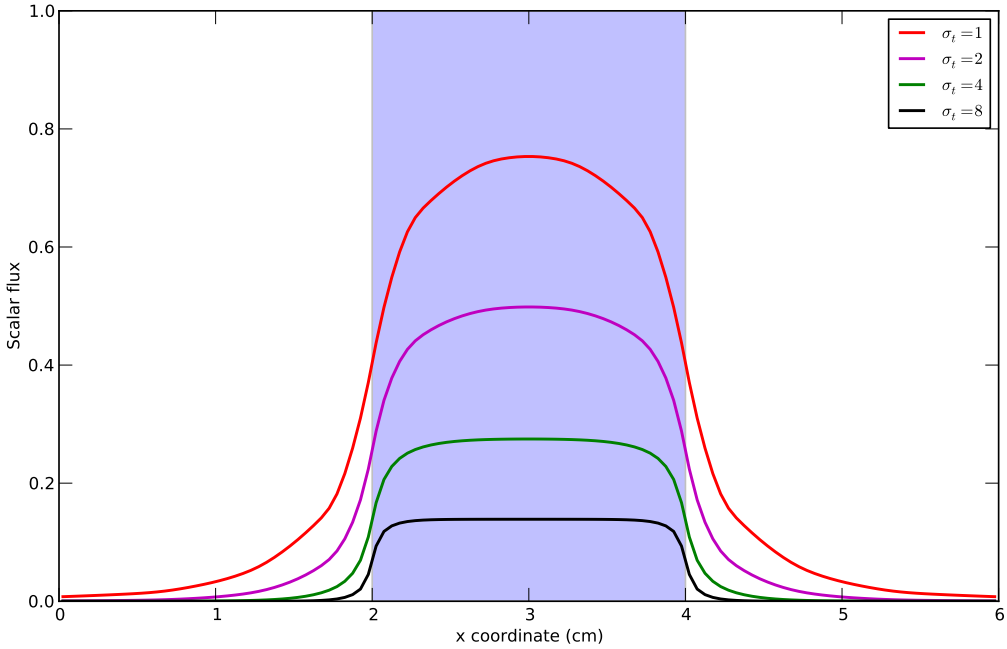


Figure 3.7: Centerline fluxes for Lathrop's test problems for set of increasing σ_t. Region I containing the external distributed source is shaded in blue.

Table 3.4: Parameter variations for Lathrop's test problem employed within this work.

Optical Thickness		Scattering Ratio	
Descriptor	σ_t	Descriptor	$c = \frac{\sigma_s}{\sigma_t}$
I	2	1	0.1
II	8	2	0.5
III	16	3	0.9

1, 2, and 3.

A particular instance of Lathrop's test problem is uniquely determined by selecting an optical thickness and a scattering ratio, e.g. L-I-1 would feature $\sigma_t = 2$ and $c = 0.1$. The external source in region I is set to a constant, cell-averaged value of unity for all test cases.

Solutions to Lathrop's test problem are obtained on six uniform meshes with $(3 \cdot 2^{k-1})^3$, $k = 1, 2, ..., 6$ mesh cells and an S_4 quadrature. A reference solution for Lathrop's test cases is not required since the information desired within this work only concerns the resilience of the target method to producing a negative flux solution from a non-negative distributed source and incoming fluxes.

3.5 Negative Flux Metrics

In order to assess the resilience of spatial discretization methods against negative fluxes, the extent of the negativity in the solution needs to be measured. This can be seen as the equivalent of measuring errors by applying an appropriate and relevant error norm to the difference of numerical and exact solution. Therefore, the same general comments apply here: the selected metric is driven by the application, i.e. the metric should measure something that is relevant to the user's purpose from the computation.

First, it needs to be defined which quantity's resilience against negative fluxes will be measured with the two obvious choices being the angular flux ψ_n and the scalar flux ϕ. Further, negative fluxes can be understood in two fundamentally different ways, namely in a pointwise and an average sense. The former implies that the solution (i.e. the flux shape) is negative if it is less than zero at any point within the domain, while the latter only asserts a negative flux if any cell- or face-averages are negative. Negative averages necessarily require the solution to be negative at some set of points such that pointwise positivity implies positivity in an average sense but not vice versa. Therefore, pointwise positivity is the stronger criterion. If positivity in an average sense is the definition of choice then we also need to distinguish between measuring negative volume averaged quantities and face averaged quantities.

Given that it is very cumbersome to check the solution for positivity in a pointwise sense[4], this work restricts its attention to checking average quantities. A total of four quantities exist to which negative flux metrics can be applied: Negative volume/face averaged angular/scalar fluxes. As negative cell averaged angular fluxes require at least one outflow average to be negative as well, it is a logical choice to focus on face-based quantities for the angular fluxes. However, scalar fluxes are typically used to compute reaction rates, which may be passed to modules that simulate other physics. Reaction rates are in essence volume based quantities and

[4]It requires to reconstruct the flux shape within each computational cell, compute the minimum of this flux shape and then check if it falls below zero.

therefore, within this work only volume-averaged scalar fluxes are scrutinized for the occurrence of negative fluxes.

With the choice being made that face-averaged angular fluxes and volume-averaged scalar fluxes shall serve as quantities of interest within Lathrop's exercise, the remaining discussion in this section shall be focused on developing metrics that assign a single number representing the "magnitude of negativity" to a given flux solution. The most straight forward metric is to count the fraction of cells featuring negative cell-averaged scalar fluxes or at least one face-averaged angular flux. The former variant is applied by Azmy in his work on nodal and characteristic methods[30]. This metric can be easily extended to angular face fluxes with the additional constraint that a cell can only be counted towards those comprising negative fluxes if all inflow face-averaged fluxes are non-negative. Thus, the two discussed metrics denoted by τ_ϕ and τ_ψ can be written as follows:

$$\begin{aligned} \tau_\phi &= \frac{\sum\limits_{\vec{i}} V^{\vec{i}} H\left(-\bar{\phi}^{h,\vec{i}}\right)}{\sum\limits_{\vec{i}} V^{\vec{i}}} \\ \tau_\psi &= \frac{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \left\{ \left[\prod\limits_{F\in\mathcal{E}^I} H\left(\bar{\psi}_{n,F}^{h,\vec{i}}\right) \right] \left[\sum\limits_{F\in\mathcal{E}^O} A_F^{\vec{i}} H\left(-\bar{\psi}_{n,F}^{h,\vec{i}}\right) \right] \right\}}{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \sum\limits_{F\in\mathcal{E}^O} A_F^{\vec{i}}}, \end{aligned} \tag{3.45}$$

where $H(x)$ is the Heaviside step function. Note, that deviating from Azmy's definition in [30] a volume/area based weighting was introduced to penalize negative averaged fluxes occurring in/on large cells volumes/face areas.

The metrics defined by Eq. 3.45 will not accurately reflect the extent of negativity if most of the negative fluxes occur in regions where the flux is negligibly small, i.e. effectively zero for all practical purposes, and the user is either not interested in these regions or considers all fluxes below a certain threshold to be effectively zero anyway. In this case it is reasonable to assume that the occurring negative fluxes are also small in magnitude such that an improved/adapted metric could require a negative flux's magnitude to be greater than a threshold $t > 0$ to be

counted:

$$\begin{aligned}
\tau_{\phi}^{t} &= \frac{\sum\limits_{\vec{i}} V^{\vec{i}} H\left(-\bar{\phi}^{h,\vec{i}} - t\right)}{\sum\limits_{\vec{i}} V^{\vec{i}}} \\
\tau_{\psi}^{t} &= \frac{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \left\{ \left[\prod\limits_{F \in \mathcal{E}^I} H\left(\bar{\psi}_{n,F}^{h,\vec{i}}\right) \right] \left[\sum\limits_{F \in \mathcal{E}^O} A_F^{\vec{i}} H\left(-\bar{\psi}_{n,F}^{h,\vec{i}} - t\right) \right] \right\}}{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \sum\limits_{F \in \mathcal{E}^O} A_F^{\vec{i}}}.
\end{aligned} \tag{3.46}$$

However, simple thresholding leaves the question open how to determine the threshold t. Clearly, t is an application driven parameter that has to be set particularly by the user. For the purpose of this work we want to derive general performance data for the selected discretization methods and therefore metric Eq. 3.46 is unpractical since it would necessitate recomputing results for various values of t. Therefore, a more viable approach is to not simply cut off negative fluxes below a certain threshold but to weight them with their magnitude such that negative fluxes with higher magnitude contribute more to the negativity metric than those with small magnitude. The metric used within this work is given by:

$$\begin{aligned}
\tau_{\phi}^{w} &= \frac{\sum\limits_{\vec{i}} V^{\vec{i}} \left|\bar{\phi}^{h,\vec{i}}\right| H\left(-\bar{\phi}^{h,\vec{i}}\right)}{\sum\limits_{\vec{i}} V^{\vec{i}} \bar{\phi}^{\vec{i}}} \\
\tau_{\psi}^{w} &= \frac{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \left\{ \left[\prod\limits_{F \in \mathcal{E}^I} H\left(\bar{\psi}_{n,F}^{h,\vec{i}}\right) \right] \left[\sum\limits_{F \in \mathcal{E}^O} A_F^{\vec{i}} \left|\bar{\psi}_{n,F}^{h,\vec{i}}\right| H\left(-\bar{\psi}_{n,F}^{h,\vec{i}}\right) \right] \right\}}{\sum\limits_{n=1}^{N} \sum\limits_{\vec{i}} \sum\limits_{F \in \mathcal{E}^O} A_F^{\vec{i}} \bar{\psi}_{n,F}^{\vec{i}}}.
\end{aligned} \tag{3.47}$$

In Eq. 3.47 the cell/face-averaged fluxes in the denominator are chosen to be computed from the exact flux solution, because in cases where negative fluxes tend to occur the numerically computed fluxes may not be reliable so that when comparing across various methods, an unfair bias may be incurred. The denominator of Eq. 3.47 serves as a normalization that makes the negativity metrics τ_{ϕ}^{w} and τ_{ψ}^{w} independent of the magnitude of the exact flux solution. In practice, the reference solution in Eq. 3.47 is replaced by a fine-mesh solution computed with a cell thickness much less than one mean free path that is verifiably positive everywhere.

3.6 Thick Diffusion Limit Test Problem

The third test problem employed within this work is used for examining if the selected spatial discretization methods possess the diffusion limit. This is important predominantly for radiative transfer problems. In subsection 3.6.1 the diffusion limit of the continuous transport equation is reviewed, and in subsection 3.6.2 a test problem is described that can be used for identifying spatial discretization methods that do not have a diffusion limit.

3.6.1 Review of the Thick Diffusive Limit (continuous S_N equations)

This review of the diffusion limit of the continuous (exact) one-speed transport equation follows [54] and [7]; an additional excellent source is Larsen's review article [55]. Consider the one-group transport equation, Eq. 1.4, in the limit as $\sigma_t = \mathcal{O}\left(\epsilon^{-1}\right)$ and $\sigma_a = \sigma_t - \sigma_s = \mathcal{O}\left(\epsilon\right)$, where ϵ is a small parameter typically identified as the ratio of a particle's mean free path and the physical extent of the domain[55]. In the limit of $\epsilon \to 0$, particles have a very small mean free path, but since σ_a goes to zero, particles also survive, on average, many collisions. Therefore, the physics of the transport process moves away from *streaming* particles' movement along their characteristic trajectory without collision, towards a Brownian type motion, characterized by a zigzag motion pattern. In this limit, the solution of the transport equation can be shown to satisfy a diffusion equation to leading order in ϵ, with boundary conditions that shall be specified later.

To sharpen the statements of the previous paragraph, a rigorous analysis can be conducted starting from a scaled version of the transport equation given by:

$$\begin{aligned} \hat{\Omega}\cdot\nabla\psi + \frac{\sigma_t(\vec{r})}{\epsilon}\psi(\vec{r},\hat{\Omega}) &= \frac{1}{4\pi}\left(\frac{\sigma_t(\vec{r})}{\epsilon} - \epsilon\sigma_a(\vec{r})\right)\phi(\vec{r}) + \epsilon\frac{Q(\vec{r})}{4\pi} \text{ for } \vec{r}\in\mathcal{D} \\ \psi(\vec{r},\hat{\Omega}) &= \psi_B(\vec{r},\hat{\Omega}) \text{ if } \vec{r}\in\partial\mathcal{D} \text{ and } \hat{n}\cdot\hat{\Omega} < 0. \end{aligned} \tag{3.48}$$

The flux solution is then postulated to be a power series in ϵ:

$$\psi(\vec{r},\hat{\Omega}) = \sum_{p=0}^{\infty}\epsilon^p\psi^{[p]}(\vec{r},\hat{\Omega}). \tag{3.49}$$

Substituting Eq. 3.49 into Eq. 3.48, collecting equal powers of ϵ and some additional manipulations lead to the following expressions:

$$\begin{aligned} \psi^{[0]} &= \frac{1}{4\pi}\phi^{[0]} \\ -\nabla\frac{1}{3\sigma_t}\nabla\phi^{[0]} + \sigma_a\left(\vec{r}\right)\phi^{[0]} &= Q(\vec{r}). \end{aligned} \tag{3.50}$$

The resulting Eq. 3.50 supports the following conclusions: First, the leading order angular flux is isotropic and equal to the leading order scalar flux up to a constant and second, the leading order scalar flux satisfies a diffusion equation. These results are the formal findings that are the commonly expressed in the (simplified) statement that the transport equation limits to the diffusion equation within diffusive regimes ($\epsilon << 1$). Additional analysis ([7], [54]) shows that the boundary conditions of the diffusion limit problem are given by the following weighted, half-range angular integral evaluated at the boundary:

$$\begin{aligned} \phi^{[0]}(\vec{r}) &= 2 \int\limits_{\hat{n}^T\hat{\Omega}<0} d\hat{\Omega} W\left(\left|\hat{n}^T\hat{\Omega}\right|\right) \psi_B(\vec{r},\hat{\Omega}) \text{ if } \vec{r} \in \partial\mathcal{D} \\ W(\mu) &= \frac{\sqrt{3}}{2}\mu H(\mu), \end{aligned} \tag{3.51}$$

where $H(x)$ is Chandrasekhars H-function[4].

The importance of these findings for numerical solution of particle transport problems are significant and can be summarized as follows: A typical length scale across which the solution of the transport equation changes is one mean free path mfp $= \frac{1}{\sigma_t} = \mathcal{O}(\epsilon)$. Therefore, as the total cross section increases, the solution changes significantly over length scales that tend to zero. However, a typical length scale of the diffusion equation is the diffusion length $L = \sqrt{3\sigma_t\sigma_a} = \mathcal{O}(1)$ which remains constant as $\epsilon \to 0$. Numerical methods require grid sizes that are at most of the order of the typical length scale of the solution to deliver reasonably accurate solutions. Therefore, as $\sigma_t \to \infty$, the mesh would typically need to be refined, which is too restrictive for certain classes of problems even for today's leadership class machines. However, as these problems tend to be diffusive as well, a grain of hope exists that we can utilize coarse meshes with $h \approx L$ and still get reasonable results from the discretization method.

Seminal work by Larsen, Morel, and Miller ([56], [57]) and later Adams[7] showed that spatial discretization methods can produce accurate results for diffusive problems for meshes characterized by $L \approx h >>$ mfp if the methods limit to a discretization of the Diffusion equation when a scaling similar to that in Eq. 3.48 is introduced into the discretized set of equations. Criteria encompassed in [7] will be described in detail in section 5.3, and later utilized to perform analysis on several linear and constant methods from the selection of spatial discretization methods for which this analysis has so far not been performed.

3.6.2 Thick Diffusion Test Case

The thick diffusion test case is adapted from [58] for three-dimensional geometries with Cartesian meshes. The domain is a uniform cube featuring an edge length of 1 cm. The diffusivity of the problem is controlled via the parameter ϵ by setting $\sigma_t = \frac{1}{\epsilon}$, $\sigma_a = \epsilon$, and $Q = \epsilon$ and thus

the scattering cross section is $\sigma_s = \frac{1}{\epsilon} - \epsilon$ and the scattering ratio is $c = 1 - \epsilon^2$. The boundary conditions are vacuum, $\psi(\vec{r}, \hat{\Omega}) = 0$, on all inflow boundary faces. A set of decreasing small parameters, $\epsilon = 0.1^l$ with $l = 0, ..., 5$, is utilized to monitor the behavior of the participating discretization methods in the thick diffusion limit.

A limiting diffusion problem (for $\epsilon \to 0$) for this case is given by the following equation:

$$\begin{aligned} -\frac{1}{3}\nabla^2\phi + \phi(\vec{r}) &= 3 \text{ if } \vec{r} \in \mathcal{D} \\ \phi(\vec{r}) &= 0 \text{ if } \vec{r} \in \partial\mathcal{D} \end{aligned} \tag{3.52}$$

The solution of the transport problem for a fixed grid should approach this solution (from above) for decreasing values of ϵ. The solution to Eq. 3.52 can be obtained analytically, but we opted to utilize an existing, verified, diffusion solver available from previous course work to compute the reference.

The investigation into discretization methods' diffusion limit is incomplete as this test problem neglects the influence of non-homogeneous boundary conditions. The described test features vacuum boundary conditions such that methods that do have the diffusion limit in the interior, but limit to inaccurate boundary conditions might perform well in this test but would fail any other test that features other types of boundary conditions.

A more comprehensive discussion of the quality of the methods' boundary conditions in the thick diffusion limit and their robustness in general is beyond the scope of this work. Therefore, a simple test problem circumventing inhomogeneous boundary conditions and their behavior in the diffusion limit is selected within this work.

Chapter 4

Implementation of Spatial Discretization Methods

4.1 Implementation of the DGFEM Methods

In this section the equations of the DGFEM methods derived using the *Lagrange* and *complete* methods are discussed putting the emphasis on the implementation of the within-cell solution process, i.e. assembly of the local system of equations, its solution and finally propagation of outflow values to the downstream cells. In subsections 4.1.1, 4.1.2 and 4.1.3 the implementation of the *Lagrange* DGFEM, *complete* DGFEM and *linear discontinuous* DGFEM is discussed, respectively.

4.1.1 Lagrange Function Space

The Lagrange function space of order Λ is the collection of polynomials satisfying

$$f_{\vec{m}} = x^{m_x} y^{m_y} z^{m_z} \text{ for } m_x, m_y, m_z = 0, ..., \Lambda. \tag{4.1}$$

Using the monomial representation of the Lagrange function space is uncommon, because it creates poorly conditioned local matrices for large expansion orders Λ. Typically, two sets of basis functions are most commonly used: (1) The Legendre polynomials (confer to section A.1) and (2) the *Lagrange* interpolatory polynomials (confer to section A.2). Note, that no matter what basis is used, Legendre polynomials or *Lagrange* interpolation polynomials, as long as the span of the utilized function space is identical as the one given by Eq. 4.1 the method is a *Lagrange* DGFEM. Within this work we exclusively utilize Legendre Polynomials Eq. 2.32.

The following discussion is based on the general formulation of the DGFEM method Eq. 2.27 with the given definitions of the mass, stiffness and edge matrices. For fully defining the

DGFEM method the explicit form of these matrices needs to be evaluated by substituting the *Lagrange* function space Eq. 2.32 into their respective definitions. For the purpose of deriving the equations for all discrete ordinates it is convenient to introduce differently scaled variables within each mesh cell centered at s_{mid}:

$$\begin{aligned} \hat{s} &= 2\ \mathrm{sg}\left(\mu_s\right)\frac{s - s_{mid}}{\Delta s} \\ \hat{p}_m\left(s\right) &= P_m\left(\hat{s}\right). \end{aligned} \tag{4.2}$$

The advantage of the scaled variable $\hat{s}$ in comparison to the scaled variable used in section A.1 is that is runs from -1 at the inflow face to 1 at the outflow face regardless of the actual signs of $\hat{\Omega}_n$ components.

Mass Matrix

First, a mapping of the three indices $\vec{m}$ into a single indexed vector $\vec{f}$ has to be defined. In this work we select to use the following non-unique mapping:

$$\left(\vec{f}\right)_{m_z+1+(\Lambda+1)m_y+(\Lambda+1)^2 m_x} = \hat{p}_{\vec{m}}\left(\vec{r}\right). \tag{4.3}$$

Using the definitions of the mass matrix given by:

$$\mathbf{M} = \left(\vec{f}, \vec{f}^T\right), \tag{4.4}$$

the elements of the mass matrix can be computed as:

$$\begin{aligned} \left(\mathbf{M}\right)_{r,c} = \left(\hat{p}_{\vec{k}}, \hat{p}_{\vec{m}}\right) &= \left[\int_{x_{i-1}}^{x_i} dx\ \hat{p}_{k_x}(x)\hat{p}_{m_x}(x)\right]\left[\int_{y_{j-1}}^{y_j} dy\ \hat{p}_{k_y}(y)\hat{p}_{m_y}(y)\right] \\ &\times \left[\int_{z_{k-1}}^{z_k} dz\ \hat{p}_{k_z}(z)\hat{p}_{m_z}(z)\right], \end{aligned} \tag{4.5}$$

where $r = k_z + 1 + (\Lambda+1)k_y + (\Lambda+1)^2 k_x$ and $c = m_z + 1 + (\Lambda+1)m_y + (\Lambda+1)^2 m_x$. Using the orthogonality property of the Legendre polynomials Eq. 4.5 can be evaluated to be a diagonal matrix given by:

$$\left(\mathbf{M}\right)_{r,c} = \frac{\Delta x_i}{2m_x+1}\frac{\Delta y_j}{2m_y+1}\frac{\Delta z_k}{2m_z+1}\delta_{m_x,k_x}\delta_{m_y,k_y}\delta_{m_z,k_z}. \tag{4.6}$$

Noting that only the mesh spacing in Eq. 4.6 may change from cell to cell, the mass matrix can be split into a varying multiplier and an invariant reduced mass matrix $\hat{\mathbf{M}}$:

$$\begin{aligned} \mathbf{M} &= V^{\vec{i}}\hat{\mathbf{M}} \\ \left(\hat{\mathbf{M}}\right)_{r,c} &= \frac{\delta_{m_x,k_x}}{2m_x+1}\frac{\delta_{m_y,k_y}}{2m_y+1}\frac{\delta_{m_z,k_z}}{2m_z+1}. \end{aligned} \tag{4.7}$$

The reduced mass matrix can be precomputed and stored for a given expansion order Λ.

Stiffness Matrices

The stiffness matrices accommodating the derivatives along the x, y and z direction are defined as

$$\begin{aligned} \mathbf{D}_x &= \left(\nabla_x \vec{f}, \vec{f}^T\right) \\ \mathbf{D}_y &= \left(\nabla_y \vec{f}, \vec{f}^T\right) \\ \mathbf{D}_z &= \left(\nabla_z \vec{f}, \vec{f}^T\right), \end{aligned} \tag{4.8}$$

respectively. Substitution of the *Lagrange* function space leads to the following expressions:

$$\begin{aligned} (\mathbf{D}_x)_{r,c} &= \left[\int_{x_{i-1}}^{x_i} dx\ \frac{d\hat{p}_{k_x}}{dx}\hat{p}_{m_x}(x)\right]\left[\int_{y_{j-1}}^{y_j} dy\ \hat{p}_{k_y}(y)\hat{p}_{m_y}(y)\right]\left[\int_{z_{k-1}}^{z_k} dz\ \hat{p}_{k_z}(z)\hat{p}_{m_z}(z)\right] \\ &= \frac{\Delta y_j}{2m_y+1}\frac{\Delta z_k}{2m_z+1}\delta_{m_y,k_y}\delta_{m_z,k_z}\left[\int_{x_{i-1}}^{x_i} dx\ \frac{d\hat{p}_{k_x}}{dx}\hat{p}_{m_x}(x)\right] \\ (\mathbf{D}_y)_{r,c} &= \left[\int_{x_{i-1}}^{x_i} dx\ \hat{p}_{k_x}\hat{p}_{m_x}(x)\right]\left[\int_{y_{j-1}}^{y_j} dy\ \frac{d\hat{p}_{k_y}}{dy}\hat{p}_{m_y}(y)\right]\left[\int_{z_{k-1}}^{z_k} dz\ \hat{p}_{k_z}(z)\hat{p}_{m_z}(z)\right] \\ &= \frac{\Delta x_i}{2m_x+1}\frac{\Delta z_k}{2m_z+1}\delta_{m_x,k_x}\delta_{m_z,k_z}\left[\int_{y_{j-1}}^{y_j} dy\ \frac{d\hat{p}_{k_y}}{dy}\hat{p}_{m_y}(y)\right] \\ (\mathbf{D}_z)_{r,c} &= \left[\int_{x_{i-1}}^{x_i} dx\ \hat{p}_{k_x}\hat{p}_{m_x}(x)\right]\left[\int_{y_{j-1}}^{y_j} dy\ \hat{p}_{k_y}\hat{p}_{m_y}(y)\right]\left[\int_{z_{k-1}}^{z_k} dz\ \frac{d\hat{p}_{k_z}}{dz}\hat{p}_{m_z}(z)\right] \\ &= \frac{\Delta x_i}{2m_x+1}\frac{\Delta y_j}{2m_y+1}\delta_{m_x,k_x}\delta_{m_y,k_y}\left[\int_{z_{k-1}}^{z_k} dz\ \frac{d\hat{p}_{k_z}}{dz}\hat{p}_{m_z}(z)\right]. \end{aligned} \tag{4.9}$$

For the evaluation of the derivative terms within the integration we first utilize the chain rule:

$$\frac{d\hat{p}_{k_x}\left(\hat{x}(x)\right)}{dx} = \frac{d\hat{p}_{k_x}}{d\hat{x}}\frac{d\hat{x}}{dx} = \mathrm{sg}\left(\mu_n\right)\frac{2}{\Delta x_i}\frac{d\hat{p}_{k_x}}{d\hat{x}}. \tag{4.10}$$

where similar expressions hold for the y and z derivatives. Now a change of variables from x to $\hat{x}$ is performed:

$$\begin{aligned}
\text{Trans. Jacobian: } d\hat{x} &= \frac{\Delta x_i}{2\text{sg}\left(\mu_n\right)} dx \\
\text{sg}\left(\mu_n\right) \frac{2}{\Delta x_i} \int_{x_{i-1}}^{x_i} dx \; \frac{d\hat{p}_{k_x}}{d\hat{x}} \hat{p}_{m_x} &\rightarrow \frac{2\text{sg}\left(\mu_n\right)}{\Delta x_i} \frac{\Delta x_i}{2\text{sg}\left(\mu_n\right)} \int_{-sg(\mu_n)}^{sg(\mu_n)} d\hat{x} \; \frac{dP_{k_x}\left(\hat{x}\right)}{d\hat{x}} P_{m_x}\left(\hat{x}\right) \\
&= \text{sg}\left(\mu_n\right) \int_{-1}^{1} d\hat{x} \; \frac{dP_{k_x}\left(\hat{x}\right)}{d\hat{x}} P_{m_x}\left(\hat{x}\right). \qquad (4.11)
\end{aligned}$$

Again, similar results can be obtained for the y and z dimensions. Further, the derivative of the Legendre polynomial can be written as a sum over lower order Legendre polynomials:

$$\frac{dP_{k_x}}{d\hat{x}} = \sum_{l=\text{mod}(k_x+1,2)}^{k_x-1} \left(2l+1\right) P_l\left(\hat{x}\right). \qquad (4.12)$$

Substituting Eq. 4.12 into the final result of Eq. 4.11 leads to:

$$\text{sg}\left(\mu_n\right) \int_{-1}^{1} d\hat{x} \; \frac{dP_{k_x}\left(\hat{x}\right)}{d\hat{x}} P_{m_x}\left(\hat{x}\right) = \text{sg}\left(\mu_n\right) \sum_{l=\text{mod}(k_x+1,2)}^{k_x-1} \left(2l+1\right) \int_{-1}^{1} d\hat{x} \; P_l\left(\hat{x}\right) P_{m_x}\left(\hat{x}\right), \qquad (4.13)$$

such that finally we can state an explicit formula for the stiffness matrix in the x direction:

$$\left(\mathbf{D}_x\right)_{r,c} = 2 \; \text{sg}\left(\mu_n\right) \frac{\Delta y_j \delta_{m_y,k_y}}{2m_y+1} \frac{\Delta z_k \delta_{m_z,k_z}}{2m_z+1} \sum_{l=\text{mod}(k_x+1,2)}^{k_x-1} \delta_{l,m_x}. \qquad (4.14)$$

Similar to the mass matrix, the stiffness matrices can be split into a varying prefactor and an invariant reduced stiffness matrix:

$$\begin{aligned}
\mathbf{D}_x &= \text{sg}\left(\mu_n\right) \Delta y_j \Delta z_k \hat{\mathbf{D}}_x \\
\left(\hat{\mathbf{D}}_x\right)_{r,c} &= 2 \frac{\delta_{m_y,k_y}}{2m_y+1} \frac{\delta_{m_z,k_z}}{2m_z+1} \left[\sum_{l=\text{mod}(k_x+1,2)}^{k_x-1} \delta_{l,m_x} \right] \\
\mathbf{D}_y &= \text{sg}\left(\eta_n\right) \Delta x_i \Delta z_k \hat{\mathbf{D}}_y \\
\left(\hat{\mathbf{D}}_y\right)_{r,c} &= 2 \frac{\delta_{m_x,k_x}}{2m_x+1} \frac{\delta_{m_z,k_z}}{2m_z+1} \left[\sum_{l=\text{mod}(k_y+1,2)}^{k_y-1} \delta_{l,m_y} \right] \\
\mathbf{D}_z &= \text{sg}\left(\xi_n\right) \Delta x_i \Delta y_j \hat{\mathbf{D}}_y \\
\left(\hat{\mathbf{D}}_z\right)_{r,c} &= 2 \frac{\delta_{m_x,k_x}}{2m_x+1} \frac{\delta_{m_y,k_y}}{2m_y+1} \left[\sum_{l=\text{mod}(k_z+1,2)}^{k_z-1} \delta_{l,m_z} \right]. \qquad (4.15)
\end{aligned}$$

Outflow Face Matrices

The outflow face matrices on the East, North and Top faces are defined as

$$\begin{aligned}
\mathbf{E}_E &= \left\langle \vec{f}\left(\hat{x}=1\right), \vec{f}^T\left(\hat{x}=1\right)\right\rangle_E \\
\mathbf{E}_N &= \left\langle \vec{f}\left(\hat{y}=1\right), \vec{f}^T\left(\hat{y}=1\right)\right\rangle_N \\
\mathbf{E}_T &= \left\langle \vec{f}\left(\hat{z}=1\right), \vec{f}^T\left(\hat{z}=1\right)\right\rangle_T .
\end{aligned} \tag{4.16}$$

Legendre polynomials $p_m(\hat{s})$ evaluated at $\hat{s}=1$ yield $p_m(\hat{s}=1)=1$ and thus the elements of the face matrices can be computed by:

$$\begin{aligned}
\left(\mathbf{E}_E\right)_{r,c} &= \left[\int_{y_{j-1}}^{y_j} dy\; p_{m_y}(y)\, p_{k_y}(y)\right]\left[\int_{z_{k-1}}^{z_k} dy\; p_{m_z}(z)\, p_{k_z}(z)\right] \\
\left(\mathbf{E}_N\right)_{r,c} &= \left[\int_{x_{i-1}}^{x_i} dy\; p_{m_x}(x)\, p_{k_x}(x)\right]\left[\int_{z_{k-1}}^{z_k} dy\; p_{m_z}(z)\, p_{k_z}(z)\right] \\
\left(\mathbf{E}_T\right)_{r,c} &= \left[\int_{x_{i-1}}^{x_i} dy\; p_{m_x}(x)\, p_{k_x}(x)\right]\left[\int_{y_{j-1}}^{y_j} dy\; p_{m_y}(y)\, p_{k_y}(y)\right],
\end{aligned} \tag{4.17}$$

where $r = k_z + 1 + (\Lambda+1)k_y + (\Lambda+1)^2 k_x$ and $c = m_z + 1 + (\Lambda+1)m_y + (\Lambda+1)^2 m_x$. The face matrices have the same dimensions as the mass and stiffness matrix, i.e. $(\Lambda+1)^3 \times (\Lambda+1)^3$. Evaluating the integrals in Eq. 4.17 leads to:

$$\begin{aligned}
\left(\mathbf{E}_E\right)_{r,c} &= \frac{\delta_{m_y,k_y}\Delta y_j}{2m_y+1}\frac{\delta_{m_z,k_z}\Delta z_k}{2m_z+1} \\
\left(\mathbf{E}_N\right)_{r,c} &= \frac{\delta_{m_x,k_x}\Delta x_i}{2m_x+1}\frac{\delta_{m_z,k_z}\Delta z_k}{2m_z+1} \\
\left(\mathbf{E}_T\right)_{r,c} &= \frac{\delta_{m_x,k_x}\Delta x_i}{2m_x+1}\frac{\delta_{m_y,k_y}\Delta y_j}{2m_y+1},
\end{aligned} \tag{4.18}$$

Finally, the outflow matrices are split into a varying prefactor and an invariant matrix $\hat{\mathbf{E}}_F$:

$$\begin{aligned}
\left(\mathbf{E}_E\right)_{r,c} &= \Delta y_j \Delta z_k \hat{\mathbf{E}}_E \\
\left(\mathbf{E}_N\right)_{r,c} &= \Delta x_i \Delta z_k \hat{\mathbf{E}}_N \\
\left(\mathbf{E}_T\right)_{r,c} &= \Delta x_i \Delta y_j \hat{\mathbf{E}}_T .
\end{aligned} \tag{4.19}$$

Inflow Face Matrices

The inflow matrices are very similar to the outflow matrix with the marked difference that the trial function expansion is taken from the respective upstream cell across the West, South or

Bottom boundary (exterior trace) and the test functions are evaluated on the inflow faces of the current cell (interior trace). Thus, the inflow face matrices can be written as:

$$\begin{aligned}
\mathbf{E}_W &= \left\langle \vec{f}\,(\hat{x}=-1)\,,\left(\vec{f}\,^{\vec{i}-s_\mu \hat{e}_x}\right)^T(\hat{x}=1)\right\rangle_W \\
\mathbf{E}_S &= \left\langle \vec{f}\,(\hat{y}=-1)\,,\left(\vec{f}\,^{\vec{i}-s_\eta \hat{e}_y}\right)^T(\hat{y}=1)\right\rangle_N \\
\mathbf{E}_B &= \left\langle \vec{f}\,(\hat{z}=-1)\,,\left(\vec{f}\,^{\vec{i}-s_\xi \hat{e}_z}\right)^T(\hat{z}=1)\right\rangle_T ,
\end{aligned} \tag{4.20}$$

where s_μ, s_η and s_ξ are the signs of μ_n, η_n and ξ_n, respectively. As the Legendre polynomials $p_m(\hat{s}=-1)$ evaluate to $(-1)^m$ the resulting expressions for the elements of the inflow face matrices cast in the form of varying prefactors and invariant matrices are:

$$\begin{aligned}
(\mathbf{E}_W)_{r,c} &= (-1)^{k_x}\frac{\delta_{m_y,k_y}\Delta y_j}{2m_y+1}\frac{\delta_{m_z,k_z}\Delta z_k}{2m_z+1} = \Delta y_j \Delta z_k \hat{\mathbf{E}}_W \\
(\mathbf{E}_S)_{r,c} &= (-1)^{k_y}\frac{\delta_{m_x,k_x}\Delta x_i}{2m_x+1}\frac{\delta_{m_z,k_z}\Delta z_k}{2m_z+1} = \Delta x_i \Delta z_k \hat{\mathbf{E}}_S \\
(\mathbf{E}_B)_{r,c} &= (-1)^{k_z}\frac{\delta_{m_x,k_x}\Delta x_i}{2m_x+1}\frac{\delta_{m_y,k_y}\Delta y_j}{2m_y+1} = \Delta x_i \Delta y_j \hat{\mathbf{E}}_B .
\end{aligned} \tag{4.21}$$

Solution Algorithm for a Single Mesh Cell

Having established explicit expressions for the entries of the mass, stiffness and face matrices, we can now proceed to describe the algorithm to assemble and solve the local linear system for each spatial mesh cell. The overall process can be divided into three stages: Assembling the local system of equations, solving the resulting linear system of equations and upstreaming the outflow face fluxes to neighboring cells. Finally, the obtained angular flux $\vec{\psi}_n^{h,\vec{i}}$ is accumulated into the scalar flux $\vec{\phi}^{h,\vec{i}}$ or angular moments thereof if anisotropic scattering is to be accounted for.

Within the first stage the local linear system is assembled. From Eq. 2.27 it follows that the local linear system can be written in the following form:

$$\mathbf{T}\vec{\psi}_n^{h,\vec{i}} = \vec{b}^{h,\vec{i}}, \tag{4.22}$$

where,

$$\begin{aligned}
\mathbf{T} &= \sigma_t V^{\vec{i}}\hat{\mathbf{M}} - |\mu_n|\,\Delta y_j \Delta z_k \hat{\mathbf{D}}_x - |\eta_n|\,\Delta x_i \Delta z_k \hat{\mathbf{D}}_y - |\xi_n|\,\Delta x_i \Delta y_j \hat{\mathbf{D}}_z \\
&\quad + |\mu_n|\,\Delta y_j \Delta z_k \hat{\mathbf{E}}_E + |\eta_n|\,\Delta x_i \Delta z_k \hat{\mathbf{E}}_N + |\xi_n|\,\Delta x_i \Delta y_j \hat{\mathbf{E}}_T \\
\vec{b} &= V^{\vec{i}}\hat{\mathbf{M}}\vec{S}^{h,\vec{i}} + \vec{\psi}_{n,W}^{h,\vec{i}} + \vec{\psi}_{n,S}^{h,\vec{i}} + \vec{\psi}_{n,B}^{h,\vec{i}},
\end{aligned} \tag{4.23}$$

and

$$\begin{aligned}\vec{\psi}^{h,\vec{i}}_{n,W} &= |\mu_n|\,\Delta y_j \Delta z_k \hat{\mathbf{E}}_W \vec{\psi}^{h,\vec{i}-s_\mu \hat{e}_x}_n \\ \vec{\psi}^{h,\vec{i}}_{n,S} &= |\eta_n|\,\Delta x_i \Delta z_k \hat{\mathbf{E}}_S \vec{\psi}^{h,\vec{i}-s_\eta \hat{e}_y}_n \\ \vec{\psi}^{h,\vec{i}}_{n,B} &= |\xi_n|\,\Delta x_i \Delta y_j \hat{\mathbf{E}}_B \vec{\psi}^{h,\vec{i}-s_\xi \hat{e}_z}_n . \end{aligned} \tag{4.24}$$

The invariant, reduced mass, stiffness and face matrices in Eq. 4.23 can conveniently be precomputed and stored before starting the solution procedure for the problem of interest. Thus, Eq. 4.23 reduces to multiplying precomputed matrices with scalar values and adding them up to obtain $\mathbf{T}$. The right hand side vector $\vec{b}$ comprises the mass matrix multiplying the (total) source vector and the upstreamed face fluxes given by Eqs. 4.24. It is unnecessary to store the angular flux vectors $\vec{\psi}^{h,\vec{i}}_n$ for all mesh cells within the mesh sweep, because the neighboring cells only require the upstreamed face flux vectors $\psi^{h,\vec{i}}_{n,W}$, $\psi^{h,\vec{i}}_{n,S}$ and $\psi^{h,\vec{i}}_{n,B}$. Therefore, Eqs. 4.24 are applied at the conclusion of the solution of cells $\vec{i}-s_\mu\hat{e}_x$, $\vec{i}-s_\eta\hat{e}_y$ and $\vec{i}-s_\xi\hat{e}_z$ which is referred to as the upstreaming phase and $\vec{\psi}^{h,\vec{i}}_{n,W}$, $\vec{\psi}^{h,\vec{i}}_{n,S}$ and $\vec{\psi}^{h,\vec{i}}_{n,B}$ are subsequently saved.

The solution of the linear system Eq. 4.22 is performed using LU decomposition[59] specifically using the *dgsev* subroutine distributed with *Lapack*[59]. It needs to be stressed here that the matrices local to cell $\vec{i}$ are of size $(\Lambda+1)^3 \times (\Lambda+1)^3$ and therefore very small when compared to the linear system that typically arises in continuous finite element methods whose number of entries grows quadratically with the number of mesh cells. For the size of such local matrices encountered in S_N algorithms direct solution methods execute faster than iterative methods.

Within the mesh sweep the scalar flux is accumulated on the fly, i.e. angular flux vectors $\vec{\psi}_{h,\vec{i}}$ are not saved for all $\vec{i}$. Note, that the trial/test functions used for deriving the finite element mass, stiffness and face matrices are dependent on the angular direction through multiplication of the sign of the appropriate direction cosines. The scalar flux is independent of the angular directions and expanded in a Legendre polynomial series using the following scaled variables:

$$\begin{aligned}\hat{s} &= 2\frac{s-s_{mid}}{\Delta s} \\ \hat{p}_m(s) &= P_m(\hat{s}) . \end{aligned} \tag{4.25}$$

Compatibility requires that the scalar flux is updated using the following prescription:

$$\left(\vec{\phi}^{h,\vec{i}}\right)_r \leftarrow \left(\vec{\phi}^{h,\vec{i}}\right)_r + w_n s_\mu^{k_x} s_\eta^{k_y} s_\xi^{k_z} \left(\vec{\psi}^{h,\vec{i}}_n\right)_r , \tag{4.26}$$

where $r = k_z + 1 + (\Lambda+1)k_y + (\Lambda+1)^2 k_x$.

Depending on the order Λ the execution time needed to perform the steps for obtaining

the mesh cell's solution varies both in time and proportion with respect to each other. The grind time, i.e. the execution time required for solving a single mesh cell for a single angular direction n, and its breakdown into the constituent operations is discussed in subsection 4.3 for all DGFEM methods utilized within this work.

4.1.2 Complete Function Spaces

The implementation of the DGFEM using the complete function space is analogous to the implementation of its Lagrange counterpart. In fact, all equations derived in the previous subsection still hold with the following modifications:

1. Each cell has a total of $\frac{1}{6}(\Lambda+3)(\Lambda+2)(\Lambda+1)$ degrees of freedom.
2. The Legendre polynomial orders k_x, k_y and k_y vary in their respective bounds such that $k_x + k_y + k_z \leq \Lambda$.
3. Row and column indices r and c of the mesh cell's mass, stiffness and face matrices are related to the k_x, k_y and k_z by:

$$r = k_z + 1 - \frac{1}{2}k_y\left(-3 + 2k_x + k_y - 2\Lambda\right) + \frac{1}{6}k_x\left(11 + k_x^2 - 3k_x(2+\Lambda) + 3\Lambda(4+\Lambda)\right).$$

4.1.3 Linear Discontinuous Method

The linear discontinuous DGFEM method (LD) is the special case of the complete DGFEM method of order $\Lambda = 1$. It is special in that the local matrix $\mathbf{T}$ is of size 4×4 and therefore its inverse can be precomputed thus saving execution time. Following [24] we decided to implement the LD method distinctly from the arbitrary order complete DGFEM kernel in order to create a highly optimized method.

For LD the local linear system specializing the general form Eq. 4.22 is given by:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ -3a_{1,2} & a_{2,2} & 0 & 0 \\ -3a_{1,3} & 0 & a_{3,3} & 0 \\ -3a_{1,4} & 0 & 0 & a_{4,4} \end{pmatrix} \begin{bmatrix} \bar{\psi}^{h,\vec{i}} \\ \psi^{h,\vec{i}}_{0,0,1} \\ \psi^{h,\vec{i}}_{0,1,0} \\ \psi^{h,\vec{i}}_{1,0,0} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \tag{4.27}$$

where:

$$
\begin{aligned}
a_{1,1} &= \Delta y_j \Delta z_k \left|\mu_n\right| + \Delta x_i \Delta z_k \left|\eta_n\right| + \Delta x_i \Delta y_j \left|\xi_n\right| + \vec{V^i}\sigma_t \\
a_{2,2} &= \Delta y_j \Delta z_k \left|\mu_n\right| + \Delta x_i \Delta z_k \left|\eta_n\right| + 3\Delta x_i \Delta y_j \left|\xi_n\right| + \vec{V^i}\sigma_t \\
a_{3,3} &= \Delta y_j \Delta z_k \left|\mu_n\right| + 3\Delta x_i \Delta z_k \left|\eta_n\right| + \Delta x_i \Delta y_j \left|\xi_n\right| + \vec{V^i}\sigma_t \\
a_{4,4} &= 3\Delta y_j \Delta z_k \left|\mu_n\right| + \Delta x_i \Delta z_k \left|\eta_n\right| + \Delta x_i \Delta y_j \left|\xi_n\right| + \vec{V^i}\sigma_t \\
a_{1,2} &= \Delta x_i \Delta y_j \left|\xi_n\right| \\
a_{1,3} &= \Delta x_i \Delta z_k \left|\eta_n\right| \\
a_{1,4} &= \Delta y_j \Delta z_k \left|\mu_n\right|, \qquad (4.28)
\end{aligned}
$$

Using Cramer's rule on the linear system Eq. 4.27 explicit expressions for the entries in $\vec{\psi}^{h,\vec{i}}$ in terms of $a_{i,j}$ and b_j can be obtained. These expressions are explicitly stated in D.1.

4.2 Transverse Moment Methods and HODD

Within this section, the implementation of the AHOTN, LL and LN methods and the HODD methods are discussed. As the final form of the HODD equations is very similar to the form of the AHOTN equations, the discussion of their implementation is grouped together within this section. In subsection 4.2.1 a direction agnostic forms of the WDD, LL and LN equations is introduced, in subsection 4.2.2 the algorithms for solving weighted Diamond Difference (WDD) and/or HODD equations are discussed and in subsection 4.2.4 a new approach to the computation of the spatial weights for the AHOTN, LL and LN methods is presented.

4.2.1 WDD Equations in Direction Agnostic Form

The balance equations Eq. 2.23 can be recast such that the outflow and inflow fluxes always appear in the same position. Recall that the East, North and Top face always denote outflow faces and the West, South and Bottom faces denote inflow faces. In the framework of solving the kernel equations, the outflow faces constitute unknowns while the inflow faces are simply

data:

$$
\begin{aligned}
& s_{\mu}^{m_x}\frac{|\mu_n|}{\Delta x_i}\left(\psi^{h,\vec{i}}_{n,E,\vec{m}^x}-(-1)^{m_x}\psi^{h,\vec{i}}_{n,W,\vec{m}^x}\right)-2s_{\mu}\frac{|\mu_n|}{\Delta x_i}\sum_{l=0}^{\left[\frac{m_x-1}{2}\right]}(2m_x-4l-1)\,\psi^{h,\vec{i}}_{n,\vec{m}-(2l+1)\hat{e}_1} \\
+ \quad & s_{\eta}^{m_y}\frac{|\eta_n|}{\Delta y_j}\left(\psi^{h,\vec{i}}_{n,N,\vec{m}^y}-(-1)^{m_y}\psi^{h,\vec{i}}_{n,S,\vec{m}^y}\right)-2s_{\eta}\frac{|\eta_n|}{\Delta y_j}\sum_{l=0}^{\left[\frac{m_y-1}{2}\right]}(2m_y-4l-1)\,\psi^{h,\vec{i}}_{n,\vec{m}-(2l+1)\hat{e}_2} \\
+ \quad & s_{\xi}^{m_z}\frac{|\xi_n|}{\Delta z_k}\left(\psi^{h,\vec{i}}_{n,T,\vec{m}^z}-(-1)^{m_z}\psi^{h,\vec{i}}_{n,B,\vec{m}^z}\right)-2s_{\xi}\frac{|\xi_n|}{\Delta z_k}\sum_{l=0}^{\left[\frac{m_z-1}{2}\right]}(2m_z-4l-1)\,\psi^{h,\vec{i}}_{n,\vec{m}-(2l+1)\hat{e}_3} \\
+ \quad & \sigma_t\psi^{h,\vec{i}}_{n,\vec{m}}=S^{h,\vec{i}}_{n,\vec{m}}. \qquad (4.29)
\end{aligned}
$$

This set of equations is shared by the AHOTN, LL, LN and HODD methods.

For closure of the AHOTN equations the WDD auxiliary equations need to be written in a direction agnostic form. The corresponding expressions are:

$$
\begin{aligned}
\frac{1+\alpha_{n,x}}{2}\psi^{h,\vec{i}}_{n,E,\vec{m}^x}+\frac{1-\alpha_{n,x}}{2}\psi^{h,\vec{i}}_{n,W,\vec{m}^x} &= \sum_{l=0,even}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(l,m_y,m_z)} \\
&+ s_{\mu}\alpha_{n,x}\sum_{l=1,odd}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(l,m_y,m_z)} \\
\frac{1+\alpha_{n,y}}{2}\psi^{h,\vec{i}}_{n,N,\vec{m}^y}+\frac{1-\alpha_{n,y}}{2}\psi^{h,\vec{i}}_{n,S,\vec{m}^y} &= \sum_{l=0,even}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(m_x,l,m_z)} \\
&+ s_{\eta}\alpha_{n,y}\sum_{l=1,odd}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(m_x,l,m_z)} \\
\frac{1+\alpha_{n,z}}{2}\psi^{h,\vec{i}}_{n,T,\vec{m}^z}+\frac{1-\alpha_{n,z}}{2}\psi^{h,\vec{i}}_{n,B,\vec{m}^z} &= \sum_{l=0,even}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(m_x,m_y,l)} \\
&+ s_{\xi}\alpha_{n,z}\sum_{l=1,odd}^{\Lambda}(2l+1)\,\psi^{h,\vec{i}}_{n,(m_x,m_y,l)} \qquad (4.30)
\end{aligned}
$$

In Eqs. 4.30 the spatial weight along direction x is computed by

$$\alpha_{n,x} = \frac{\cosh|t_{n,x}| - \sum_{m_x,odd}^{\Lambda} e_{n,m_x}}{\sinh|t_{n,x}| - \sum_{m_x,even}^{\Lambda} e_{n,m_x}}$$
$$e_{n,m_x} = \frac{2m_x+1}{\Delta x_i}\int_{x_{i-1}}^{x_i} dx e^{\sigma_t x/|\mu_n|} P_{m_x}(x), \tag{4.31}$$

and equivalent expressions hold for the y and z directions. The actual computation of the spatial weights is described in detail in subsection 4.2.4.

It is straight forward to obtain the corresponding auxiliary relations for HODD given Eqs. 4.30. HODD can be obtained as the limiting case of a general WDD method by taking

$$\begin{aligned} \Lambda \text{ is even:} \qquad & \alpha_{n,k} \to 0 \\ \Lambda \text{ is odd:} \qquad & \alpha_{n,k} \to \infty. \end{aligned} \tag{4.32}$$

This amounts to removing all terms in Eq. 4.30 that do (do not) contain α_n for even (odd) expansion orders.

Finally, the auxiliary relations for the LN and LL equations can also be written in a direction agnostic form. It is difficult to devise a short hand notation for the set of equations and therefore we opted to spell out the full set of nine equations per method here.
For the LN method:

$$\begin{aligned}
\frac{1+\alpha_{n,x,0}}{2}\psi^{h,\vec{i}}_{n,E,(0,0)} + \frac{1-\alpha_{n,x,0}}{2}\psi^{h,\vec{i}}_{n,W,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\mu\alpha_{n,x,0}\psi^{h,\vec{i}}_{n,(1,0,0)} \\
\frac{1+\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,E,(1,0)} + \frac{1-\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,W,(1,0)} &= \psi^{h,\vec{i}}_{n,(0,1,0)} \\
\frac{1+\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,E,(0,1)} + \frac{1-\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,W,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)} \\
\frac{1+\alpha_{n,y,0}}{2}\psi^{h,\vec{i}}_{n,N,(0,0)} + \frac{1-\alpha_{n,y,0}}{2}\psi^{h,\vec{i}}_{n,S,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\eta\alpha_{n,y,0}\psi^{h,\vec{i}}_{n,(0,1,0)} \\
\frac{1+\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,N,(1,0)} + \frac{1-\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,S,(1,0)} &= \psi^{h,\vec{i}}_{n,(1,0,0)} \\
\frac{1+\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,N,(0,1)} + \frac{1-\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,S,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)} \\
\frac{1+\alpha_{n,z,0}}{2}\psi^{h,\vec{i}}_{n,T,(0,0)} + \frac{1-\alpha_{n,z,0}}{2}\psi^{h,\vec{i}}_{n,B,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\xi\alpha_{n,y,0}\psi^{h,\vec{i}}_{n,(0,0,1)} \\
\frac{1+\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,T,(1,0)} + \frac{1-\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,B,(1,0)} &= \psi^{h,\vec{i}}_{n,(0,1,0)} \\
\frac{1+\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,T,(0,1)} + \frac{1-\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,B,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)},
\end{aligned} \tag{4.33}$$

For the LL method:

$$
\begin{aligned}
\frac{1+\alpha_{n,x,0}}{2}\psi^{h,\vec{i}}_{n,E,(0,0)} + \frac{1-\alpha_{n,x,0}}{2}\psi^{h,\vec{i}}_{n,W,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\mu\alpha_{n,x,0}\psi^{h,\vec{i}}_{n,(1,0,0)} \\
\frac{1+\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,E,(1,0)} + \frac{1-\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,W,(1,0)} &= \psi^{h,\vec{i}}_{n,(0,1,0)} \\
&- 3\frac{s_\mu\alpha_{n,x,1}}{s_\eta t_y}\left(\psi^{h,\vec{i}}_{n,N,(1,0)} + \psi^{h,\vec{i}}_{n,S,(1,0)} - 2\psi^{h,\vec{i}}_{n,(1,0,0)}\right) \\
\frac{1+\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,E,(0,1)} + \frac{1-\alpha_{n,x,1}}{2}\psi^{h,\vec{i}}_{n,W,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)} \\
&- 3\frac{s_\mu\alpha_{n,x,1}}{s_\xi t_z}\left(\psi^{h,\vec{i}}_{n,T,(1,0)} + \psi^{h,\vec{i}}_{n,B,(1,0)} - 2\psi^{h,\vec{i}}_{n,(1,0,0)}\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{1+\alpha_{n,y,0}}{2}\psi^{h,\vec{i}}_{n,N,(0,0)} + \frac{1-\alpha_{n,y,0}}{2}\psi^{h,\vec{i}}_{n,S,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\eta\alpha_{n,y,0}\psi^{h,\vec{i}}_{n,(0,1,0)} \\
\frac{1+\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,N,(1,0)} + \frac{1-\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,S,(1,0)} &= \psi^{h,\vec{i}}_{n,(1,0,0)} \\
&- 3\frac{s_\eta\alpha_{n,y,1}}{s_\mu t_x}\left(\psi^{h,\vec{i}}_{n,E,(1,0)} + \psi^{h,\vec{i}}_{n,W,(1,0)} - 2\psi^{h,\vec{i}}_{n,(0,1,0)}\right) \\
\frac{1+\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,N,(0,1)} + \frac{1-\alpha_{n,y,1}}{2}\psi^{h,\vec{i}}_{n,S,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)} \\
&- 3\frac{s_\eta\alpha_{n,y,1}}{s_\xi t_z}\left(\psi^{h,\vec{i}}_{n,T,(0,1)} + \psi^{h,\vec{i}}_{n,B,(0,1)} - 2\psi^{h,\vec{i}}_{n,(0,1,0)}\right) \\
\frac{1+\alpha_{n,z,0}}{2}\psi^{h,\vec{i}}_{n,T,(0,0)} + \frac{1-\alpha_{n,z,0}}{2}\psi^{h,\vec{i}}_{n,B,(0,0)} &= \psi^{h,\vec{i}}_{n,(0,0,0)} + 3s_\xi\alpha_{n,y,0}\psi^{h,\vec{i}}_{n,(0,0,1)} \\
\frac{1+\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,T,(1,0)} + \frac{1-\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,B,(1,0)} &= \psi^{h,\vec{i}}_{n,(0,1,0)} \\
&- 3\frac{s_\xi\alpha_{n,z,1}}{s_\mu t_x}\left(\psi^{h,\vec{i}}_{n,E,(0,1)} + \psi^{h,\vec{i}}_{n,W,(0,1)} - 2\psi^{h,\vec{i}}_{n,(0,0,1)}\right) \\
\frac{1+\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,T,(0,1)} + \frac{1-\alpha_{n,z,1}}{2}\psi^{h,\vec{i}}_{n,B,(0,1)} &= \psi^{h,\vec{i}}_{n,(0,0,1)} \\
&- 3\frac{s_\xi\alpha_{n,z,1}}{s_\eta t_y}\left(\psi^{h,\vec{i}}_{n,N,(0,1)} + \psi^{h,\vec{i}}_{n,S,(0,1)} - 2\psi^{h,\vec{i}}_{n,(0,0,1)}\right).
\end{aligned}
\tag{4.34}
$$

In Eqs. 4.33 and 4.34 the spatial weights are given by the following expressions:

$$
\alpha_{n,k,l} = \frac{\coth|t_{n,x}| - \frac{1}{|t_{n,x}|}}{1 - \frac{\nu_l}{|t_{n,x}|}\left[\coth|t_{n,x}| - \frac{1}{|t_{n,x}|}\right]},
\tag{4.35}
$$

where $\nu_0 = 1$ and $\nu_1 = 0$. The spatial weights Eq. 4.35 are computed via the method presented in subsection 4.2.4.

The local linear system that is assembled and solved via substitution of the outflow un-

knowns within the balance equations are derived and pre-solved as demonstrated in Appendix D.2 and D.3 for LN and LL, respectively. For LN the encountered structure and complexity is similar to the LD method given in Eq. 4.27 but LL creates a much more difficult local linear system.

4.2.2 Assembly and Solution of Local Linear System

Within the AHOTN, LL, LN and HODD systems of equations two distinct sets of equations coexists: the balance equations and the auxiliary equations. There is one balance equation for each volume moment and one auxiliary relation for each outflow face moment. Therefore, the two obvious approaches to solving the aforementioned systems of linear equations are:

1. Analytically solve the balance relations for the volume moments.
2. Eliminate the volume moments from the auxiliary equations.
3. Solve (potentially numerically) a linear system for the outflow face moments.
4. Compute the volume moments from the balance equations.

or

1. Solve the auxiliary equations for the face moments.
2. Eliminate the face moments from the balance equations.
3. Solve a linear system for the volume moments.
4. Compute the outflow face fluxes from the auxiliary equations.

The second approach is selected for implementation in this work for two main reasons: (a) solving the balance equations analytically for the volume moments is algebraically tedious[1], (b) for $\Lambda \leq 2$ the linear system solved for the first approach is not smaller than the linear system of equations arising from the second approach; thus there is no advantage to using the first approach for $\Lambda \leq 2$.

The solution process within the AHOTN, LL, LN and HODD kernels consists of the following steps:

1. Computation of the spatial weights (not necessary for HODD).
2. Assembly of the local linear system $\mathbf{T}\vec{\psi}^{h,\vec{i}} = \vec{b}$.
3. Solution of the local linear system.

[1]System of balance equations is lower triangular so a recursion can be used to solve for the volume moments.

4. Using the auxiliary relations to compute the unknown outflow face flux moments.

These steps are very similar to the process described for the DGFEM method in section 4.1 except that no matrix templates are used within the assembly stage of the local linear system which makes the process slightly less efficient. Instead the process is implemented using various loops.

The final matrix obtained for the AHOTN and HODD method can be symmetrized by multiplying each row r of the linear system of equations with a multiplier c_r given by

$$\begin{aligned} c_r &= \frac{(-1)^{\mathrm{mod}(i_x,2)}\,(-1)^{\mathrm{mod}(i_y,2)}\,(-1)^{\mathrm{mod}(i_z,2)}}{(2i_x+1)(2i_y+1)(2i_z+1)} \\ r &= i_z + 1 + (\Lambda+1)i_y + (\Lambda+1)^2 i_x. \end{aligned} \tag{4.36}$$

The resulting matrix is symmetric but not positive definite such that the *dsysv*[59] solver can be applied for its inversion. However, it is found that the *dgesv* solver is significantly faster than the *dsysv* routine for small matrices which is likely attributed to initial overhead that is not saved in the later solution process when the matrix is too small. Due to the small size of the local matrices the *dgesv* routine is used for the AHOTN and HODD methods.

In addition to the standard, general order AHOTN implementation, a hard-coded AHOTN-1 version was created that will be referred to as AHOTN-1*. The difference from the standard AHOTN implementation is that the expressions for the matrix elements, right hand side vector and the upstream relations have been pre-computed in Mathematica and are explicitly applied within the kernel such that no loops are necessary to perform the corresponding steps within the kernel operation. The corresponding expressions are contained within the Mathematica notebook listing in section D.4.

As for the LD methods the LL and LN methods result in linear systems of size 4×4. Therefore, the solution of their respective linear systems of equations can be precomputed and hard-coded which streamlines the execution of the LL and LN methods. For the structure of the resulting matrices and their precomputed solutions consult sections D.3 and D.2.

4.2.3 Stopping Criterion of Source Iterations

Consistent for all methods, the source iterations are successfully terminated if the maximum relative change of the cell-averaged scalar flux from iteration p to $p+1$ is smaller than the stopping criterion ϵ_s:

$$\max_{\vec{i}} \left| 1 - \frac{\bar{\phi}_N^{h,\vec{i},p+1}}{\bar{\phi}_N^{h,\vec{i},p}} \right| < \epsilon_s. \tag{4.37}$$

Observe that stopping the source iterations is solely based on the cell-averaged scalar fluxes. Higher-order cell moments convergence is not monitored at all. This is common practice in S_N transport codes, e.g. in DENOVO[24] and PARTISN[60].

4.2.4 Computation of the Spatial Weights

The spatial weights $\alpha_{m,n,l}$ capture the within-cell transport physics included in the AHOTN, LL and LN methods and therefore their stable and accurate computation is important. Using the recursion relation Eq. 2.49 for computing the AHOTN weights, poses the risk of contamination with numerical imprecision especially for small cell optical thicknesses and high polynomial expansion orders. In addition, the computation of the spatial weights via the recursion relation can consume a non-negligible fraction of the total solve time. Because of the particular problem with the weight's numerical stability, near-zero optical thickness expansion of the spatial weights for the AHOTN method were devised in [41] that are stated in Eq. 2.50. The expressions in Eq. 2.50 solve the problem of the numerical instability and are very cheap to evaluate thus reducing the burden of the weight computation that is part of the within-cell solve load to practically zero.

However, the near zero expansion cannot be utilized across the whole range of optical thicknesses so that a cut-off value of the optical thickness has to be used above which Eq. 2.49 is used and below which Eq. 2.50 is employed. Within this work a new approach was implemented that (1) is numerically stable, (2) is applicable to all optical thicknesses and (3) is computationally inexpensive. In Fig. 4.1 exact AHOTN weights for expansion orders 0 to 3 are plotted versus the optical thickness. The functions are smooth, monotonically increasing for even orders and monotonically decreasing for odd orders, and limit to unity for increasing optical thicknesses. Their asymptotic behavior for small optical thicknesses has already been discussed and is expressed in Eq. 2.50.

The fact that the spatial weights depend only on one parameter, i.e. the optical thickness, makes them amenable to an easy table lookup procedure. A table lookup is based on a pre-computed list of base point values and an interpolation procedure associated for computing values between two base points. The challenge for designing a lookup procedure for the spatial weights is that for odd expansion orders the weight limits to infinity as $t \to 0$ requiring a very fine "mesh" of base points close to unity to ensure accurate computation of the spatial weight in the vicinity of $t = 0$. This would entail either maintaining a very fine grid of base points everywhere or using a non-uniform spacing of the grid points. Both alternatives may lead to a loss of efficiency of the table lookup algorithm: Maintaining a very fine grid everywhere would lead to an unacceptably large number of base points to be saved, while a non-uniform grid necessitates a more complicated/expensive algorithm for finding the two bounding base points

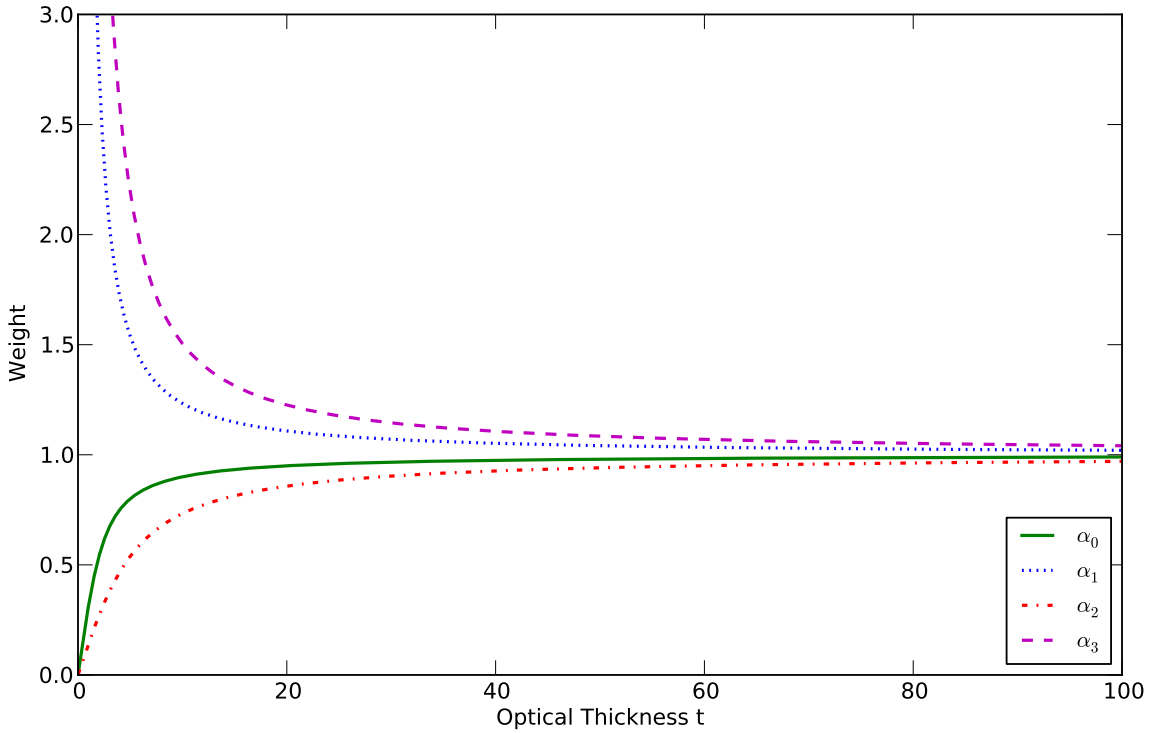


Figure 4.1: Exact spatial weights for AHOTN plotted versus the cell optical thickness for AHOTN-$0, .., 3$.

for interpolation.

The solution we found within this work to overcome this challenge is to utilize a piecewise Pade approximation[61] as interpolation prescription between base points. In particular the (1,1) Pade approximation given by:

$$\alpha^P(t) = \frac{a+bt}{1+ct}, \tag{4.38}$$

is employed. The great advantage of this interpolant is that it naturally accommodates the limiting behavior of the spatial weights for both even and odd expansion orders for $t \to 0$ and $t \to \infty$. Therefore, it is not necessary to maintain a very fine mesh even for $t \to 0$ and odd polynomial expansion orders.

The efficient algorithm for computing the spatial weights is based on using a uniform spacing of the support points on the optical thickness axis so that the l-th support point is located at position $t_l = l \cdot \Delta t$ for $l = 0, 1, ...$ Let us for the moment assume that l may run up to infinity,

i.e. the table has an infinite length. For each support point the exact Pade approximation is computed and the respective values of a_l, b_l and c_l are saved. For the sake of practicality, a Mathematica script listed in section B.3 was used to perform this task. The algorithm to compute the spatial weights can then be stated as follows:

$$\begin{aligned} l &= \text{round}\left(\frac{t}{\Delta t}\right) \\ \text{Retrieve} & \quad a_l,\ b_l,\ c_l \\ \alpha^P(t) &= \frac{a_l + b_l t}{1 + c_l t}. \end{aligned} \tag{4.39}$$

The computations involved in Eqs. 4.39 are cheap: they require a division followed by rounding a real number to the nearest integer, an array lookup, two multiplications, two additions and an additional division.

In reality, the lookup table must be truncated at some finite t_L. As the spatial weights approach unity for $t \to \infty$ the error associated with truncating the table and extrapolating if optical thicknesses $t > t_L$ occur can be bounded by $|1 - \alpha(t_L)|$ as long as:

$$\lim_{t\to\infty} \frac{a_L + b_L t}{1 + c_L t} = 1, \tag{4.40}$$

because the Pade approximation exactly reproduces the value of the spatial weight at the last support point $\alpha^P(t_L) = \alpha(t_L)$ and the extrapolated value does not leave the interval $[1, \alpha(t_L))$ for any value of $t \in [t_L, \infty)$. For the tables used within this work condition 4.40 holds for all utilized spatial expansion orders.

For the purpose of this work we found it to be sufficient to select $\Delta t = 0.01$ and $t_L = 200$. A plot of the relative difference (in %) of the exact and Pade approximated spatial weights is presented in Fig. 4.2 for orders $\Lambda = 0, 1$. The error is bounded above by 1 %.

4.3 Methods' Grind Times

The solution of the S_N equations is typically facilitated using the Source Iteration method which, from an implementation point of view, is basically a loop wrapped around the space-angle sweeps. The space-angle sweep is a (double)-loop wrapped around the execution of the kernel subroutine that solves the equations for a single mesh cell. It is expected that the lion's share of the code's execution time is spent within the Kernel subroutine. Therefore, the code's total execution time should be close to the execution time of the kernel times the number of calls to the kernel. The time it takes to execute the kernel for a single mesh cell and angular direction is defined to be the grind time of the method.

The grind time depends on the method, method order, implementation and compilation

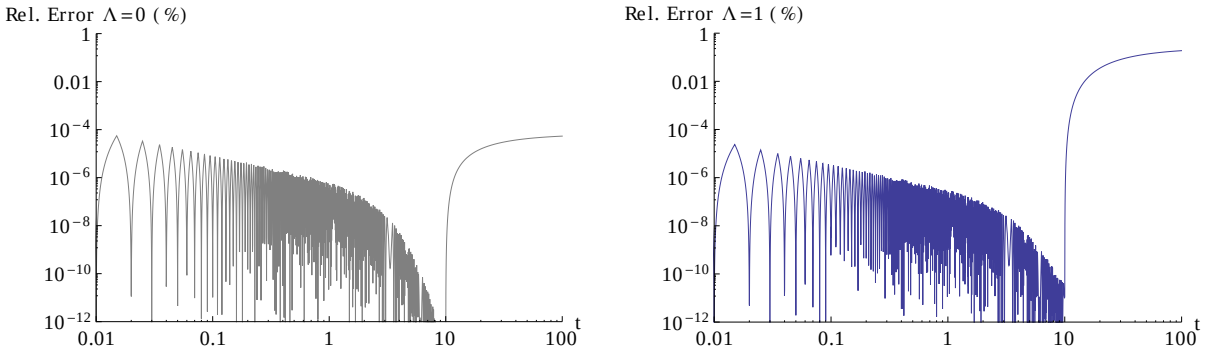


Figure 4.2: Relative error associated with Pade computed spatial weight for orders $\Lambda = 0, 1$ in (%).

details, the hardware that the code is executed on and possibly many more factors, but it does not depend on the size of the problem, i.e. the number of cells, angles, or energy groups. The number of function calls to the kernel subroutine only depends on the size of the problem and the iterative convergence properties and stopping criteria.

In Tables 4.1 through 4.4 the grind times Δt_g of all methods employed in this work are listed along with a breakdown into the constituent operations performed within the kernel execution: computation of spatial weight Δt_w, matrix/right hand side assembly Δt_b, solution of the linear system Δt_s and upstreaming Δt_u.

The grind times are measured by executing the kernel within a loop 10^6 times, placing timing commands *cpu_time* before entering and after exiting the loop and dividing the total execution time by the number of traversals through that loop. The implicit assumption is that the execution time spent for administering the loop is negligible compared to the execution of the respective kernel routines. Note, that for determining Δt_g no calls to *cpu_time* are performed within the kernel subroutines and therefore the timing method is non-intrusive.

For computing Δt_w, Δt_b, Δt_s and Δt_u calls to *cpu_time* are placed within the kernel subroutines at appropriate positions, i.e. before and after the blocks that perform the actions lumped into the four identified constituent categories. Similar to measuring the grind time the kernels are executed 10^6 times and five counters are incremented for $\Delta t_j,\ j = w, b, s, u$. In contrast to measuring the grind time where a total of only two calls was required, four to five calls to *cpu_time* within every traversal through the kernel subroutine are necessary. The execution time spent for a single call to *cpu_time* is thereby comparable to some of the measured block execution times. Measuring the block execution times is intrusive and distorts the measured execution times. Thus, the actually measured block execution times will be inaccurate but we

conjecture that their ratios are more accurate. Therefore, the $\Delta t_j,\ j = w, b, s, u$ are computed as follows:

$$\Delta t_j = \frac{\Delta t'_j}{\sum_i \Delta t'_i} \Delta t_g, \tag{4.41}$$

where $\Delta t'_j$ are the measured block execution times. Still, the $\Delta t_j,\ j = w, b, s, u$ presented in Tables 4.1 through 4.4 should be regarded with caution due to the potential inaccuracy of their measurement.

The fastest executing methods are as expected the zeroth order Diamond Difference and the Linear Discontinuous method. These methods are followed by the Linear-Linear and the Linear Nodal methods which are about five and 9 times slower, respectively. The five fastest methods are either constant or linear approximation (with reduced number of cross moments) and none of them need to call an external linear solver subroutine either because the linear system is presolved or because no linear system has to be solved.

With increasing the expansion order, the grind time increases dramatically which is mainly driven by the linear solve time t_s which makes up the fastest growing part of the grind time: the LU decomposition's execution time scales cubically with the number of degrees of freedom of the linear system of equations, i.e. Λ^6. It is therefore not surprising that among the arbitrary order methods the DGFEM with *complete* function space requires the least execution time, HODD is only marginally cheaper than AHOTN, and DGFEM with *Lagrange* function space and $\Lambda > 1$ surprisingly takes the longest execution time. The reason why DGLA-Λ with $\Lambda > 1$ features much longer execution times than AHOTN or HODD of the same order is the significantly more expensive solution of the linear system of equations. We conjecture that the structure of the DGLA matrices causes the *Lapack* routine *dgesv* to execute slower.

It should be stressed that the difference between AHOTN and HODD is not driven by the additional computation of the spatial weights required for the AHOTN method but rather by the smaller number of terms in the WDD equations. Finally, the hard-coded AHOTN-1* method is significantly faster than the standard AHOTN-1 method and the SCB methods is both faster than the DGFEM-L-1 method which it is derived from and the AHOTN-1 method. The shorter execution time of SCB compared to DGFEM-L-1 is attributed to the fact that SCB is a hard-wired $\Lambda = 1$ application while DGFEM-L-1 is a general DGFEM implementation.

4.4 A New SCT-Step Method

The exact solution of the S_N transport equations may be discontinuous across the singular planes if the inflow fluxes on the three inflow faces differ. In this case, the discontinuity will lead to a non-convergence of the cells intersected by the singular planes (see [21] and [22]). As these cells are of measure zero, i.e. the volume fraction assumed by them goes to zero as

Table 4.1: Grind time and its constituents for the AHOTN, LL and LN methods in μs.

	AHOTN-1	AHOTN-2	AHOTN-3	AHOTN-4	AHOTN-1*	LL	LN
Δt_g	5.30	21.01	71.72	233.09	2.86	0.93	0.49
Δt_w	0.69	0.94	0.79	1.37	0.49	0.22	0.13
Δt_b	1.72	8.06	23.36	71.31	0.49	0.28	0.13
Δt_s	1.95	9.75	44.56	152.61	1.44	0.24	0.12
Δt_u	0.95	2.27	3.01	7.80	0.44	0.20	0.12

Table 4.2: Grind time and its constituents for the HODD method in μs.

	HODD-1	HODD-2	HODD-3	HODD-4	DD
Δt_g	4.54	18.92	62.20	211.57	0.12
Δt_b	1.76	7.37	21.16	56.95	0.00
Δt_s	2.02	10.02	38.98	151.32	0.00
Δt_u	0.76	1.53	2.07	3.31	0.00

Table 4.3: Grind time and its constituents for the DGFEM Lagrange method of order 1 through 4, the simple corner balance method and the step characteristic method in μs.

	DGFEM-L-1	DGFEM-L-2	DGFEM-L-3	DGFEM-L-4	SCB
Δt_g	4.30	31.22	137.45	1167.50	3.04
Δt_b	0.84	3.98	17.42	57.51	0.59
Δt_s	2.25	21.58	116.36	1025.42	1.86
Δt_u	1.20	5.66	3.67	84.64	0.59

Table 4.4: Grind time and its constituents for the DGFEM Complete method of order 1 through 4 and the LD method in μs.

	DGFEM-C-1	DGFEM-C-2	DGFEM-C-3	DGFEM-C-4	LD
Δt_g	1.72	6.42	17.78	53.10	0.11
Δt_b	0.41	1.16	2.03	7.09	0.00
Δt_s	0.80	3.52	12.55	38.09	0.00
Δt_u	0.51	1.74	3.20	7.92	0.00

the mesh is refined, convergence, however slow, is still warranted in any norm $\|\cdot\|_{d,p}$ except $p = \infty$. As the infinity norm indicates the largest cell wise error in the average flux, it cannot converge to zero in a C_0 configuration, because cells intersected by the singular planes do not converge. Duo and Azmy[17] have previously stated this fact and labeled it "lack of pointwise" convergence in the presence of discontinuities. However, in order to distinguish pointwise errors from errors in the average, we shall refer to this particular lack of convergence as "cell-wise".

In very simple terms, the deficiency of standard discretization methods when it comes to C_0 configurations is the mixing of solution segments illuminated by different inflow boundaries. To illustrate the issue, consider a cell intersected by the singular characteristic. Let the inflow on the West, South, and Bottom boundary faces be $\bar{\psi}_W$, $\bar{\psi}_S$, and $\bar{\psi}_B$, respectively, and let them all be different. Also assume that the attenuation within the domain is negligible and that no external or scattering source is present. Clearly, this is not a problem of any practical relevance, but we will show that standard discretization methods will not even obtain the correct answer in case at least one of the $\bar{\psi}_m$ differs from the other two. The exact cell-averaged angular flux for this case would be:

$$\bar{\psi}_n^{\vec{i}} = \frac{\bar{\psi}_W V_W^{\vec{i}} + \bar{\psi}_S V_S^{\vec{i}} + \bar{\psi}_B V_B^{\vec{i}}}{V^{\vec{i}}}, \tag{4.42}$$

where $V_m^{\vec{i}}$ is the volume of the region illuminated by boundary face m.

Note, that this flux does not depend on the actual physical extent of the cell but on the volume fractions $V_m^{\vec{i}}/V^{\vec{i}}$, which remain constant as long the cell's aspect ratio and $\hat{\Omega}_n$ remain constant. Without computing the volume fractions $V_m^{\vec{i}}/V^{\vec{i}}$ correctly a discretization will not produce the exact answer to the posed problem. In order for a method to obtain the right answer, it needs to track the position of the singular planes and in some form (if not explicitly) compute the volumes in Eq. 4.42 correctly.

In two-dimensional geometry Duo[17] suggested tracking of the singular characteristic line through the mesh and applying a sub-cell approach in intersected cells to keep segments in these cells isolated from each other. For the solution of the subcell equations, Duo used the *Step Characteristic* method applied to each of the segments separately. For further details of the Duo's SCT algorithm, references [17] and [1] may be consulted. The results found in these two references were that the SCT algorithm (1) restored convergence in the infinity norm for C_0 type problems and (2) improved accuracy and observed rate of convergence for C_0 and C_1 test problems. Encouraged by the success of Duo's SCT algorithm we decided to implement a similar algorithm for three-dimensional Cartesian geometry.

The immediate difference from the two-dimensional problem is the increased difficulty of tracking the Singular Characteristic and Singular Planes. However, an efficient algorithm for performing all necessary tracking operations was already implemented within this work for the MMS3D code. The techniques developed for the MMS3D implementation reported in section

3.2.3, in particular algorithms 1 for tracking the singular characteristic line and 2 for tracking the singular planes can be used to obtain all necessary information to identify all cells intersected by SC and SPs and split them into sub-cells according to the boundary face they are illuminated from.

Although the tracking algorithm requires the use of quadruple precision to ensure that intersections with edges and corners are handled adequately, the algorithm is still reasonably efficient because its complexity is at most $\mathcal{O}\left((\max(I,J,K))\right)^2$, with the most expensive calculations done only during execution of algorithm 1. The basic step in algorithm 1 only has to be repeated $\mathcal{O}\left(\max(I,J,K)\right)$ times, and therefore the hope is that for a small number of cells the execution time required for the tracking algorithm will grow like $\mathcal{O}\left(\max(I,J,K)\right)$.

Timing data for the tracking algorithm for increasing $\max(I,J,K)$ are plotted in Fig. 4.3, substantiating the described complexity estimates. The complexity of an S_N flux solution is $\mathcal{O}\left(I \cdot J \cdot K\right)$ and, under the assumption that the cell aspect ratio is retained within each mesh refinement step, its complexity is $\mathcal{O}\left(\left(\max\ (I,J,K)\right)^3\right)$. Therefore, in the limit of fine meshes, the tracking computation will require negligible execution time compared with the mesh sweep execution time.

From the tracking computation described in the context of the MMS test problem, the convex hull $\mathcal{P}$ of the illumination segments in all cells intersected by SC or SPs are known. For applying the sub-cell discretization for the cell, the volume of each of the slices $V_m^{\vec{i}}$ and the area that the slices cut out of the cell faces $A_{m,F}^{\vec{i}}$ need to be computed.

Volume computation

The volume $V_m^{\vec{i}}$ inscribed within the convex hull $\mathcal{P}_m$ may have any kind of polyhedral shape. In order to automate the computation of its volume, the segment is therefore decomposed into simple subvolumes. In the described discretization, the volume is tessellated into T tetrahedrons utilizing the geompack software package[52]. The volume $V_m^{\vec{i}}$ is then computed as the sum of the tetrahedras' volumes:

$$\begin{aligned} V_m^{\vec{i}} &= \sum_{t=1}^{T} V_{m,t}^{\vec{i}} \\ V_{m,t}^{\vec{i}} &= \frac{1}{6}\left|\det\left[\vec{r}_{t,2}-\vec{r}_{t,1}, \vec{r}_{t,3}-\vec{r}_{t,1}, \vec{r}_{t,4}-\vec{r}_{t,1}\right]\right|, \end{aligned} \tag{4.43}$$

where $\vec{r}_{t,l},\ l=1,...4$, are the four corner points of the tetrahedron t, det takes the determinant of the matrix argument and $\vec{r}_{t,l}-\vec{r}_{t,1}, l=2,3,4$ are the columns of the said matrix.

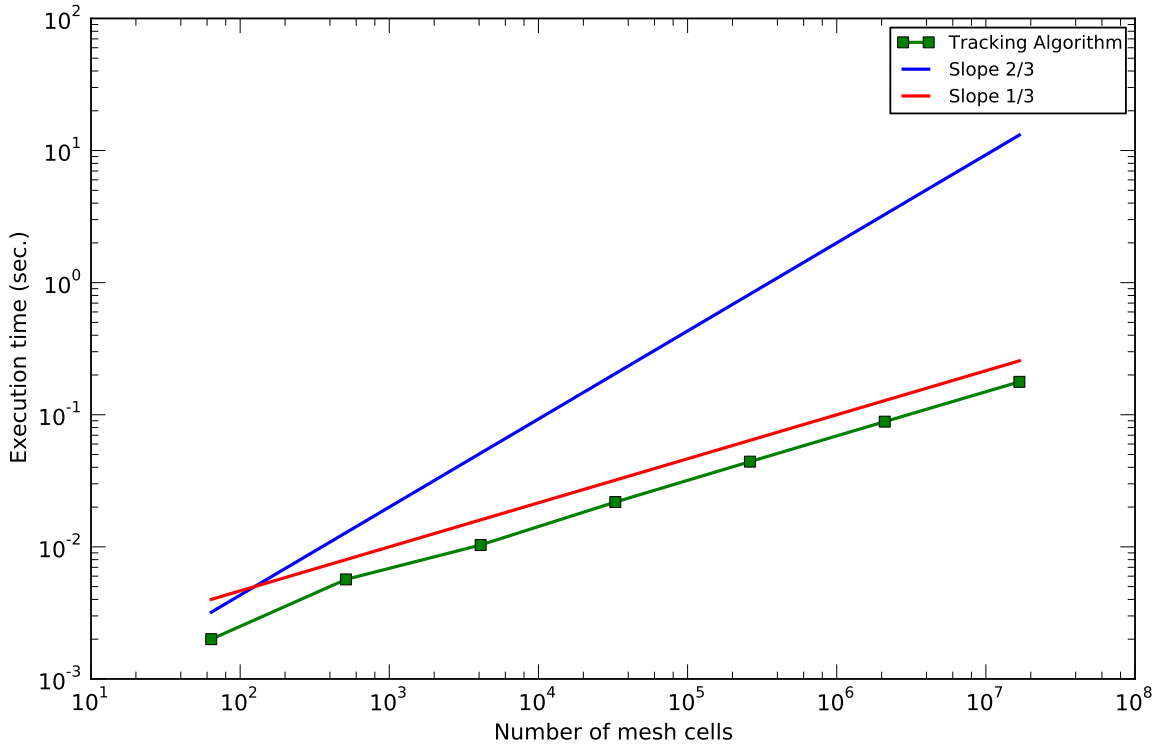


Figure 4.3: Scaling of tracking algorithm for test case with $I = J = K$ and S_4 level symmetric quadrature. The plotted execution time is the average of the execution times obtained for the different angular directions in the S_4 quadrature set.

Face area computation

For the face area computation, consider first Fig. 4.4 depicting a face that is intersected by the singular characteristic (left) and a face that is intersected by only one singular plane (right). In Fig. 4.4 the red lines represent the intersection of the singular planes with the respective cell face delineating the segment's areas $A^{\vec{i}}_{m,F}$ from each other. The thus created polygons feature either three, four, or five corners so that easy formulae can be devised for the computation of their areas. In case the polygon is a triangle, the area can be computed using Heron's formula:

$$\begin{aligned} A_\Delta &= \sqrt{s(s-a)(s-b)(s-c)} \\ s &= \frac{a+b+c}{2}, \end{aligned} \tag{4.44}$$

where a, b and c are the edge lengths of the triangle. In case a quadrilateral or pentagram

is encountered, the polygon is triangulated using the geompack software package[52], and the area can be computed as the sum of the triangles' areas:

$$A^{\vec{i}}_{m,F} = \sum_{t=1}^{T} A_{\Delta,t}. \tag{4.45}$$

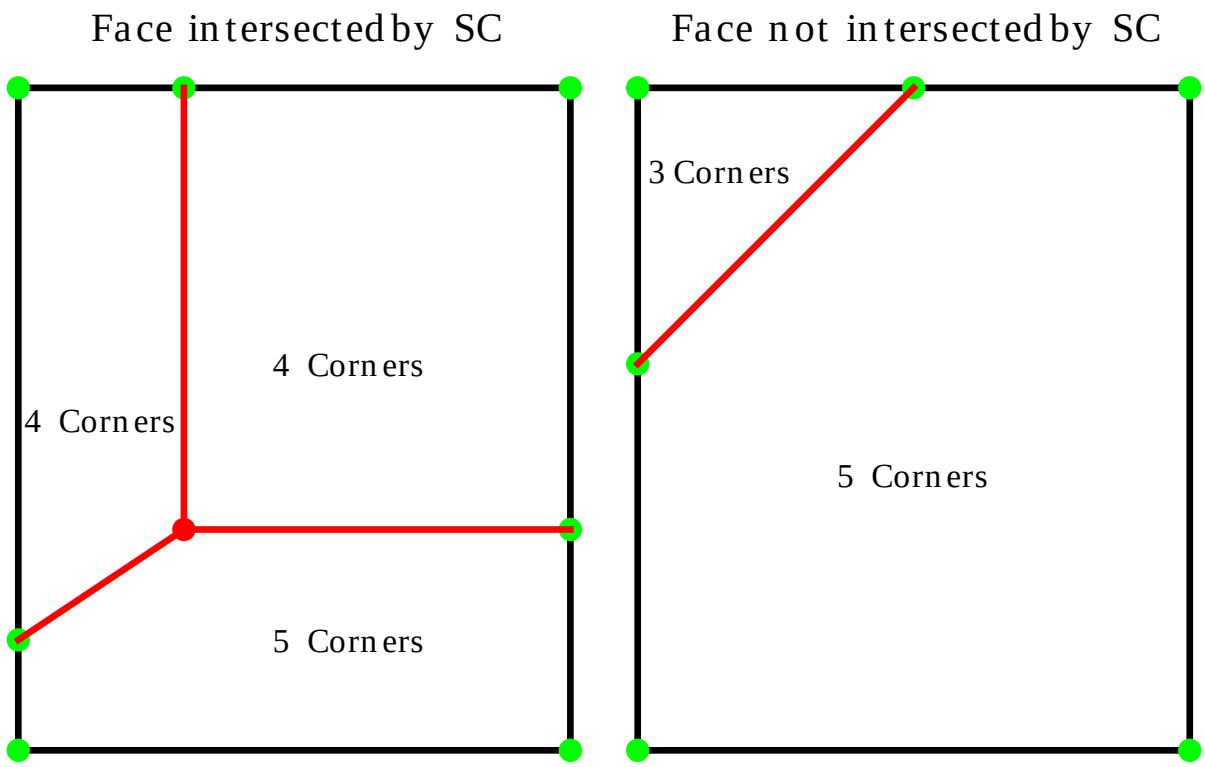


Figure 4.4: Decomposition of faces intersected by the singular characteristic and all singular planes (left) and single singular plane (right). Faces can always be decomposed into simple polygons featuring three, four, or five corners.

Subcell Discretization via the Step Method

First, the S_N equation is integrated over the extent of the sub-cell to obtain the subcell balance

equation:

$$\sum_F A_{m,F}^{\vec{i}} \hat{n}_F^T \hat{\Omega}_n \bar{\psi}_{n,m,F}^{h,\vec{i}} + \sigma_t V_m^{\vec{i}} \bar{\psi}_{n,m}^{h,\vec{i}} = V_m^{\vec{i}} S^{h,\vec{i}}, \ m = [W,E],[S,N],[B,T]. \tag{4.46}$$

Equation 4.46 is exact but underdetermined because the outflow face fluxes and the cell-averaged angular flux are unknowns and there is only one equation to determine them. Therefore, the step approximation given by

$$\bar{\psi}_{n,m,F}^{h,\vec{i}} = \bar{\psi}_{n,m}^{h,\vec{i}} \text{ for } \hat{n}_F^T \hat{\Omega}_n > 0, \tag{4.47}$$

is used to make Eq. 4.46 amenable for solution. Solving for the cell-averaged angular flux gives:

$$\bar{\psi}_{n,m}^{h,\vec{i}} = \frac{V_m^{\vec{i}} S^{h,\vec{i}} + \sum\limits_{\hat{n}_F^T \hat{\Omega}_n < 0} A_{m,F}^{\vec{i}} \left| \hat{n}_F^T \hat{\Omega}_n \right| \bar{\psi}_{n,m,F}^{h,\vec{i}}}{\sigma_t V_{m,t}^{\vec{i}} + \sum\limits_{\hat{n}_F^T \hat{\Omega}_n > 0} A_{m,F}^{\vec{i}} \hat{n}_F^T \hat{\Omega}_n}. \tag{4.48}$$

The outflow face-averaged fluxes are then computed using Eq. 4.47. Finally, for computing the cell-averaged scalar flux within the original cell a volume weighted average is utilized:

$$\bar{\phi}^{h,\vec{i}} = \sum_{n=1}^{N} w_n \sum_m \left(\frac{V_m^{\vec{i}}}{V^{\vec{i}}} \bar{\psi}_{n,m}^{h,\vec{i}} \right). \tag{4.49}$$

The set of equations 4.47, 4.48, and 4.49 completely determine the SCT-Step method.

For the implementation of the SCT-Step method, the mesh sweep has to account for the possibility of multiple distinct outflow segments on a single cell face as, for example, depicted in Fig. 4.4. It is at the heart of the algorithm that the outflow averages are not mixed across the boundaries imposed by the singular planes because mixing would defeat the initial purpose of this algorithm: separating the solution slices illuminated by different boundary faces. Within the described work, the SCT-Step algorithm was included into a standard sweep algorithm that sweeps the cells in a certain order dependent on $\hat{\Omega}_n$, for example the x-index runs fastest, followed by the y-index and finally the z-index runs slowest.

The subcell expressions, Eqs. 4.47, 4.48 and 4.49, are applied locally, i.e. whenever a cell intersected by at least a singular planes is encountered, Eqs. 4.47 and 4.48 are used to compute the segment's outflow and cell averages. Then the segment volume-averaged flux $\bar{\psi}_{n,m}^{h,\vec{i}}$ is immediately collapsed into one cell-averaged scalar flux using Eq. 4.49.

This is in contrast to the treatment of the face-averaged fluxes, because they need to be stored by illumination segment. Take for example a face that is intersected by the singular characteristic as depicted on the left in Fig. 4.4: In this case, the cell downstream across this

face from the one just solved, will be intersected by the singular characteristic, and the three face-averaged segment fluxes will be needed as input for the three instances of Eq. 4.48 to be solved. Similarly, for a face intersected by a single singular plane, two solution segments exist and the two face-averaged segment fluxes will be required as input.

In the absence of reflective boundary conditions, at most $n_a = J \cdot K + J + 1$ angular face information sets need to be stored (compared to at least $I \cdot J \cdot K$ scalar fluxes), and therefore the SCT-Step method does not increase the memory consumption significantly to store three instead of one angular face flux per cell face. The memory consumption for solving the angular face fluxes therefore increases to $n'_a = 3n_a$.

The SCT-Step method does not superimpose an additional grid over the Cartesian mesh, it merely splits a mesh cell appropriately into segments and collapses them immediately before the cell's solution is complete. It is important to contrast this to the idea of creating an unstructured mesh specifically to isolate the illumination segments from each other e.g. as suggested in [11]. The disadvantage of this latter approach is that the mesh becomes dependent on the angular direction $\hat{\Omega}_n$ so that, for a practical algorithm, separate meshes need to be created for each discrete ordinate and, in addition, restriction and prolongation operators need to be devised to exchange information between these meshes.

The SCT-Step method is added to the selection of promising discretization methods because it is expected to perform well in C_0 configurations where it is expected to restore cell-wise convergence.

Chapter 5

Numerical Results

This chapter discusses numerical results obtained from the MMS, Lathrop and thick diffusion limit test cases. The purpose of this chapter is twofold: first, the numerical results are directly used to discuss the benefits and detriments of the contending methods culminating into a qualitative ranking of what method performs well given a specific list of desired qualities. Second, it sets the stage for the development of the decision metric by providing the necessary data to compute a predictive fitness value for a particular application. In section 5.1 results from the MMS test case are discussed, followed by results from Lathrop's test case in section 5.2 and finally in section 5.3 both analysis and numerical results are provided regarding the possession of the thick diffusion limit. Section 5.4 then concludes this chapter with a qualitative comparison of the contending methods regarding the discussed performance aspects.

5.1 Accuracy and Efficiency: The MMS Test Case

Within this section, results from the MMS test suite are presented. First, the notion of efficiency is introduced, followed by a description of the nomenclature and choice of parameters for the various test cases, and finally a discussion and comparison across methods is performed based on the obtained data.

5.1.1 Efficiency of Discretization Methods

Efficiency is the ability of a discretization method to produce accurate results within a short execution time. The two obvious mechanism that improve the efficiency are the reduction of execution time, i.e. reduction of the grind time, or increase of accuracy, i.e reduction of the error on a given mesh.

Plotting the error measured in some norm versus the execution time, a strictly more efficient method is characterized by its curve laying below the one of the less efficient method. However,

under certain circumstances, e.g. when pre-asymptotically increasing errors or dips in the error versus mesh refinement curves are observed, two method's curves can intersect. In this case it is unclear which method is the more efficient. A more practical approach to defining efficiency is therefore to fix either execution time ("I need the results by this date/time.") or error ("The discretization error must be smaller than this threshold") and define the more efficient method as that with a smaller error or execution time, respectively.

The assessment of efficiency has gained prominence in the comparison of various variance reduction methods for Monte Carlo simulations[62], [63]. Some variance reduction techniques are designed to decrease the variance, others control the history population which is intended to reduce execution time, either strategy may or may not increase execution time and variance.

For a fair comparison of various variance reduction techniques the FOM is defined:

$$\text{FOM} = \frac{1}{vT}, \tag{5.1}$$

where v is the variance of the response of interest and T is the total execution time. Asymptotically, the FOM approaches a constant value because the execution time becomes proportional to the number of histories and the variance becomes inversely proportional to the number of histories in the asymptotic regime. The higher the FOM, the more efficient the Monte-Carlo method. Comparison of two Monte-Carlo methods (with different variance reduction techniques) is strictly valid for the asymptotic regime after both FOMs plateaued.

For realistic S_N problems, the irregularity of the exact solution limits the attainable order of accuracy of utilized spatial discretization methods. Therefore, given a smoothness C_p and assuming the solution is in the asymptotic regime, the error follows:

$$\|\epsilon\| = C_\epsilon h^\lambda, \tag{5.2}$$

where $h = \max_{\vec{i}} (\Delta x_i, \Delta y_j, \Delta z_k)$, C_ϵ is a constant independent of h, and λ depends on p and the utilized error norm. Further, the total execution time per source iteration of the discretization method is the product of the number of cells n_c, number of angular directions N, and the grind time Δt_g:

$$T = n_c N \Delta t_g. \tag{5.3}$$

It is observed within this work that the number of source iterations required to converge to a given tolerance do not depend on the discretization method, expansion order, or mesh spacing h. Certainly, this is a consequence of basing the stopping criterion solely on the cell-averaged scalar fluxes as opposed to all moments. However, as the number of iterations to successfully terminate the source iterations only depends on the stopping criterion ϵ_s and the problem configuration, we can discuss performance of different methods based on the execution time per

source iteration.

Assuming uniform mesh refinement, the number of mesh cells and h are asymptotically related by:

$$h = C_h n_c^{-1/3}, \tag{5.4}$$

where C_h depends on the mesh alone. Defining $\tilde{C}_h = C_h^{\lambda}$ and combining Eqs. 5.2 through 5.4 leads to:

$$\|\epsilon\| T^{\lambda/3} = C_\epsilon \tilde{C}_h N^{\lambda/3} \Delta t_g^{\lambda/3} = \text{const.} \tag{5.5}$$

The left hand side of Eq. 5.5 is independent of h and therefore the right hand side, given the solution is in the asymptotic regime, must also level off. It is desirable that the FOM should increase with improved performance, hence we used the reciprocal of Eq. 5.5 as FOM for S_N spatial discretization methods:

$$\text{FOM} = \frac{1}{\|\epsilon\| T^{\lambda/3}}. \tag{5.6}$$

The smaller the error and the execution time, the more efficient the discretization methods and the larger the FOM. Therefore, Eq. 5.6 is similar in its meaning for comparison of discretization methods as Eq. 5.1 is for Monte-Carlo methods.

5.1.2 Nomenclature of the Test Cases

Table 5.1: Variations in parameter space associated with the MMS test cases I through VII. The domain ranges from $x \in [0, X]$, $y \in [0, Y]$, and $z \in [0, Z]$.

Case number	σ_t	c	X (cm)	Y (cm)	Z (cm)	Comment
I	1.0	0.2	1	1	1	-
II	2.0	0.2	4	4	4	-
III	2.0	0.2	10	10	10	only C_1
IV	2.0	0.2	20	20	20	only C_1
V	1.0	0.8	1	1	1	only C_1
VI	1.0	0.2	1.4	1	0.8	-
VII	1.0	0.2	2	0.2	0.2	-

For assessment of the accuracy and efficiency of the set of discretization methods, a test harness covering a range in parameter space is set up using the three-dimensional MMS code MMS3D. The parameters that are being varied are the domain optical thickness, the scattering

Table 5.2: Boundary conditions and auxiliary source for C_0 and C_1 cases.

Smoothness	Q	ψ_W	ψ_S	ψ_B
C_0	4.8	3.0	2.0	1.0
C_1	1.0	0.0	0.0	0.0

ratio, and the domain aspect ratio. The utilized cases are listed in Table 5.1. To fully specify a test case, a parameter variation is selected from Table 5.1 and paired with the desired smoothness, e.g. C_0(I) uses the parameters from test case I with a discontinuous exact solution. The solution's smoothness is controlled by the inflow fluxes on the West, South, and Bottom faces, and the auxiliary distributed source Q. In Table 5.2 utilized values of the auxiliary source Q and the boundary conditions are listed for C_0 and C_1 problem setups.

All cases are meshed using a uniform spatial mesh featuring $4 \cdot 2^l, l = 0, ..., 6$ mesh cells per dimension.

The particular test cases are selected to cover a range in parameter space. Case I is the base case, cases II-IV gradually increase the domain optical thickness, case V varies the scattering ratio and finally cases VI and VII increase the domain optical aspect ratio.

5.1.3 Dependence of the Convergence on Norm, Smoothness and Method's Order

Table 5.3: Rate of convergence for C_0(I) and C_1(I) test case solved with AHOTN method of order one through three. The rate of convergence is computed as the slope of the last two plotted points within each graph.

	C_0				C_1			
	L_1	L_2	L_∞	Integral	L_1	L_2	L_∞	Integral
AHOTN-1	0.23	0.17	< 0	3.01	1.08	1.00	0.85	3.02
AHOTN-2	< 0.01	0.04	< 0	3.49	1.02	0.91	0.57	3.49
AHOTN-3	0.16	0.11	< 0	2.99	1.26	1.24	0.79	2.98

This section illustrates that different norms in conjunction with different solution smoothness result in different convergence rates of the solution. If no special precautions are taken, the

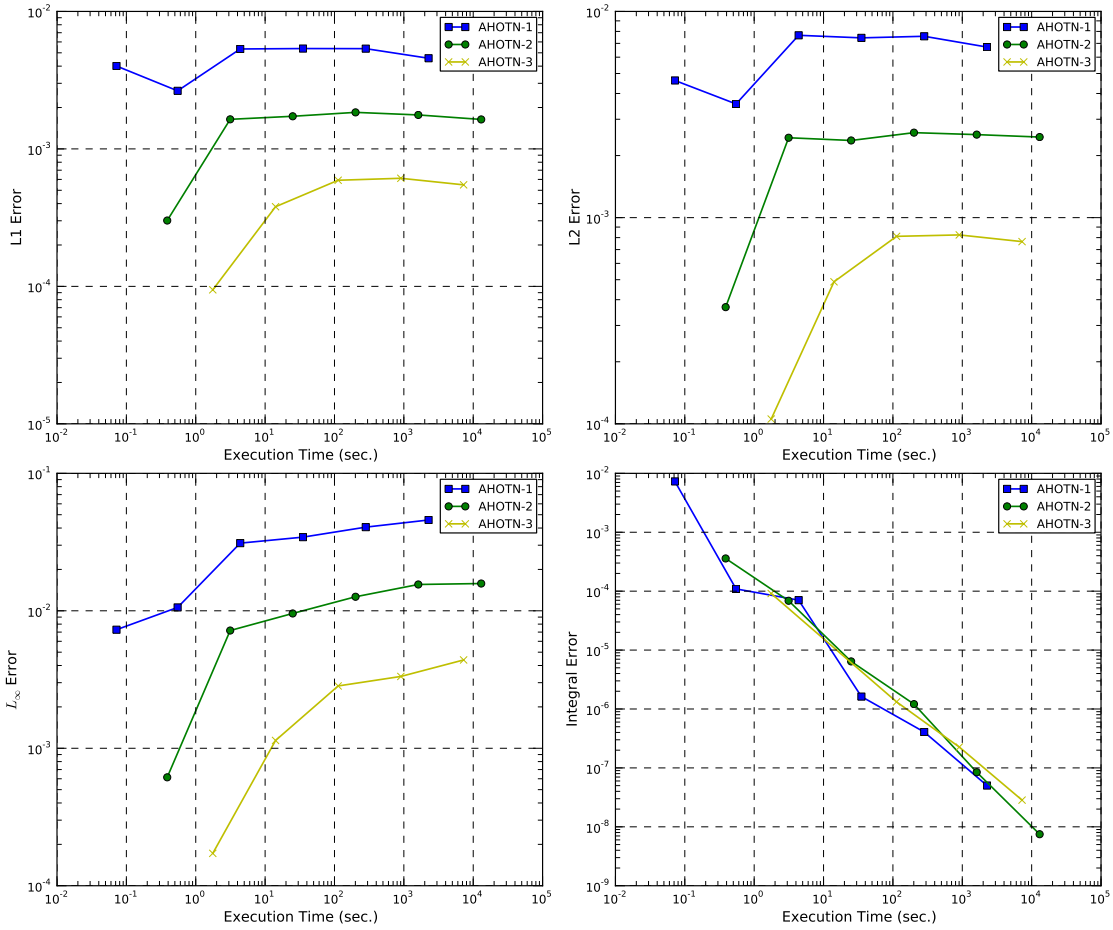


Figure 5.1: Convergence of the AHOTN method of orders one through 3 for the C_0(I) test case.

convergence rate in any norm is limited by the solution smoothness and cannot be increased by increasing the method's order even if the error magnitude itself is decreased. This is illustrated in Figs. 5.1 and 5.2, which depict the error measured in the discrete L_1 (upper left), L_2(upper right), L_∞(lower left), and integral norms (lower right) versus execution time for the C_0(I) and C_1(I) test cases, respectively. The results in these plots are obtained using the AHOTN method of order one, two, and three on uniform meshes of size $h = 0.5^l, l = 2, ..., 7$. Figures 5.1 and 5.2 are augmented by Table 5.3, listing the convergence orders for the AHOTN method depending on the employed error norm and the smoothness of the underlying exact solution.

A word of caution regarding rates of convergence is in order here: convergence rates are truly

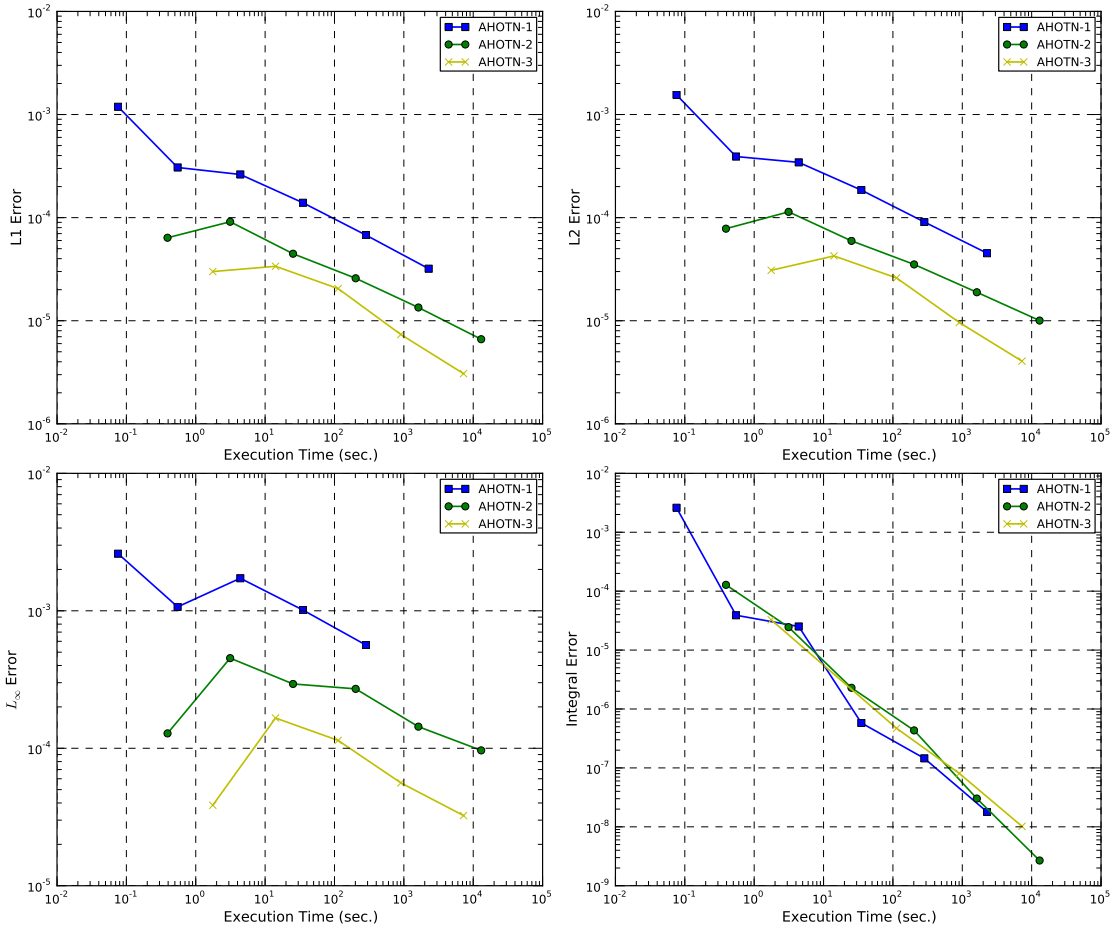


Figure 5.2: Convergence of the AHOTN method of orders one through 3 for the C_1(I) test case.

meaningful only in the asymptotic regime where the error is dominated by the leading order term. In contrast to two-dimensional results, as for example reported in [1] and [53], sufficient mesh refinement for reaching the asymptotic regime is often impossible in three-dimensional geometry, especially in the presence of discontinuous exact solutions (C_0). Therefore, the stated rates of convergence for the C_0 test cases should be regarded with caution. Errors in the asymptotic regime appear as straight lines in a log-log error versus mesh spacing/execution time plot. A typical indicator could be three consecutive points that deviate only insignificantly from a line drawn through two of them. From Fig. 5.1 it is apparent that none of the errors are truly in the asymptotic regime. When looking at the C_0 results obtained with the AHOTN

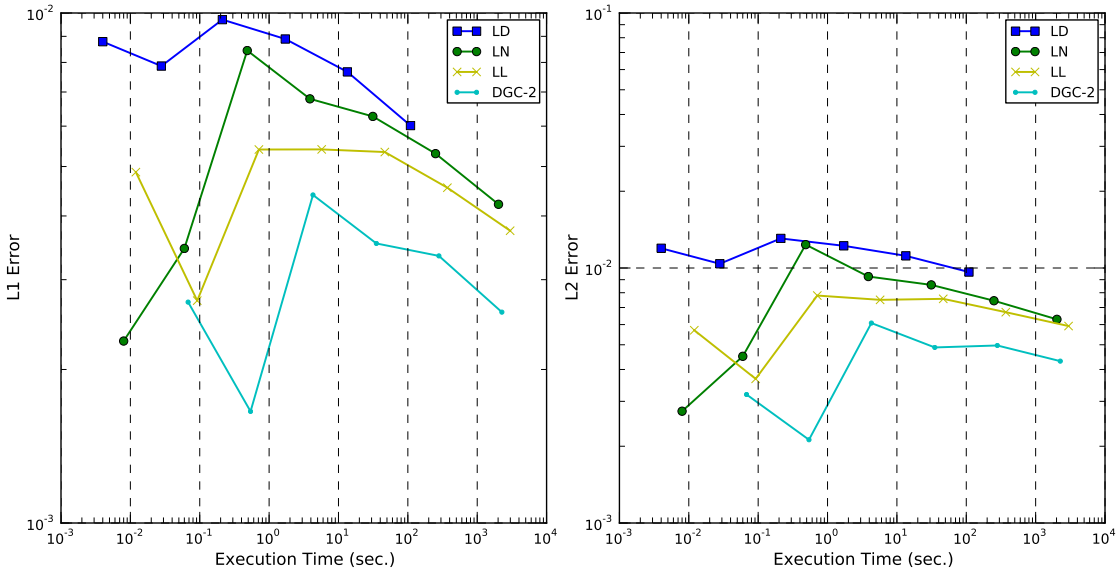


Figure 5.3: Convergence of selected other method of orders one and two for the C_0(I) test case.

method in Fig. 5.1 measured in the L_1 and L_2 norm, one could draw the conclusion that the error does not converge with mesh refinement. However, with further mesh refinement that is impossible to realize due to computational memory limitations, convergence to the exact solution is expected. For further support of convergence of the error measured in the L_1 and L_2 norm, Fig. 5.3 depicts errors associated with solutions of various discretization methods of orders one and two for the C_0(I) test case. On coarse meshes, the error may increase with mesh refinement or even decrease initially and then increase again. This behavior is associated with non-asymptocity of the solution and we shall elaborate on it in section 5.1.4. With further mesh refinement, the error begins to follow a straight line indicating slow but steady decrease of the error with further mesh refinement and thus convergence in the L_1 and L_2 norms.

As pointed out in Refs. [21], [22], and [11] the rate of convergence in realistic transport problems is (1) limited by the smoothness of the underlying exact solution, (2) may be fractional (not integer) and may depend on the order and type of the utilized norm. Figures 5.1 and 5.2 illustrate these facts. Even when increasing the expansion order from one to three, the observed orders of accuracy do not increase for each of the two degrees of smoothness and four norms. Also, the C_0 case exhibits significantly smaller observed orders of accuracy for all L_p norms and non-convergence of the discrete L_∞ norm, i.e. lack of cell wise convergence. All these facts are well reported in the above references and this section only serves the purpose of showing that these references' conclusions made in two-dimensional configuration are valid for three spatial

dimensions as well.

An additional observation from Figs. 5.1 and 5.2 is that the integral error norm exhibits an observed order of convergence of about three for both the C_0 and C_1 cases. This is remarkable for two reasons: (1) the observed accuracy is much larger than the one observed for L_p norms and (2) the rate is identical for the C_0 and C_1 test cases and does not vary significantly with increasing Λ.

In case the underlying solution is discontinuous, the mode of convergence manifests the isolation of the cells intersected by the singular characteristic/singular planes that form a region of measure zero[1] as $h \to 0$. As stated in section 4.4, without special precaution, a numerical method cannot obtain the right answer (or approximation thereof) for a cell that contains a discontinuity. The fraction of cells n_{SP} affected by singular planes is proportional to

$$n_{SP}/n_c \propto n_c^{\frac{d-1}{d}}/n_C \propto n_c^{\frac{-1}{d}}, \tag{5.7}$$

where n_c is the total number of cells and d is the dimensionality of the problem. The fraction of cells affected by the singular planes in 3D decreases slower than for problems in two spatial dimensions.

We conjecture the reason for significant mesh refinement to be necessary for a convergent behavior to emerge is that initially all or at least most mesh cells are affected by the singular characteristic and/or singular planes, and only when this set of cells is sufficiently isolated can a reduction of error take place. Initial increase in error is caused by cancellation of errors on the coarser meshes. The exact mechanism that leads to cancellation of errors is discussed in section 5.1.4. In contrast, the fraction of cells affected by the singular characteristic in two spatial dimensions as reported in [1] diminishes quicker than in 3D.

5.1.4 Cancellation of Errors

Cancellation of error is a process that occurs when pointwise errors are averaged over a subregion of the domain before applying an absolute value. When comparing the two error norms $\|\cdot\|_{c,\psi,2}$ and $\|\cdot\|_{d,\phi,2}$ given by Eqs. 3.42 and 3.43, respectively, we recognize that for the latter error norm the difference of exact and numerical solution is first averaged before taking an absolute value, allowing positive and negative (pointwise) contributions to offset each other. Simplifying to one spatial dimension, this describes a case where:

$$\int_{\mathcal{D}} dx\,|\epsilon| > \sum_i \Delta x_i \left| \frac{1}{\Delta x_i} \int_{x_{i-1}}^{x_i} dx\epsilon \right|. \tag{5.8}$$

A candidate measure for the cancellation of error Ca is:

$$\mathrm{Ca} = \int_{\mathcal{D}} dx\, |\epsilon| - \sum_{i=1}^{I} \left| \int_{x_{i-1}}^{x_i} dx \epsilon \right|. \tag{5.9}$$

Breaking up the first integral over the domain into a sum of integrals over the cells gives:

$$\mathrm{Ca} = \sum_{i=1}^{I} \left(\int_{x_{i-1}}^{x_i} dx\, |\epsilon| - \left| \int_{x_{i-1}}^{x_i} dx \epsilon \right| \right) = \sum_{i=1}^{I} \mathrm{Ca}_i. \tag{5.10}$$

Further, it holds that:

$$\mathrm{Ca} = \sum_{i} \mathrm{Ca}_i \leq I \max_i \mathrm{Ca}_i \leq \frac{\tilde{C}}{h} \max_i \mathrm{Ca}_i, \tag{5.11}$$

where $\tilde{C}$ is a constant independent of h. It can be shown that Ca_i decreases with mesh refinement if the function ϵ possesses bounded first partial derivatives:

$$\begin{aligned}
\mathrm{Ca}_i &= \int_{x_{i-1}}^{x_i} dx \left| \epsilon\left(x_{mid}\right) + \left[\frac{d\epsilon}{dx}\right]_{x_{mid}} \left(x - x_{mid}\right) \right| - \Delta x_i \bar{\epsilon} \\
\mathrm{Ca}_i &\leq \Delta x_i \left(\underbrace{\epsilon\left(x_{mid}\right) - \bar{\epsilon}}_{\mathcal{O}\left(\Delta x_i^2\right)} \right) + \left| \left[\frac{d\epsilon}{dx}\right]_{x_{mid}} \right| \underbrace{\int_{x_{i-1}}^{x_i} dx \left| x - x_{mid} \right|}_{\Delta x_i / 4} \\
\mathrm{Ca}_i &\leq \tilde{C}_i' h^2,
\end{aligned} \tag{5.12}$$

where $\tilde{C}'$ is a constant independent of h. Using Eq. 5.12 and Eq. 5.11 gives:

$$\mathrm{Ca} \leq \tilde{C}_i' \tilde{C} \max_i \; h. \tag{5.13}$$

Equation 5.13 demonstrates that cancellation of errors reduces as the mesh is refined if the error is differentiable over the domain.

Cancellation of error for C_0 Results

On very coarse meshes, most cells are either intersected by at least one of the singular planes or are in the range of influence of the singular planes. The main instrument for reduction of error is to isolate the cells affected by the singular planes. In other words, the global $L_p,\ p < \infty$ error norm decreases not because the error everywhere decreases, but because the volume comprising the non-decreasing error contributions decreases.

On coarse mesh cells, the pointwise distribution of the error can vary greatly over the extent

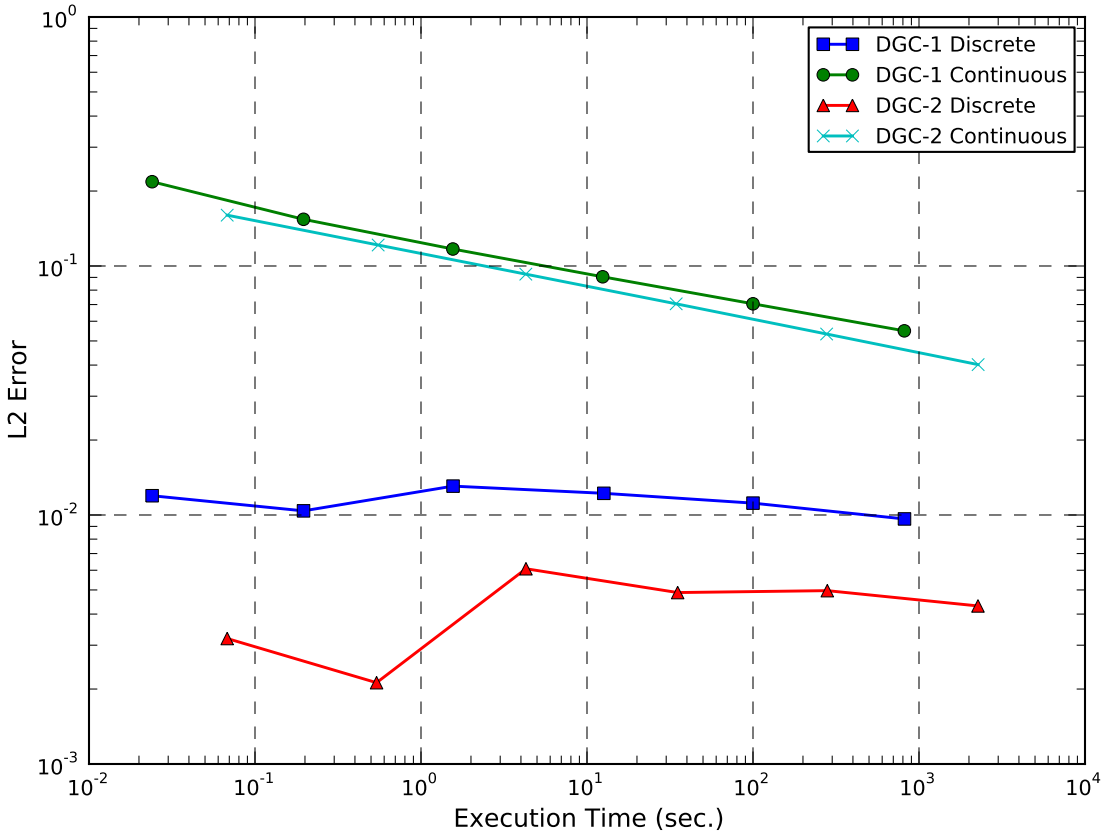


Figure 5.4: Comparison of continuous $\|\cdot\|_{c,\psi,2}$ norm and discrete $\|\cdot\|_{d,\phi,2}$ norm of the error for $C_0(\mathrm{I})$ test case solved using the complete DGFEM method of order $\Lambda = 1$.

of the cell. This situation supports cancellation of errors as demonstrated in Eq. 5.12. When the mesh is refined just once starting from a coarse mesh, positive and negative contributions may be isolated from each other such that one daughter mesh cell contains all (or most) of the positive and another contains all (or most) of the negative contributions. As averaging is performed on a mesh cell basis cancellation as on the parent mesh cannot occur and therefore the error increases with mesh refinement. Cancellation of error is impossible to predict. It may occur to a great extent on one mesh, but disappear with a single mesh refinement with the possibility to reemerge later in the mesh-refinement process. We may refer to this as the volatile behavior of cancellation of error.

Figure 5.4 illustrates the occurrence of cancellation of error when solving the $C_0(\mathrm{I})$ test case using the complete DGFEM method of orders $\Lambda = 1, 2$. When measured in a discrete L_2 norm applied to the scalar flux the error on coarse meshes does not follow a "well-behaved"

decreasing trend with mesh refinement but rather increases or even decreases and then increases again (DGC-2). When measuring the error in a norm that does not allow cancellation of errors (for example the $\|\cdot\|_{c,\psi,2}$ norm), the described behavior vanishes and the error follows a straight line from the outset.

Cancellation of error is more pronounced in the presence of non-smooth solutions because both the exact and the approximate solution are less well-behaved. Approximate solutions to non-smooth test-cases often feature unphysical oscillations in the vicinity of the non-smoothness. These oscillations actually benefit cancellation of error, because the shape of the pointwise error distribution will then oscillate around a very small mean but with potentially large amplitude.

Cancellation of error for Integral Norm Results

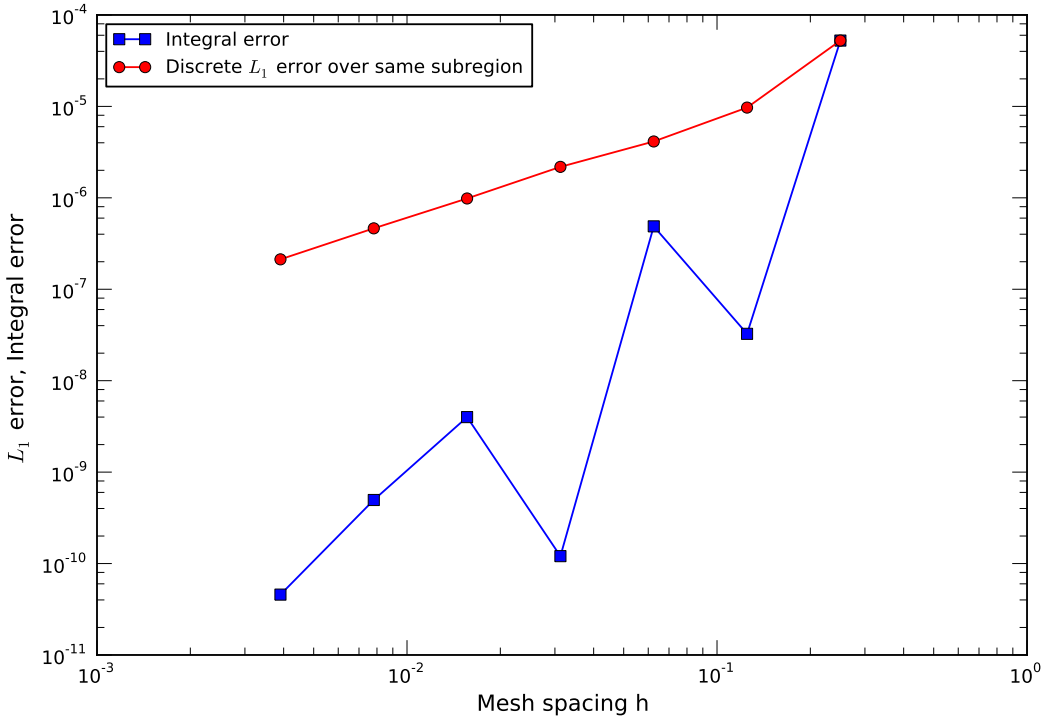


Figure 5.5: Integral error (1/8 subcube) and discrete L_1 scalar flux error (for the same region) for test C_1(I) solved with the Linear-Linear method.

Among the set of error norms utilized within this work the integral error norms allow

most cancellation of errors because absolute values are applied after fluxes are averaged over subdomains rather than single mesh cells. Therefore, cancellation of errors does not follow a bound as in Eq. 5.12 or more precisely h as used in Eq. 5.12 is constant for all meshes in the mesh refinement study.

Following Eq. 5.9, a measure for the cancellation of error for the integral error norm is given by:

$$\begin{aligned}
\mathrm{Ca}_I &= \int_{\mathcal{D}_s} dx\,|\epsilon(x)| - \left|\int_{\mathcal{D}_s} dx\epsilon(x)\right| \\
&= \left(\int_{\mathcal{D}_s} dx\,|\epsilon(x)| - \sum_{i\in\mathcal{D}_s}\left|\int_{\Delta x_i} dx\epsilon(x)\right|\right) + \left(\sum_{i\in\mathcal{D}_s}\left|\int_{\Delta x_i} dx\epsilon(x)\right| - \left|\int_{\mathcal{D}_s} dx\epsilon(x)\right|\right) \\
&= \mathrm{Ca} + \left(\sum_{i\in\mathcal{D}_s}\left|\int_{\Delta x_i} dx\epsilon(x)\right| - \left|\int_{\mathcal{D}_s} dx\epsilon(x)\right|\right). \qquad (5.14)
\end{aligned}$$

The first term in Eq. 5.14 is the measure of cancellation when going from a continuous L_1 norm to a discrete L_1 norm and it has been shown to decrease with mesh refinement in Eq. 5.13. Therefore, its contribution to the total Ca diminishes as the mesh is refined. The second term measures the cancellation when going from a discrete L_1 norm to an integral norm. It generally does not diminish with mesh refinement. Therefore, even in the limit $h \to 0$ cancellation of error does not vanish if the error is measured in an integral error norm.

As the first summand in Eq. 5.14 is difficult to compute and also diminishes with mesh refinement, we only illustrate the second summand via Fig. 5.5, which plots the integral error and $\|\|_{d,\phi,1}$ error for the lower, left eighth subcube for the C_1(I) case solved with the LL method. The space between the two curves is the amount of cancellation, Ca, associated with the second summand in Eq. 5.14.

Of special interest are the dips in the integral error norm's curve occurring for mesh refinement levels two and four. These dips are absent for the L_1 error curve. The dips in the integral error norm curve underscore the volatility of cancellation of error that may be present to varying degrees on two different meshes. However, Fig. 5.5 also shows a general trend that the difference between the integral error and the L_1 error is growing with mesh refinement suggesting that the cancellation between the L_1 and integral error norm consists of two parts: a consistent, monotonically growing part and a volatile part that is present only on a subset of the utilized meshes.

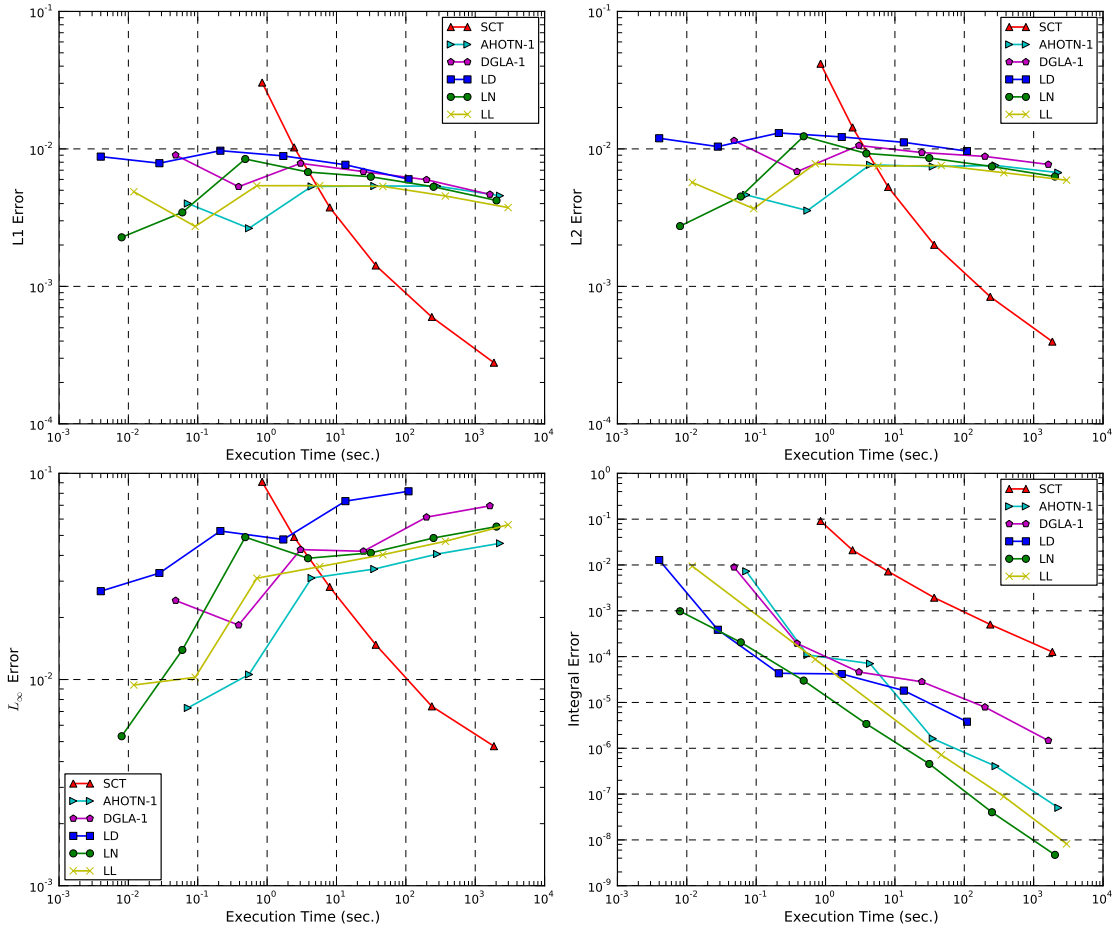


Figure 5.6: Comparison of the performance of the SCT method (order $\Lambda = 0$) and various other method of order $\Lambda = 1$ for the C_0(I) test case.

5.1.5 Performance of the SCT-Step Method

The implemented SCT method was created for mediating the detrimental effect that discontinuous exact solutions have on the accuracy of standard methods. Its performance under smoother conditions is not expected to be competitive from the computational efficiency perspective because it is only first order accurate (SP intersected cells are solved using the *Step method*). For the C_0(I) and C_0(II) test cases the SCT method's performance is compared (in various discrete norms) to various other methods of order one through three in Figs. 5.6 through 5.8.

In general, the SCT method exhibits a comparatively large error on coarse meshes and it

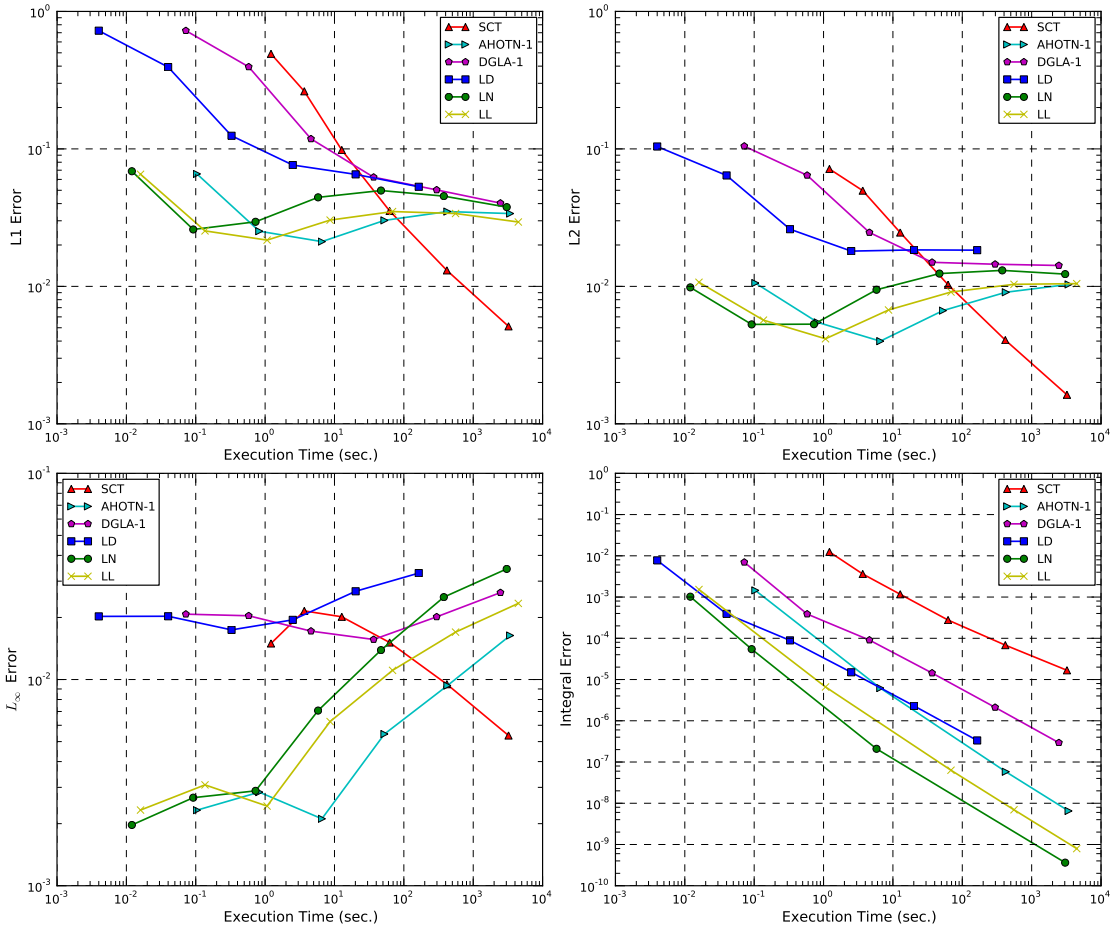


Figure 5.7: Comparison of the performance of the SCT method (order $\Lambda = 0$) and various other method of order $\Lambda = 1$ for the C_0(II) test case.

is also rather expensive because a large fraction of the execution time is used for the tracking procedure. Compared to other discretization methods, SCT's performance on the coarsest meshes is not competitive. However, for the first few mesh refinement steps the error diminishes faster than $\mathcal{O}(h)$ and the execution time does not increase over-proportionally fast, because the expensive tracking computation only scales with the cubic root of the number of mesh cells. The asymptotic rate of convergence is, however, only $\mathcal{O}(h)$ due to utilizing the *step* method for cells intersected by the SPs.

For both the C_0(I) and C_0(II) cases, the SCT method eventually becomes more efficient than all other methods (including high-order methods) in the L_1 and L_2 error norms: for the

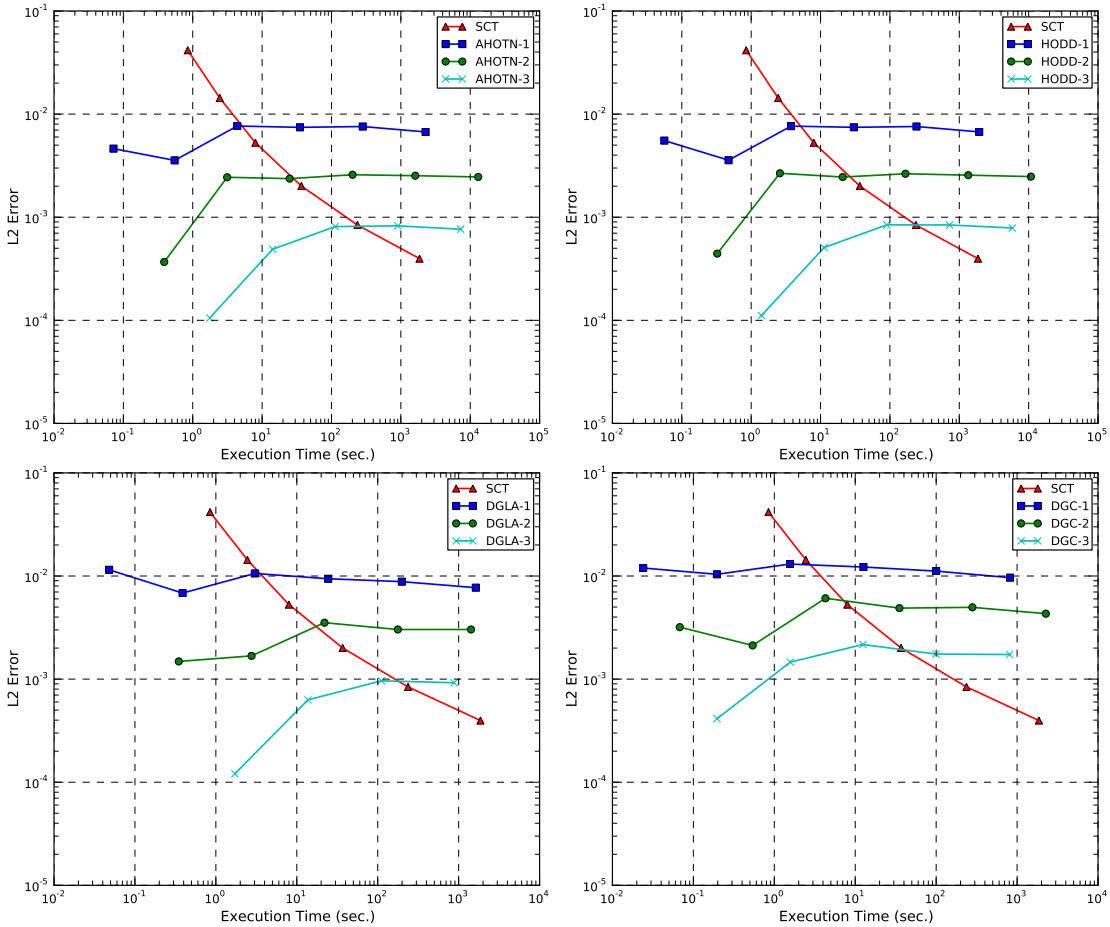


Figure 5.8: Comparison of the L_2 norm performance of the SCT method (order $\Lambda = 0$) and various other method of orders $\Lambda = 1$ to $\Lambda = 3$ for the C_0(II) test case.

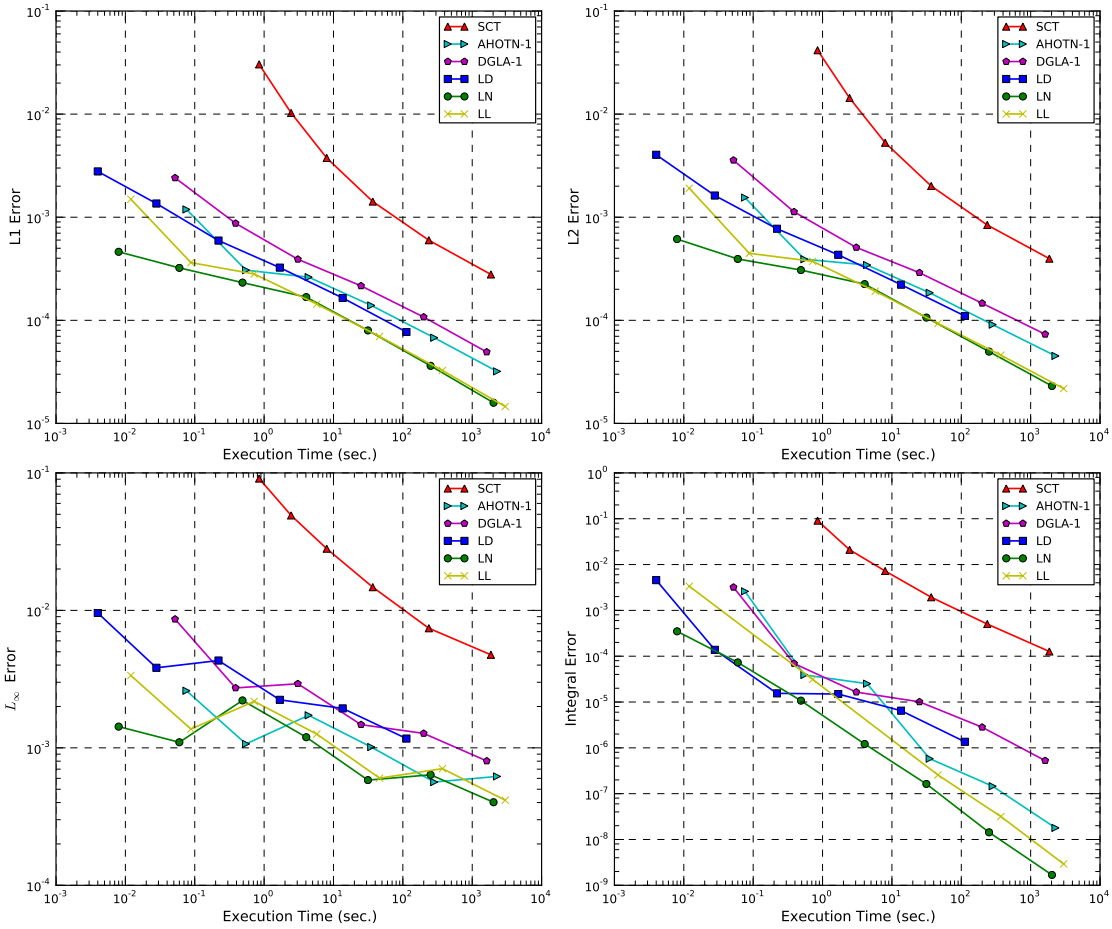


Figure 5.9: Comparison of the performance of the SCT method (order $\Lambda = 0$) and various other method of order $\Lambda = 1$ for the C_1(I) test case.

C_0(I) setup the cross-over occurs earlier than for the C_0(II) case. For both test cases, the SCT is the only viable alternative to reducing the non-integral discretization error significantly below 10^{-2} due to the extremely small orders of accuracy observed for the other competing methods.

In the L_∞ norm the SCT method restores cell-wise convergence and is thus the only viable method available in the set of selected schemes. This is a significant property of the SCT method for example for shielding applications where shadowing effects often lead to discontinuous solutions. If for example the dose rate is desired everywhere behind a shield, then convergence should be ensured everywhere to ensure that proper numerical estimates of the flux are obtained.

The SCT method is not competitive with the standard methods when the flux is at least continuous (C_1) or if the error is measured in the integral error norm regardless of the configuration's smoothness. For both scenarios the reason for this outcome is obvious: the integral norm allows a convergence order of about three, while C_1 problems allow a convergence order of unity. The SCT method invests additional resources into tracking SC and SPs but still the underlying *Step method* is only first order accurate; in addition the *Step method* is a really poor approximation, even among first order accurate methods.

In conclusion, the SCT method proved superior to standard methods for cellwise error norms when the flux is discontinuous. It is less efficient if the flux is at least C_1 because its current implementation is limited to first order accuracy and uses the inaccurate *Step method*. However, the SCT performance for the C_0 test cases impressively demonstrates the value of a larger convergence order. Using the SCT algorithm in conjunction with higher-order discretization schemes could create a method that allows for convergence orders equal to or larger than two.

An additional observation that does not pertain to SCT's efficiency is that it does not exhibit the irregular behavior, i.e. non-monotonic decrease in error with mesh refinement, typical of other discretization methods in the presence of discontinuous angular fluxes. This behavior was attributed to cancellation of error in section 5.1.4 where it is demonstrated that using an error norm that allows cancellation of errors is a necessary condition for the dips in in the error versus time curves to appear. Further, non-smoothness of the exact solution was found to be a factor supporting cancellation of errors. For example, in Fig. 5.6 the *SCT-Step method* curve does not suffer from pre-asymptotic behavior associated with cancellation of error.

5.1.6 HODD versus AHOTN

The HODD method is related to the AHOTN method in the sense, that for optically thin cells, the AHOTN method asymptotically approaches the HODD method (see section B.4). The HODD method executes about 10-15% faster than the AHOTN method of the same order as indicated by the grind times in Tables 4.1 and 4.2. For optically thin cells, the solutions

provided by the AHOTN and HODD methods are virtually identical; and therefore the HODD is slightly more efficient under these conditions (for example Fig. 5.10 C_1(I) test cases). However, the HODD method's advantage is marginal even under circumstances that favor it most. For optically thicker problems, the HODD method is significantly less accurate and therefore less efficient than the AHOTN method, which is reflected in the HODD curves depicting test cases C_1(III) and C_1(IV) in Fig. 5.10. The HODD curves exhibit much larger errors when the mesh cells are optically thick.

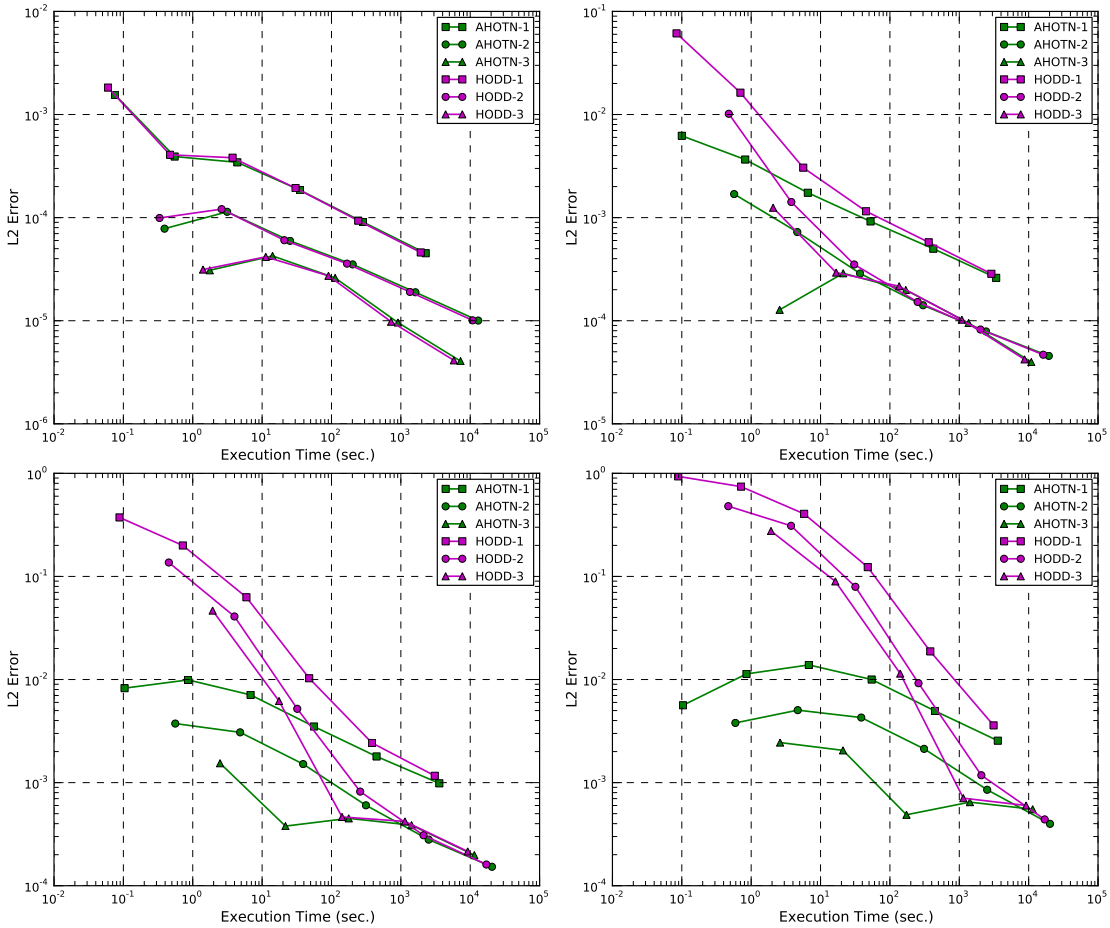


Figure 5.10: Comparison of the performance of the HODD and AHOTN methods of orders one through three measured in the discrete L_2 error norm. The four subplots depict results for the C_1(I)(upper left), C_1(II)(upper right), C_1(III)(lower left), and C_1(IV)(lower right) test cases.

The reason for HODD's dramatic failure in optically thick cells can be attributed to the fixed value of the spatial weight used in the corresponding WDD relationships, i.e. zero and infinity for even and odd expansion orders, respectively, that are suitable in the thin-cell limit but are inadequate for optically thick cells. To demonstrate this, let us look at the S_N transport equation in the limit as $\sigma_t \rightarrow \infty = c/\sigma_s$. Recall, the continuous S_N equations are give by:

$$\hat{\Omega}_n \cdot \nabla \psi_n + \sigma_t \psi_n = \frac{\sigma_s}{4\pi}\phi + \frac{q}{4\pi}. \tag{5.15}$$

The small parameter ϵ is defined as $\epsilon = 1/\sigma_t$ and the angular and scalar flux are postulated to be representable as a power series in ϵ:

$$\begin{aligned} \psi_n &= \sum_{p=0}^{\infty} \epsilon^p \psi_n^{[p]} \\ \phi &= \sum_{p=0}^{\infty} \epsilon^p \phi^{[p]}. \end{aligned} \tag{5.16}$$

Substituting Eq. 5.16 into Eq. 5.15 results in the following expression:

$$\sum_{p=0}^{\infty} \epsilon^p \hat{\Omega}_n \cdot \nabla \psi_n^{[p]} + \sum_{p=0}^{\infty} \epsilon^{p-1} \psi_n^{[p]} = \frac{c}{4\pi} \sum_{p=0}^{\infty} \epsilon^{p-1} \phi^{[p]} + \frac{q}{4\pi}. \tag{5.17}$$

Separating Eq. 5.17 by powers of ϵ and looking only at the ϵ^{-1} and ϵ^0 terms leads to:

$$\begin{aligned} \mathcal{O}(\epsilon^{-1}): \ \psi_n^{[0]} &= \frac{c}{4\pi}\phi^{[0]} \\ \mathcal{O}(\epsilon^0): \hat{\Omega}_n \cdot \nabla \psi_n^{[0]} + \psi_n^{[1]} &= \frac{c}{4\pi}\phi^{[1]} + \frac{q}{4\pi}. \end{aligned} \tag{5.18}$$

From the $\mathcal{O}(\epsilon^{-1})$ term we infer that $\psi_n^{[0]}$ is isotropic. Applying the quadrature operator $\sum_{n=1}^{N} w_n \cdot$ to Eqs. 5.18 and using Eq. 5.17 yields:

$$\begin{aligned} \phi^{[0]}(1-c) &= 0 \\ \frac{c}{4\pi}\left[\sum_{n=1}^{N} w_n \hat{\Omega}_n \cdot \nabla \phi^{[0]}\right] + \phi^{[1]}(1-c) &= q. \end{aligned} \tag{5.19}$$

Let us restrict our attention to two cases: (1) the scattering cross section is constant $\sigma_s = \text{const} \Rightarrow c = \mathcal{O}(\epsilon)$, and (2) $c = \text{const} << 1$ (we are outside of the diffusive regime!). In both

cases $1-c=\text{const} >> 0$ so we can multiply by $\tilde{C}=1/(1-c)$:

$$\begin{aligned} \phi^{[0]} &= 0 \\ \tilde{C}\frac{c}{4\pi}\underbrace{\left[\sum_{n=1}^{N} w_n \hat{\Omega}_n \cdot \nabla\phi^{[0]}\right]}_{=0} + \phi^{[1]} &= \tilde{C}q. \end{aligned} \tag{5.20}$$

Note, $\tilde{C} \neq f(\epsilon)$ for case (2) and $\lim_{\epsilon\to 0}\tilde{C}=1$ for case (1). Therefore, for both cases (1) and (2), the scalar flux to leading order in the limit $\sigma_t \to \infty$ will vary as

$$\phi \propto \frac{1}{\sigma_t}. \tag{5.21}$$

For a discretization method to yield reasonable solutions in the said limit, the same asymptotic behavior needs to be reproduced. For HODD, the nodal unknowns within the cells, i.e. the cell Legendre moments of the angular and scalar flux are collected in vectors $\vec{\psi}_n^h$ and $\vec{\phi}^h$, respectively, and analogously to the continuous case, expanded as follows:

$$\begin{aligned} \vec{\psi}_n^h &= \sum_{p=0}^{\infty} \epsilon^p \vec{\psi}_n^{h,[p]} \\ \vec{\phi}^h &= \sum_{p=0}^{\infty} \epsilon^p \vec{\phi}^{h,[p]}, \end{aligned} \tag{5.22}$$

where the vectors are ordered according to $\left(\vec{\psi}_n^h\right)_m = \psi_{n,\vec{m}}^h$ and $m = m_z + 1 + (\Lambda+1)\, m_y + (\Lambda+1)^2\, m_x$. Similarly, the face flux moments are collected in vectors and expanded into a power series in ϵ:

$$\vec{\psi}_{n,F}^h = \sum_{p=0}^{\infty} \epsilon^p \vec{\psi}_{n,F}^{h,[p]}. \tag{5.23}$$

The face moment vectors are ordered according to:

$$\begin{aligned} \left(\vec{\psi}_{n,E}^h\right)_m &= \psi_{n,E,\vec{m}^x}^h, \quad m = m_z + 1 + (\Lambda+1)\, m_y \\ \left(\vec{\psi}_{n,N}^h\right)_m &= \psi_{n,N,\vec{m}^x}^h, \quad m = m_z + 1 + (\Lambda+1)\, m_x \\ \left(\vec{\psi}_{n,T}^h\right)_m &= \psi_{n,T,\vec{m}^x}^h, \quad m = m_y + 1 + (\Lambda+1)\, m_x, \end{aligned} \tag{5.24}$$

with analogous expressions for the W, S, and B faces.

The HODD method consists of two sets of equations: balance equations and WDD equa-

tions. In this derivation Eqs. 4.29 and 4.30 will be used so that it is slightly more general than necessary because it includes the AHOTN method as well. Equation 4.29 can be written in matrix notation as:

$$\begin{aligned} &\left(\mathbf{P}_n^{+x}\vec{\psi}_{n,E}^h - \mathbf{P}_n^{-x}\vec{\psi}_{n,W}^h\right) + \left(\mathbf{P}_n^{+y}\vec{\psi}_{n,N}^h - \mathbf{P}_n^{-y}\vec{\psi}_{n,S}^h\right) + \left(\mathbf{P}_n^{+z}\vec{\psi}_{n,T}^h - \mathbf{P}_n^{-z`}\vec{\psi}_{n,B}^h\right) \\ + \quad &\mathbf{D}\vec{\psi}_n^h + \sigma_t\vec{\psi}_n^h = \frac{c}{4\pi}\sigma_t\vec{\phi}^h + \frac{\vec{q}}{4\pi}, \end{aligned} \tag{5.25}$$

where:

$$\begin{aligned} \mathbf{P}_n^{+r}: &\quad (\Lambda+1)^3 \times (\Lambda+1)^2 \text{ matrix containing the term: } s_{\mu_r}\frac{|\mu_r|}{\Delta_r} \\ \mathbf{P}_n^{-r}: &\quad (\Lambda+1)^3 \times (\Lambda+1)^2 \text{ matrix containing the term: } s_{\mu_r}\frac{|\mu_r|}{\Delta_r}(-1)^{m_k} \\ \mathbf{D}: &\quad (\Lambda+1)^3 \times (\Lambda+1)^3 \text{ matrix containing the terms:} \\ &\quad -2s_\mu\frac{|\mu_n|}{\Delta x_i}\sum_{l=0}^{\left[\frac{m_x-1}{2}\right]}(2m_x-4l-1) \\ &\quad -2s_\eta\frac{|\eta_n|}{\Delta y_j}\sum_{l=0}^{\left[\frac{m_y-1}{2}\right]}(2m_y-4l-1) \\ &\quad -2s_\xi\frac{|\xi_n|}{\Delta z_k}\sum_{l=0}^{\left[\frac{m_z-1}{2}\right]}(2m_z-4l-1)\,. \end{aligned} \tag{5.26}$$

The weighted Diamond Difference relationships can be written in matrix notation as:

$$\frac{1+\alpha_{n,r}}{2}\vec{\psi}_{n,+r}^h + \frac{1-\alpha_{n,r}}{2}\vec{\psi}_{n,-r}^h = \mathbf{K}_n^r\vec{\psi}_n^h + \alpha_{n,r}\mathbf{H}_n^r\vec{\psi}_n^h, \tag{5.27}$$

where:

$$\begin{aligned} \mathbf{K}_n^r: &\quad (\Lambda+1)^3 \times (\Lambda+1)^3 \text{ matrix containing the term: } \sum_{l=0,even}^{\Lambda}(2l+1) \\ \mathbf{H}_n^r: &\quad (\Lambda+1)^3 \times (\Lambda+1)^3 \text{ matrix containing the term: } s_{\mu_r}\sum_{l=1,odd}^{\Lambda}(2l+1)\,. \end{aligned} \tag{5.28}$$

Solving Eq. 5.27 for the outflow face moments yields:

$$\vec{\psi}_{n,+r}^h = -\frac{1-\alpha_{n,r}}{1+\alpha_{n,r}}\vec{\psi}_{n,-r}^h + \frac{2}{1+\alpha_{n,r}}\left(\mathbf{K}_n^r\vec{\psi}_n^h + \alpha_{n,r}\mathbf{H}_n^r\vec{\psi}_n^h\right). \tag{5.29}$$

Substituting Eq. 5.29 into Eq. 5.25 and reordering gives:

$$\left(\mathbf{D}+\sum_{r=x,y,z}\frac{2}{1+\alpha_{n,r}}\mathbf{P}_n^{+r}\left[\mathbf{K}_n^r+\alpha_{n,r}\mathbf{H}_n^r\right]\right)\vec{\psi}_n^h+\sigma_t\vec{\psi}_n^h=\frac{c}{4\pi}\sigma_t\vec{\phi}^h+\frac{\vec{q}}{4\pi}$$
$$+\quad\left(\mathbf{P}_n^{-x}+\frac{1-\alpha_{n,x}}{1+\alpha_{n,x}}\mathbf{P}_n^{+x}\right)\vec{\psi}_{n,W}^h+\left(\mathbf{P}_n^{-y}+\frac{1-\alpha_{n,y}}{1+\alpha_{n,y}}\mathbf{P}_n^{+y}\right)\vec{\psi}_{n,S}^h+\left(\mathbf{P}_n^{-z}+\frac{1-\alpha_{n,z}}{1+\alpha_{n,z}}\mathbf{P}_n^{+z}\right)\vec{\psi}_{n,B}^h. \tag{5.30}$$

Now, the expansion into powers of ϵ of the volume and face moments, Eqs. 5.22 and 5.23, respectively, are substituted into Eq. 5.30, and the resulting expression is separated by powers of ϵ. Note that for both AHOTN and HODD, $\alpha_{n,r}$ is asymptotically constant, i.e. in the limit $\epsilon\to 0$ it does not depend on σ_t. The $\mathcal{O}\left(\epsilon^{-1}\right)$ is given by:

$$\vec{\psi}_n^{h,[0]}=\frac{c}{4\pi}\vec{\phi}^{h,[0]}. \tag{5.31}$$

From Eq. 5.31 we can infer that the leading order solution is isotropic. Applying the quadrature operator to Eq. 5.31 we get:

$$\vec{\phi}^{h,[0]}(1-c)=0. \tag{5.32}$$

As discussed, for the cases of interest in this study $(1-c)>0$ and therefore we conclude:

$$\begin{aligned}\vec{\phi}^{h,[0]} &= 0\\ \vec{\psi}_n^{h,[0]} &= 0.\end{aligned} \tag{5.33}$$

Substituting the power expansions Eqs. 5.22 and 5.23 into Eq. 5.29 and retrieving the $\mathcal{O}(1)$ term yields:

$$\vec{\psi}_{n,+r}^{h,[0]}=-\frac{1-\alpha_{n,r}}{1+\alpha_{n,r}}\vec{\psi}_{n,-r}^{h,[0]}+\frac{2}{1+\alpha_{n,r}}\left(\mathbf{K}_n^r\vec{\psi}_n^{h,[0]}+\alpha_{n,r}\mathbf{H}_n^r\vec{\psi}_n^{h,[0]}\right)\quad\xrightarrow{\;\to 0\;}. \tag{5.34}$$

The $\mathcal{O}(1)$ term in the expansion of the outflow moments Eq. 5.23 is only zero in case $\lim_{\sigma_t\to\infty}\alpha_{n,r}=1$ which is satisfied by the AHOTN weights but not by the HODD weights. For HODD, the $\mathcal{O}(1)$ outflow moments are given by a constant times the inflow moments and therefore do not follow a $1/\sigma_t$ trend. This finding disqualifies the HODD method for application on optically thick meshes and explains the large errors of the HODD method for the optically thick test cases C_1(III) and C_1(IV) in Fig. 5.10.

Due to the described deficiencies of the HODD method, it will be discarded from the comparison from this point onwards: HODD performs very similar to AHOTN on fine meshes but is not suited for coarse meshes where its inherent, detrimental flaws lead to poor performance.

5.1.7 Influence of the Quadrature Rule on Accuracy and Efficiency

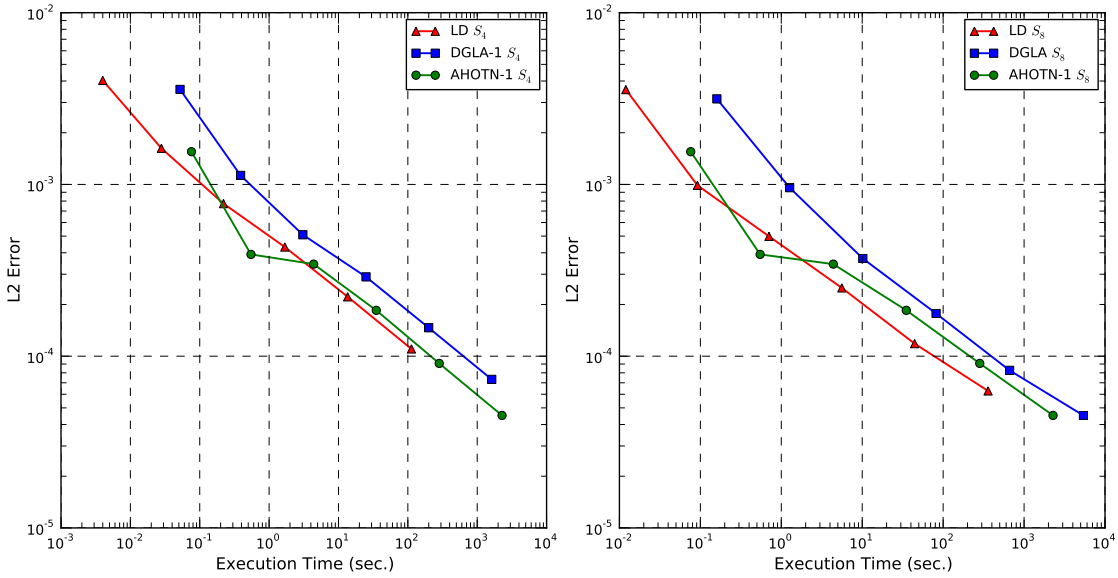


Figure 5.11: Discrete L_2 error versus execution time for the LD, DGLA-1, and AHOTN-1 method for the C_1(I) test case. The left subplot contains S_4 level symmetric results while the right subplot contains results obtained with the S_8 level symmetric quadrature.

Within this section, most discussed results are obtained using the S_4 level symmetric quadrature. All discussed errors are associated with the spatial discretization method only and do not comprise any angular discretization component. This is achieved by designing the MMS test suite for the S_N equations, i.e. incorporating a particular angular quadrature from the start. This subsection will demonstrate that changing the angular quadrature indeed does not change the general conclusions of this section. Therefore, one could swap another level-symmetric quadrature (or even another quadrature type) for the utilized S_4 quadrature and the results discussed within this section would still hold.

In Fig. 5.11 the errors obtained with the LD, DGLA-1 and AHOTN-1 methods for the C_1(I) test and S_4 (left subplot) and S_8 (right subplot) level symmetric quadrature are plotted versus the methods' execution time. The curves in the left and right subplot are strikingly similar supporting the expected independence of the angular quadrature on the results discussed within this section.

5.1.8 Influence of the Scattering Ratio on Accuracy and Efficiency

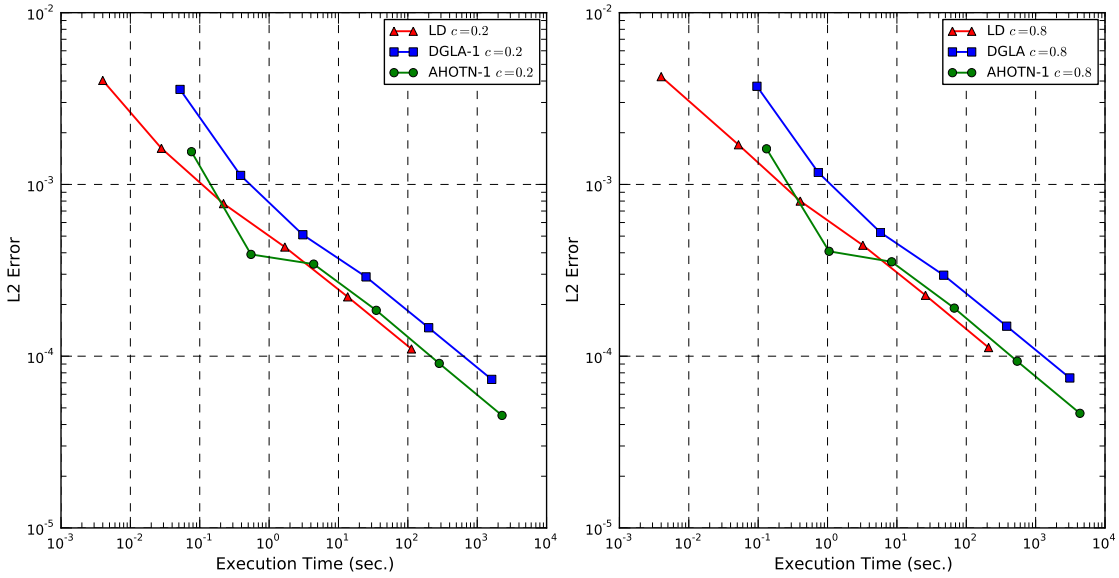


Figure 5.12: Discrete L_2 error versus execution time for the LD, DGLA-1, and AHOTN-1 methods for the C_1(I) (left subplot) and C_1(V) (right subplot) test case.

In Fig. 5.12 a comparison of the performance of LD, DGLA-1 and AHOTN-1 is presented for test cases C_1(I) and C_1(V), where the only difference between these two test cases is the scattering ratio of $c = 0.2$ and $c = 0.8$. The obtained error versus execution time curves are almost identical, indicating a negligible influence of the scattering ratio on the outcome of this study. This is partially attributed to the way in which the MMS problem is constructed. The scattering ratio enters only in the computation of the source and not in the flux shape. This is a shortcoming of the test problem because in general transport problems the scattering ratio influences the flux shape significantly, for example in problems in the diffusion limit.

5.1.9 Methods' Performance for C_1 Smoothness

A comparative study of the set of methods for the C_1(I) test case is presented in Figs. 5.13 through 5.17 for the discrete L_1, L_2 and L_∞ norms, the integral error norm, and the continuous L_2 error norm, respectively.

The discrete L_1 error results in Fig. 5.13 are very similar to the L_2 norm results in Fig. 5.14. This behavior is not only observed for the C_1(I) test case but throughout the whole

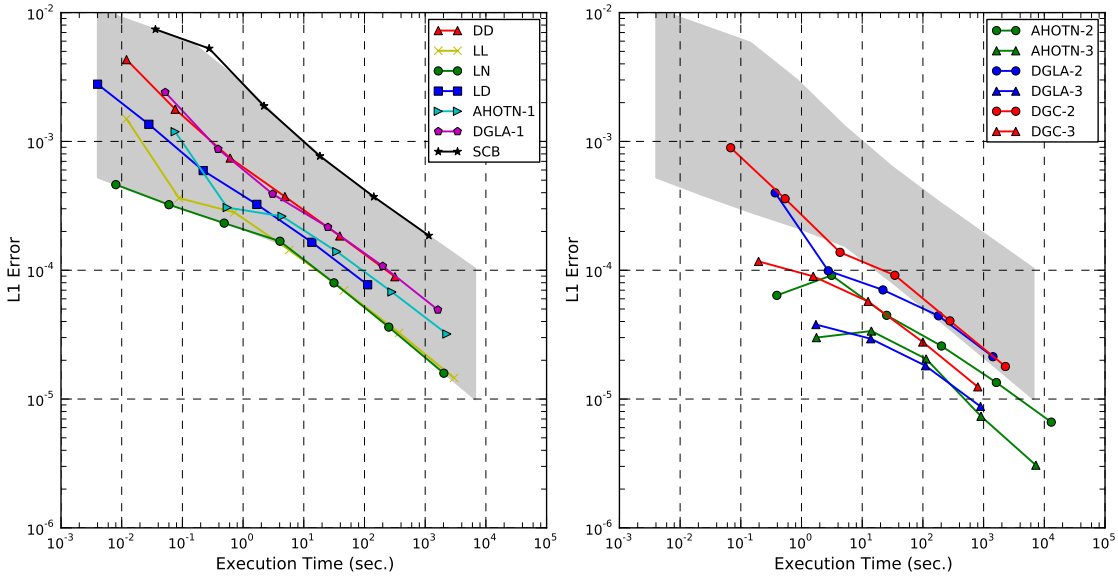


Figure 5.13: Discrete L_1 error versus execution time for various spatial discretization methods for orders for the C_1(I) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

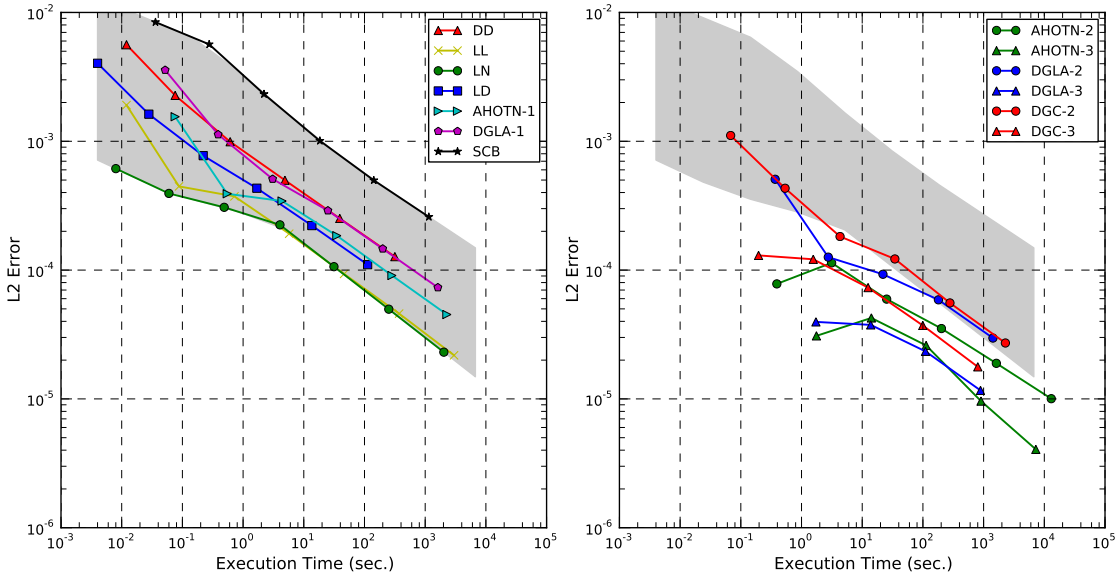


Figure 5.14: Discrete L_2 error versus execution time for various spatial discretization methods for orders for the C_1(I) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

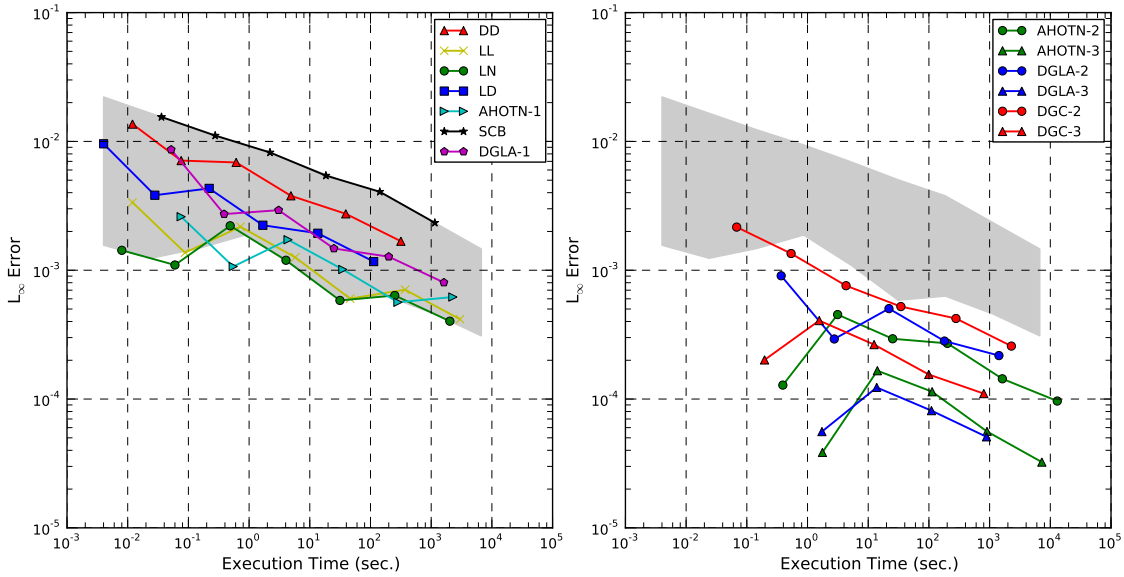


Figure 5.15: Discrete L_∞ error versus execution time for various spatial discretization methods for orders for the C_1(I) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

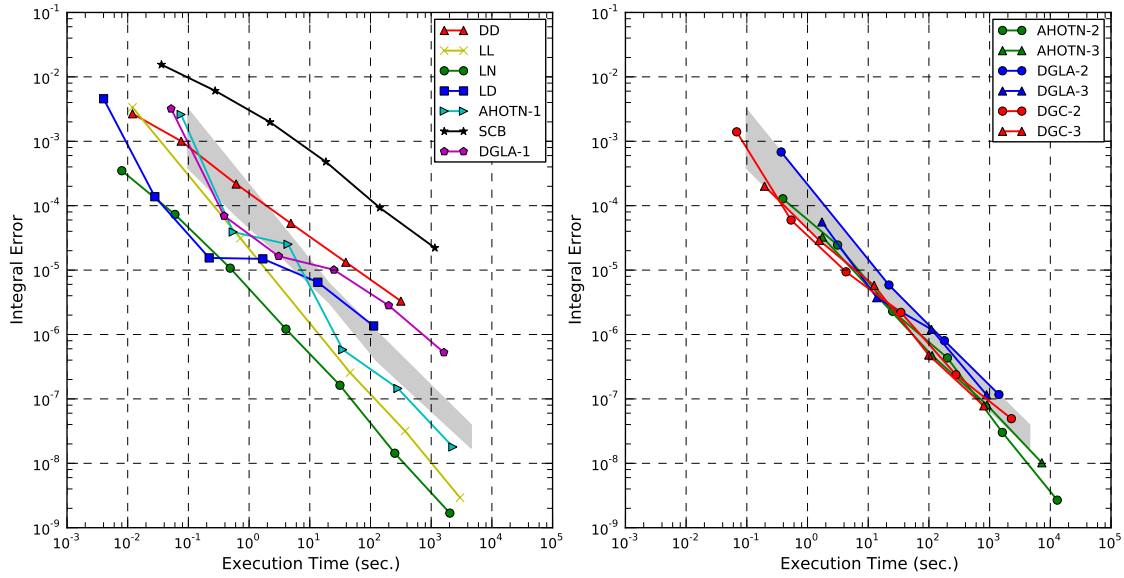


Figure 5.16: Integral error norm (computed for the left lower eighth subcube) versus execution time for various spatial discretization methods and orders for the C_1(I) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

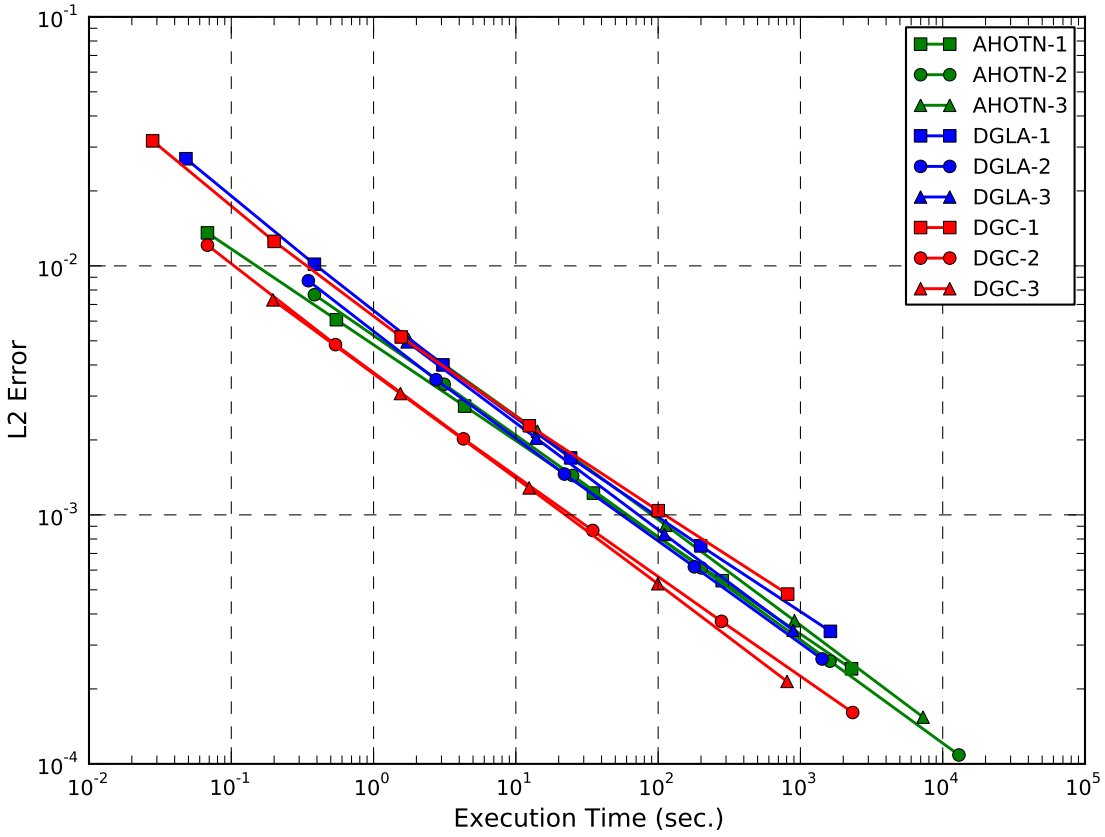


Figure 5.17: Continuous L_2 error versus execution time for various spatial discretization methods for orders for the C_1(I) test cases.

parameter space study. Therefore, the L_1 error norm will not be discussed in the remainder of this work and attention will be focused on the performance of the methods measured in the L_2 error norm.

For the performance in the discrete L_2 and L_∞ norms, the general rule holds that higher-order methods are more efficient. The gray shaded areas on the left and right in both Fig. 5.14 and 5.15 cover identical areas in their respective figures to facilitate comparison between the two subplots. Clearly, the higher-order versions of the discretization methods on the right-hand side are more efficient than the methods of order zero or one. In particular, the relatively cheap but inaccurate Simple-Corner balance method (SCB) performs worst, and the cheapest Diamond Difference method is the second least efficient method. Generally, more expensive, but accurate methods offer an advantage over cheaper methods. The particular problem with the SCB method is that accuracy is sacrificed for robustness in the thick diffusion limit: it is comparable in execution time to DGLA-1 on the same mesh but much less accurate. Throughout the whole parameter study the SCB method is found to be the worst performing method in the range of parameter space considered.

The most efficient methods are the the DGLA-3 and the AHOTN-3 methods, which perform (almost) identically well, followed at some distance by the DGC-3 methods. Reducing the order to two, we find that the AHOTN-2 method performs just slightly worse than the DGC-3 method, followed at some distance by the DGLA-2 and then the DGC-2 method. Comparing the two DGFEM families, LD performs better than DGLA-1, while the performance is similar for DGC-2 and DGLA-2 and finally for order three DGLA outperforms DGC. With increasing expansion order, DGLA's efficiency increases in comparison with the competing methods, in particular the DGC method and AHOTN, its main competitor for the best performing method. It should be pointed out here that the LD method's grind time is significantly shorter than DGLA grind time because of the streamlined kernel implementation.

Among the first order methods, the LL and LN methods are the best performers, while SCB, Diamond Difference, and DGLA-1 perform worst (in improving performance order). The reason for DGLA-1 to perform worse than one would expect is because it is one of the first order methods (besides AHOTN-1 and SCB) that uses a Lapack routine within each kernel solution, thus reducing the efficiency. In contrast to AHOTN-1, its accuracy is not offsetting its longer execution time. The LL and LN methods are comparatively efficient because their kernels are optimized while the obtained solutions are reasonably accurate on a given mesh.

The picture changes somewhat when looking at the integral error norm presented in Fig. 5.16. The higher-order methods' error versus execution time curves are clustered within a tight band with only minor differences occurring between the AHOTN, DGLA, and DGC methods of order two and three. Among the zero and first order methods are some that outperform the higher-order methods: the AHOTN-1 is slightly better than the higher-order methods, while

LL and LN significantly surpass the higher-order methods' efficiency. Even though it is not a general rule that the lower-order methods beat the higher-order methods when using integral error norms, it is noteworthy that the conclusions at least partially change with respect to $\mathcal{L}_p$ error norms in that the LL and LN appear as the most promising methods in the said case.

In order to elaborate on this shift of results, consider the ratio of the FOM Eq. 5.5 for a high and a low-order method:

$$d = \frac{\left(C_\epsilon \Delta t_g^{\lambda/3}\right)_H}{\left(C_\epsilon \Delta t_g^{\lambda/3}\right)_L}. \tag{5.35}$$

If $d > 1$ the lower-order method is more efficient, while for $d < 1$ the high-order method is more efficient. Reordering terms gives:

$$d = \frac{C_\epsilon^H}{C_\epsilon^L}\left(\frac{\Delta t_g^H}{\Delta t_g^L}\right)^{\lambda/3} = d_\epsilon d_t^{\lambda/3}. \tag{5.36}$$

Typically, $d_\epsilon < 1$ and $d_t > 1$ such that a balance must be struck between grind time and accuracy to achieve an efficient method. However, for increasing λ the influence of the execution time becomes more prominent, i.e. cheaper but less accurate methods gain an advantage over more expensive, but accurate methods. Two comments are in order here:

- That higher-order methods are less efficient when the permissible order λ is larger, seems counter-intuitive . However, it is a consequence of the fact that the order of convergence is identical for all methods of any order, and is stipulated by solution smoothness and error norm. In other branches of computational physics high-order methods are known to significantly outperform low-order methods when their potentially higher accuracy order can be utilized, but this advantage does not carry over to S_N problems due to the inherently low smoothness of the underlying exact solution.

- Even though the described comparison involves a high and a low-order method, it is in fact more general in that high-order method could be a more accurate and expensive method of the same order, e.g. we could compare LD and AHOTN-1 and would conclude that with larger λ, for example permissible by a different norm or solution smoothness, LD would gain an advantage.

Again, the underlying condition that the preceding discussion is valid, and in addition that the FOM for S_N methods, Eq. 5.6, makes sense is that λ is the same for all participating methods. If the underlying solution is smooth, the λ depends on the utilized methods' expansion orders and none of the stated results applies.

Finally, the continuous L_2 error norms plotted in Fig. 5.17 are distinguished from the discrete and integral error norms by several facts. First, the error versus execution time curves form straight lines in the log-log plots indicating that the asymptotic regime is reached immediately for the considered cell sizes. Second, the methods' curves are contained within a tight band, indicating that the methods' efficiencies when measured in the continuous L_2 norm are very similar and no method has a significant advantage over the others.

It should be pointed out that not all methods' continuous L_2 error norm was computed because the procedure by which it is computed requires the knowledge of the underlying finite element function space. The solution for a single mesh cell, encoded either in the expansion coefficients for DGLA and DGC or in the spatial Legendre polynomials moments for AHOTN, is used to reconstruct the flux shape within the said cell, and the continuous L_2 error norm can be accumulated using a quadrature rule. For some methods, for example the LL and LN methods, no such procedure was implemented and therefore the respective results are missing.

Even though the methods' performance in the continuous L_2 error norm do not exhibit a large spread, the DGC methods of orders two and three still are a bit more efficient than the remaining methods. The DGLA and AHOTN methods feature a similar efficiency across all orders, while the LD method is least efficient. It should be stressed that here too high-order methods (even though DGC instead of DGLA and AHOTN in this particular case) are most efficient, consistent with the results when measuring error in the discrete L_2 and L_∞ norm, Figs. 5.14 and 5.15, respectively.

However, among the higher-order methods, the DGC method is special in that it does not retain all flux moments but only those whose sum of all moment indices is less than the prescribed order. For $\Lambda = 3$ this would translates into 64 volume flux moments for DGLA versus 20 flux moments for DGC. The high-order flux moments are typically small compared to the low-order flux moments such that they constitute small corrections to an otherwise decent flux shape.

However, the higher the order of the flux moment, which shall be denoted by $\tilde{m} = m_x + m_y + m_z$, the harder it is to achieve its iterative convergence. For flux moments with $\tilde{m} > 2$, convergence would almost certainly stall at some $\epsilon >> \epsilon_{mach}$. Therefore, due to contamination with iterative convergence error, higher-order moments may not have the corrective effect on the flux shape that they would have otherwise. Note that given a fixed mesh size, DGLA is more accurate than DGC but it is not accurate enough to offset the additional cost of executing its kernel operation.

To summarize the results inferred from the C_1(I) test case regarding the accuracy and efficiency of the competing discretization methods:

- For error norms that permit only a small order of convergence, higher-order methods such as AHOTN-3, DGLA-3, or DGC-3 are most efficient.

- For the integral error norms, first order methods such as LN and LL outperform the higher-order methods.
- For orders $\Lambda > 1$, the DGLA family outperforms the DGC family for the discrete L_2 and L_∞ norms, both families perform about equally well for the integral error norms, and DGC outperforms DGLA when measuring error in the continuous L_2 error norm.
- The worst-performing method is the SCB method, which suffers from a relatively expensive grind time and inadequate accuracy. By design, it sacrifices accuracy for robustness in the thick diffusion limit, not tested in the MMS but separately in section 5.3, which explains its poor performance for this test problem.

Variation of the Domain Optical Thickness

In Figs. 5.18, 5.19, and 5.20 discrete L_2 error norms are plotted versus execution time corresponding to test problems C_1(II), C_1(III), and C_1(IV) with successively larger domain optical thickness. Test cases II-IV all feature a total cross section of $\sigma_t = 2$ and physical domain thicknesses of 4, 10, and 20 cm, respectively. As the solution of these problems is performed on meshes that are not in the asymptotic regime, asymptotic behavior, i.e. straight lines in the error versus execution time plots cannot be expected.

L_2 error norm results:

With respect to the base case, the C_1(I) problem depicted in Fig. 5.14, several trends can be inferred from the set of optically thick problems. The general conclusion that higher-order methods perform better than low-order methods ($\Lambda = 1$) remains only partially valid for test cases II-IV. While the AHOTN method of orders two and three still perform better than any of the lower-order methods, the same is not true for the DGLA and DGC methods of order two and three. The LL, LN, and AHOTN-1 methods are more efficient for short execution times, in particular for test cases C_1(III) and C_1(IV), and execution times $\Delta t < 10$ seconds.

Among the first order methods a clear separation emerges when increasing the domain optical thickness. On the one hand, the TMB methods: LL, LN, and AHOTN-1, and on the other hand, the remaining methods. The former's advantage in performance becomes more prominent as the optical thickness is increased. On the other end of the performance spectrum are the SCB and Diamond Difference methods. As already pointed out, SCB is not intended to be very accurate for the low scattering ratio cases considered here, and DD's properties make it unsuitable for solving problems on coarse meshes.

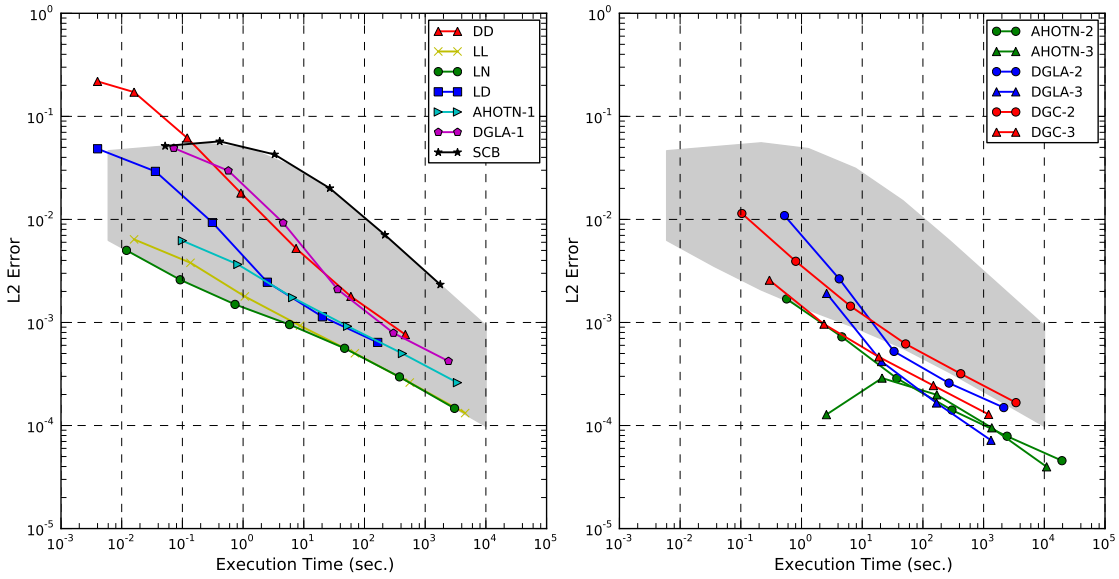


Figure 5.18: Discrete L_2 error versus execution time for various spatial discretization methods and orders for the C_1(II) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

Among the higher-order methods, the AHOTN method emerges as the most efficient method: While the advantage for longer execution times (corresponding to finer meshes) is not as pronounced as for short execution times, it still beats both the DGC and DGLA methods of orders two and three. In contrast to the C_1(I) test case, where AHOTN-3 had to share its top position with DGLA-3 for most efficient method, for test problems II-IV AHOTN-3's closest contestant is AHOTN-2.

The two DGFEM families exhibit errors that are much larger (up to 10 times) than AHOTN-2,3 for execution times less than 10 seconds, while the differences diminish as execution times increase, i.e. the mesh is refined. It has been found in [53] that AHOTN is more accurate than DGLA given the same mesh spacing. The findings presented here demonstrate that under certain conditions it is also more efficient.

Comparing the DGLA with the DGC methods' results unveils the unexpected finding that DGC-2,3, particularly for test cases III and IV in Figs. 5.19 and 5.20, respectively, performs better than DGLA-2,3 for coarse meshes. For the C_1(II) test problem this trend is also visible, but it is more pronounced in cases III and IV. With mesh refinement, DGLA catches up and eventually becomes more efficient than DGC. It should be mentioned that LD outperforms DGLA-1 in all presented test problems mainly because of its streamlined implementation.

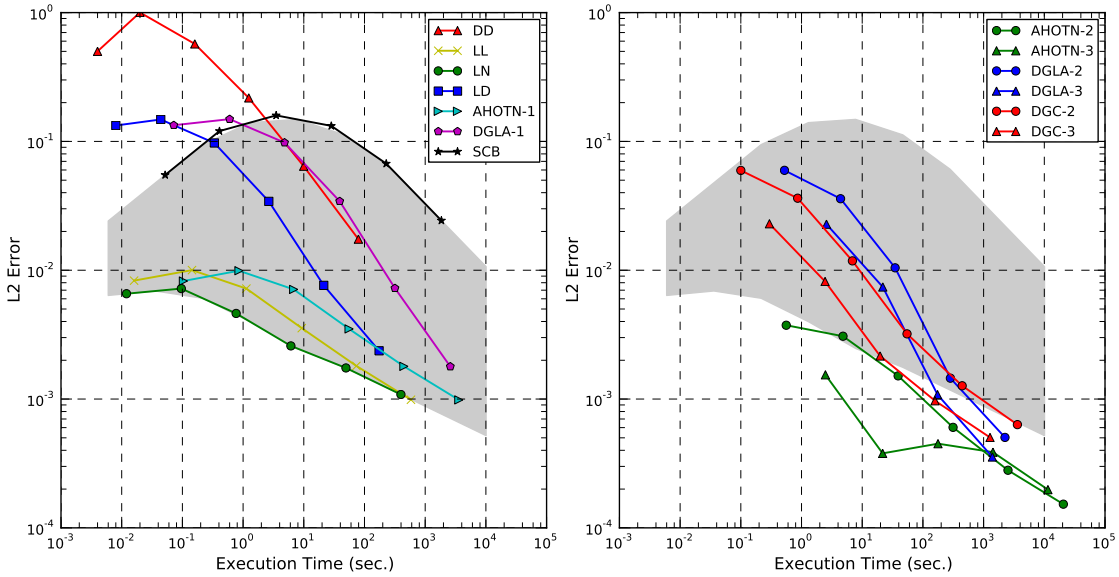


Figure 5.19: Discrete L_2 error versus execution time for various spatial discretization methods and orders for the C_1(III) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

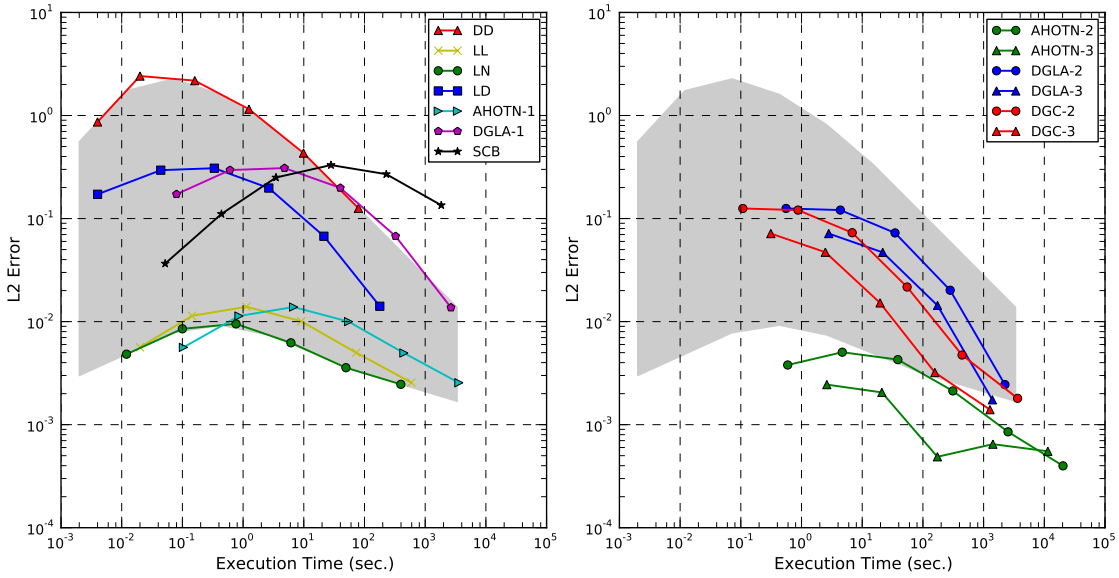


Figure 5.20: Discrete L_2 error versus execution time for various spatial discretization methods and orders for the C_1(IV) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

In contrast to the C_1(I) test case, Fig. 5.14, where the three best performing methods all feature $\Lambda = 3$ in the order AHOTN, DGLA, DGC followed by the $\Lambda = 2$ methods in exactly the same order, the results for optically thick domains tend to be grouped by method. Best performance AHOTN-3 followed by AHOTN-2, followed by the first order TMB methods and finally the DGC methods of orders two and three followed by the DGLA method of orders two and three. It appears that the discretization method rather than the expansion order plays the decisive role when solutions on coarse meshes are desired. However, fixing the method, a higher-order still trumps lower-order methods.

With increasing the domain optical thickness starting from Fig. 5.18 to 5.20, several of the methods develop maxima in the error versus execution time curves that move from the right edge of the figures (coarse meshes) towards the middle (finer) meshes as the domain optical thickness is increased. This phenomenon is particularly pronounced for first order methods but higher-order methods show onsets of this behavior as well for test case C_1(IV) in Fig. 5.20.

The reason for the maximas' occurrence is cancellation of errors, but not in the random, volatile process described before that is enhanced by non-smooth solutions and oscillations of the solution in their vicinity. The angular flux in the MMS test problem is either monotonically decreasing equivalent to e^{-x}, or monotonically increasing equivalent to $1 - e^{-x}$. The greater the total cross section, the faster either case approaches its saturation value (0 and 1 in the example). Therefore, the larger the total cross section and the physical domain thickness are, the larger the volume fraction in which the angular flux is essentially flat; see Fig. 5.21. Any discretization method will obtain the exact solution for cases where the exact solution is essentially flat.

A necessary condition for the occurrence of broad maxima is an error norm that allows cancellation of error like the discrete L_2 error norm. The process that causes the particular shape of the error curves is illustrated in Fig. 5.21: The angular flux equivalent to $\psi = 2(1 - e^{-\sigma_t x})$ is plotted for three different cases. In the first case either the mesh size h is very large with respect to the physical domain thickness or the total cross section is very large. Therefore the flux assumes 99% of its asymptotic value within the first mesh cell and the majority of the volume is occupied by a flat flux region.

The distance at which the flux assumes 99 % of its asymptotic value is denoted by t and the ratio $\gamma = t/h$ is a measure of the regime that the solution is in.

Case: $\gamma << 1$: Most mesh cells are fully within the flat flux region, but the boundary cells feature a small volume fraction characterized by a step gradient. Any discretization method of any value will obtain an accurate solution for cells fully in the flat flux region, regardless in which norm the error is measured. However, the shape of the solution in the boundary cells is hard to approximate; certainly, if the error is measured in a pointwise norm as for example the

continuous L_2 norm, the error would be large. However, the discrete L_2 norm uses only the average values. Because of the small volume fraction of the steep region, it does not significantly influence the average value within the cell.

To demonstrate this, consider the average over the mesh cell:

$$\bar{\psi} = \frac{1}{h}\int_0^h dx 2\left(1 - e^{-\sigma_t x}\right) = \frac{2}{h}\left(h + \left[\frac{1}{\sigma_t}e^{-\sigma_t x}\right]_0^h\right) = 2 + \frac{2}{h\sigma_t}\left[e^{-\sigma_t h} - 1\right]. \tag{5.37}$$

Thus for $\sigma_t \to \infty \Rightarrow \bar{\psi} \to 2$ and therefore the steep region does not influence the cell-average. Numerical methods will yield inadequate resolution for the steep region, but the error will cancel over the much larger flat-flux sub-volume of the cell. Therefore, for $\gamma << 1$, numerical methods may obtain cell-wise errors that tend to zero on coarse meshes([44]).

Case: $\gamma \approx 1$: When the mesh size h is comparable to t then the slope and curvature within the boundary cell are hard to approximate using a numerical method. Furthermore, the computed average within the cell is not easy to infer as in the case $\gamma << 1$, i.e. you need to get slope and curvature right to obtain a reasonable approximation of the average. Therefore, the computed average flux will feature a large error. The $\gamma \approx 1$ range corresponds to the broad maximum in the error versus execution time/mesh refinement curves.

Case: $\gamma >> 1$: The exact angular flux looks almost linear on the fine mesh, i.e. slope and curvature are well resolved and can be well approximated by the numerical method. The case $\gamma >> 1$ corresponds to a data point close to or within the asymptotic regime.

Going through the cases $\gamma << 1$, $\gamma \approx 1$ and $\gamma >> 1$ the solutions first start with a small error, then the error increases until it reaches its peak at $\gamma \approx 1$, and finally decreases as the slope and curvature of the exact solution is well approximated.

Within the preceding discussion the necessary condition for the occurrence of the maxima in the error versus execution time curves was an error norm that allows smearing of results over the extent of one mesh cell. The discrete L_2 norm allows that because it only takes into account averages which are computed before absolute values are applied. As a corollary the said phenomenon must vanish if an error norm is used that does not allow cancellation of error/smearing over the extent of a mesh cell. In Fig. 5.22 continuous and discrete L_2 error norms are plotted versus DGLA-1's execution time for test problems C_1(I) to C_1(IV). Maxima in these curves occur for the discrete L_2 norm results but not for the continuous L_2 norm which is consistent with the explanation given above.

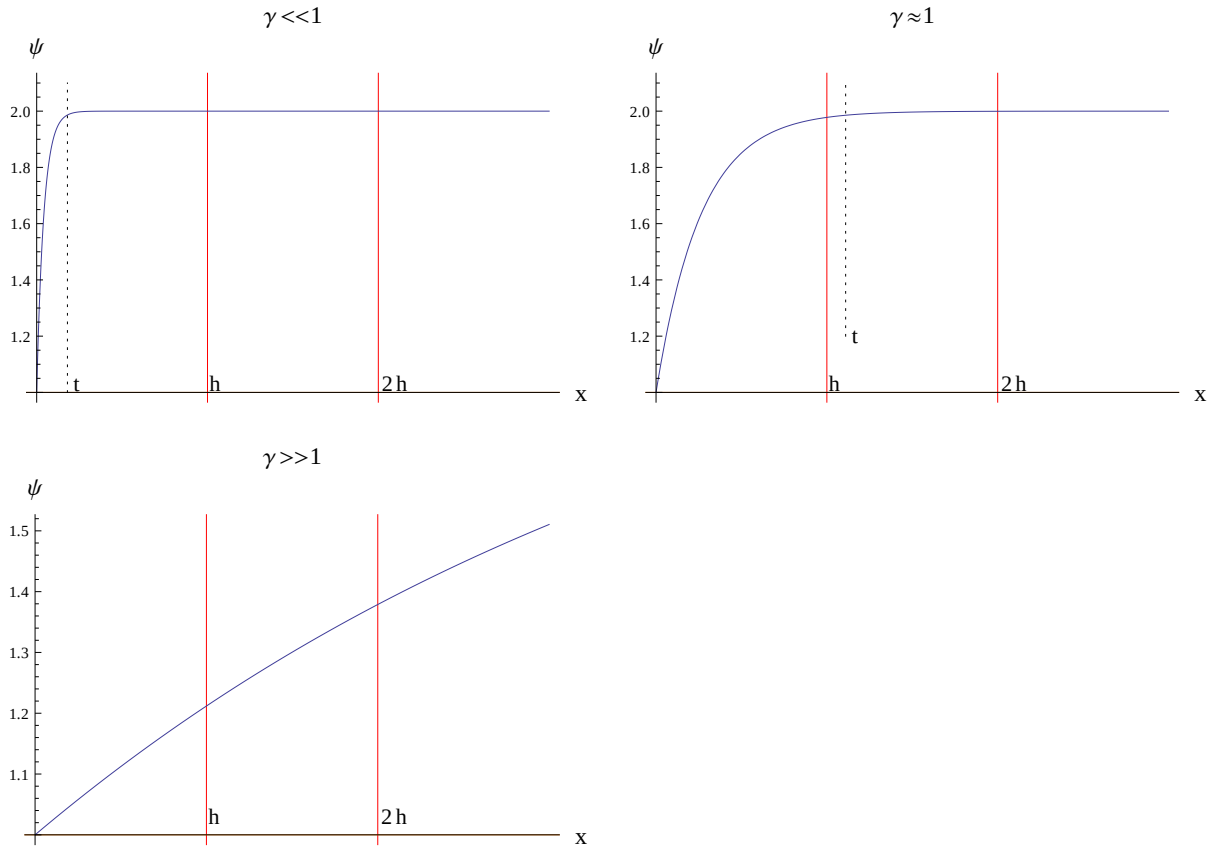


Figure 5.21: Illustration of the process that leads to broad maxima in the error versus execution time/mesh spacing curves. Cases $\gamma << 1$, $\gamma \approx 1$ and $\gamma >> 1$ are snapshots of scenarios corresponding to an under-resolved solution with small error, poorly resolved solution with large discretization error (maximum of error) and well resolved solution with small error, respectively.

Integral error norm results:

For the C_1(I) test case, the main conclusion for errors measured in the integral error norm was that first order TMB methods: LL, LN and AHOTN-1 are the best performers with LN outperforming any other method. In Figs. 5.23 to 5.25, results are presented for test problems C_1(II)-C1(IV), respectively, with the discretization error measured in the integral error norm.

The general findings from the C_1(I) test case do not change when the optical domain thickness is increased. The LN method still emerges as the most efficient method followed by the LL method. The higher-order methods are less efficient than LL and LN, but more efficient than the remainder of the first order methods.

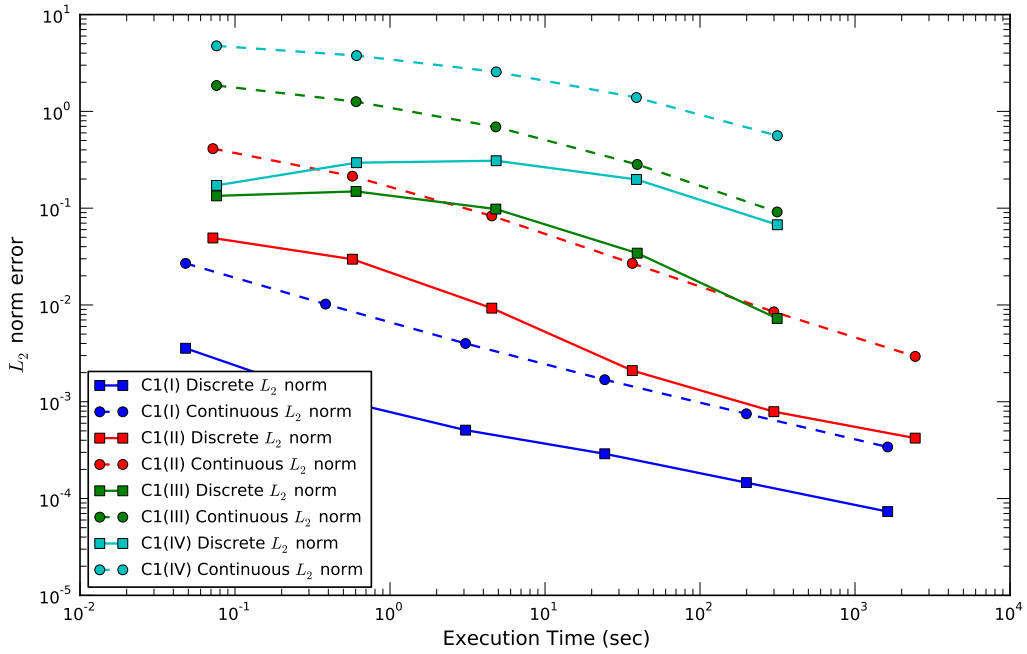


Figure 5.22: Discrete and Continuous L_2 error norm results for test cases C_1(I) through C_1(IV) obtained using the DGLA-1 method.

It is noteworthy that among the high-order methods, things change significantly when increasing the domain optical thickness. For test case C_1(I), Fig. 5.16 , AHOTN-2,3, DGLA-2,3 and DGC-2,3's error curves were confined to a tight band, while for cases C_1(II) to C_1(IV), deviation among the high-order methods is observed. For case C_1(II), Fig. 5.23, the DGC methods perform best followed by the DGLA and finally AHOTN. The described best-to-worst performer order changes for the C_1(III) and C_1(IV) test cases to DGC/AHOTN/DGLA and AHOTN/DGC/DGLA, respectively, i.e. the AHOTN method improves in its rank as the domain optical thickness is increased. This is consistent with the observation that TMB methods perform better than straight polynomial DGFEM methods on coarse meshes.

Some of the curves in Figs. 5.23 to 5.25 exhibit a peculiar L shape, initial steep slopes that quickly change to a smaller slopes. We shall elaborate on the reason for this particular shape. As an example we use the DGC-2 results from test case C_1(II). We suppose that the error is made up of two components in the following fashion:

$$\|\epsilon\|_2 = C_1 T^{\lambda_1} + C_2 T^{\lambda_2}, \tag{5.38}$$

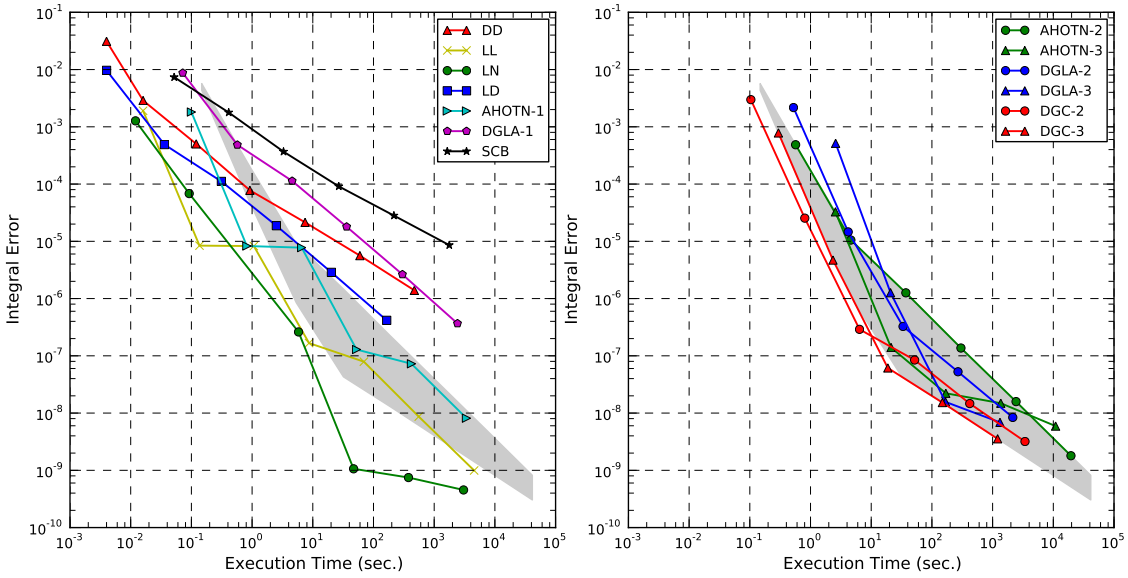


Figure 5.23: Integral error norm versus execution time for various spatial discretization methods and orders for the C_1(II) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

where T is the execution time and λ_1, λ_2, C_1, and C_2 are constants. Without loss of generality we assume $|\lambda_1| > |\lambda_2|$ such that in the limit $T \to 0$ the first term will dominate and in the limit $T \to \infty$ the second term will dominate.

This will now be demonstrated based on the DGC-2 results obtained for the C_1(II) test case. Indexing the times and errors from lowest to highest execution time and denoting them by: t_j and $\|\epsilon\|_2^{(j)}$ the coefficients in Eq. 5.38 are computed using the expressions in Table 5.4. The computed values are used for plotting Eq. 5.38 together with the DGC-2 data in Fig. 5.26. The model reproduces the observed L shape very well and therefore we conclude that the described peculiar shape is caused by an error that is composed of two parts, one that dominates for short execution times/coarse meshes and another that dominates for longer execution times/finer meshes.

In summary the following findings are most important within this subsection:

- Discrete L_2 error: When increasing the domain optical thickness the TMB methods are more efficient when compared to the other methods. However, higher-order DGFEM methods, by and large, outperform low-order TMB methods: LL, LN, and AHOTN-1.

- Discrete L_2 error: The most efficient method for optically thick test cases is the AHOTN-3 method.

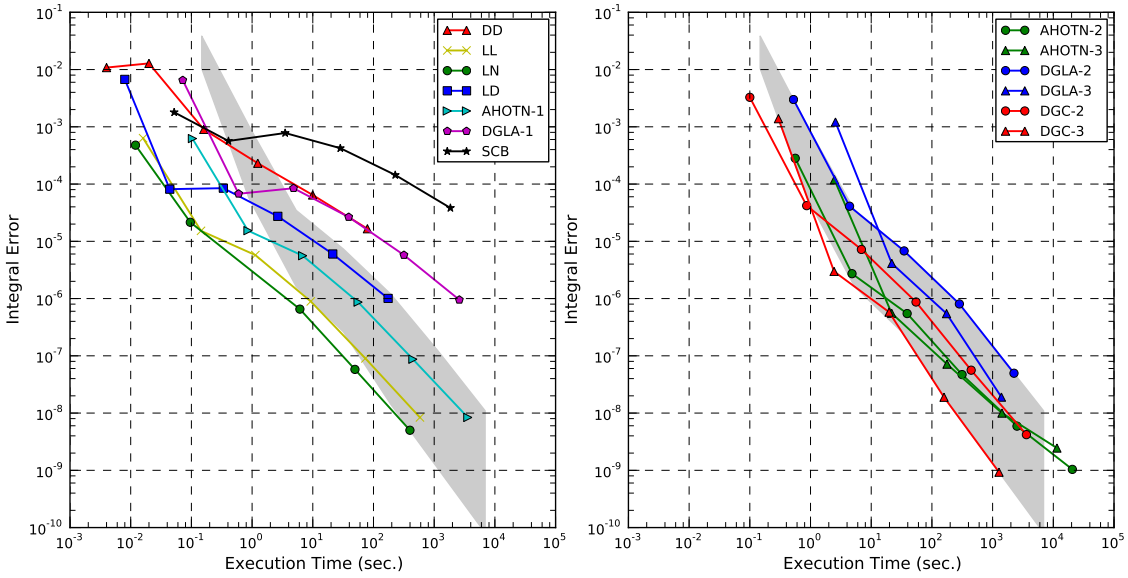


Figure 5.24: Integral error norm versus execution time for various spatial discretization methods and orders for the C_1(III) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

- Integral error: LN followed by LL remain the most efficient methods.
- The discrete L_2 error norm allows for broad maxima in the error versus execution time curves. These maxima are related to a thin boundary layer featuring large variation in the angular flux and large flat-flux regions in the domain's interior. Continuous norms that do not allow cancellation of error do not exhibit maxima in the error vs. execution time curves.

Variation of Aspect Ratio

Within this subsection, the domain aspect ratio is varied starting from $X = Y = Z = 1$ (C_1(I)) to $X = 1.4, Y = 1.0, Z = 0.8$ (C_1(VI)) and finally $X = 2.0, Y = 0.2, Z = 0.2$ (Table 5.1). The domain aspect ratio translates directly into the cells' aspect ratio because a uniform mesh with an identical number of intervals per dimension is used. The total cross section and optical thickness remain constant at $\sigma_t = 1$ and $c = 0.2$. The results for the C_1(VI) and C_1(VII) test case measured in the discrete L_2 error norms are plotted in Figs. 5.27 and 5.28.

The curves obtained for the C_1(VI) test case are very similar to their C_1(I) counterparts depicted in Fig. 5.14 because the aspect ratio of each cell has not change significantly. However, it is important to observe that going to a non-unity aspect ratio does not significantly alter

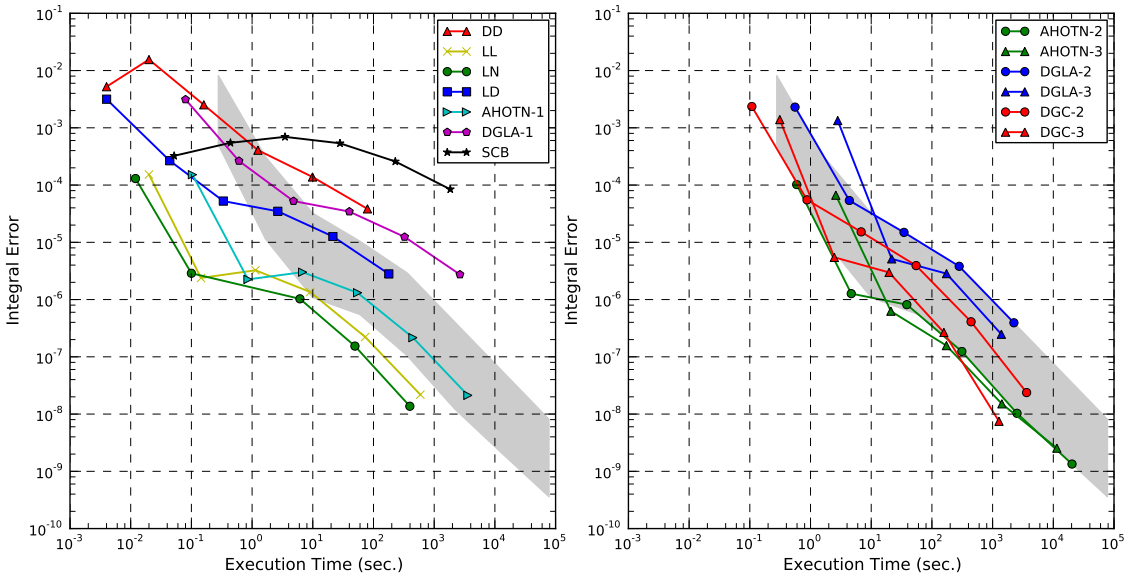


Figure 5.25: Integral error versus execution time for various spatial discretization methods and orders for the C_1(IV) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

the best-to-worst-performer ordering of the methods, i.e. for example one method is much better at unity aspect ratio than another method. For the lower-order methods, the results are virtually identical, but for the higher-order methods slight changes in ordering are observed. In particular, the DGC-3 methods moves very close to the AHOTN-3 and DGLA-3 curves such that these three methods perform virtually equally well.

When moving to test case C_1(VII) more significant changes in the efficiency ranking of the methods occur. First, among the low-order methods the LD method's performance beats the AHOTN-1 method by a significant margin while both methods performed about equally well for the C_1(I) test case. For higher expansion orders, the AHOTN-2,3 methods lose ground and are clearly outperformed by both DGFEM families. Particularly interesting, however, is that the DGC family now has an edge over the DGLA family: DGC-3 outperforms DGLA-3 slightly and DGC-2 outperforms DGLA-2 significantly. For the same mesh DGLA is still more accurate, but its advantage in accuracy cannot offset its longer execution time.

AHOTN and DGLA are at a disadvantage compared to DGC because payoff from retaining all mixed order flux moments (or expansion terms) decreases when the cell aspect ratio is highly skewed. In order to demonstrate the reduction of importance of higher-order flux moments, a cell-wise measure of importance of higher-order moments κ of how much the additional DGLA

Table 5.4: Computation of coefficients in Eq. 5.38 and the resulting values using DGC-2 data.

	λ	C
Set 1	$\lambda_1 = \frac{\log \frac{\|\epsilon\|_2^{(3)}}{\|\epsilon\|_2^{(2)}}}{\log \frac{t_3}{t_2}} = -2.15$	$C_1 = \frac{\|\epsilon\|_2^{(3)}}{t_3}$ =1.6-5
Set 2	$\lambda_2 = \frac{\log \frac{\|\epsilon\|_2^{(7)}}{\|\epsilon\|_2^{(6)}}}{\log \frac{t_7}{t_6}} = -0.72$	$C_2 = \frac{\|\epsilon\|_2^{(7)}}{t_7}$ =1.2-6

expansion terms change the solution is defined:

$$\kappa = \frac{\int_{\Omega_{\vec{i}}} dV \left(p_\Lambda - p'_\Lambda\right)^2}{\int_{\Omega_{\vec{i}}} dV p_\Lambda^2}, \tag{5.39}$$

where p_Λ is the reconstructed angular flux within a single cell obtained with DGLA-Λ, and p'_Λ is the corresponding flux shape obtained with DGC. Using orthogonality of the Legendre polynomial basis functions, κ can be expressed as:

$$\kappa = \frac{\sum\limits_{m_x+my+m_z>\Lambda} \frac{\psi_{\vec{m}}^{h,\vec{i}}}{2\vec{m}+1}}{\sum\limits_{\vec{m}\leq\Lambda} \frac{\psi_{\vec{m}}^{h,\vec{i}}}{2\vec{m}+1}}. \tag{5.40}$$

In order to test the conjecture, DGLA-1,2,3 equations are solved for a single cell with uniform unit inflow on all inflow boundary conditions and no distributed source. The aspect ratio δ is varied, and the physical cell thickness is computed as:

$$\begin{aligned} \Delta x_i &= \delta^{-1/3} \\ \Delta y_j &= \delta^{-1/3} \\ \Delta z_k &= \delta^{2/3}. \end{aligned} \tag{5.41}$$

These physical domain thicknesses ensure that the volume is always unity.

The importance of higher-order moments κ is plotted versus δ for $\Lambda = 1, 2, 3$ in Fig. 5.29. With decreasing optical aspect ratio, κ quickly drops according to a power law:

$$\kappa = C\delta^p, \; p \approx 3, \tag{5.42}$$

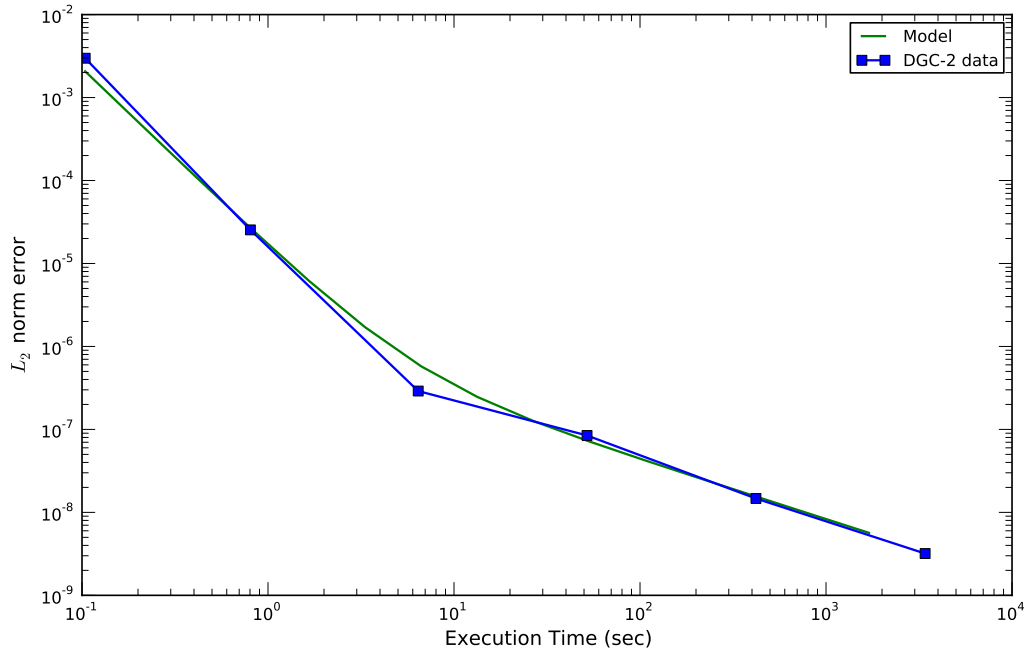


Figure 5.26: Comparison of the model Eq. 5.38 and the DGC-2 error versus execution time curve for test case C_1(II).

where C is some constant. The provided evidence supports the conjecture that for skewed aspect ratios, mixed flux moments carry less importance than for optical aspect ratios close to unity.

Summary of C_1 Results

In summary, the decision of which discretization method is the most efficient for C_1 problems depends on the error norm used. When using a discrete L_p error norm, higher-order methods beat lower-order methods with AHOTN-3 and DGLA-3 typically leading the field. However, LN or LL should be favored when integral quantities are desired. Some methods are never competitive such as the SCB and DD methods. The former finding is not surprising because SCB is designed to be robust in the thick diffusive limit which is achieved by sacrificing accuracy in non-diffusive regimes. Diamond Difference features the shortest grind time, but it's not accurate enough even on optically resolved meshes. For optically thick problems TMB methods comparatively perform better than the remaining straight polynomial methods. In contrast, when cells feature skewed aspect ratios DGC's performance improves drastically and should be

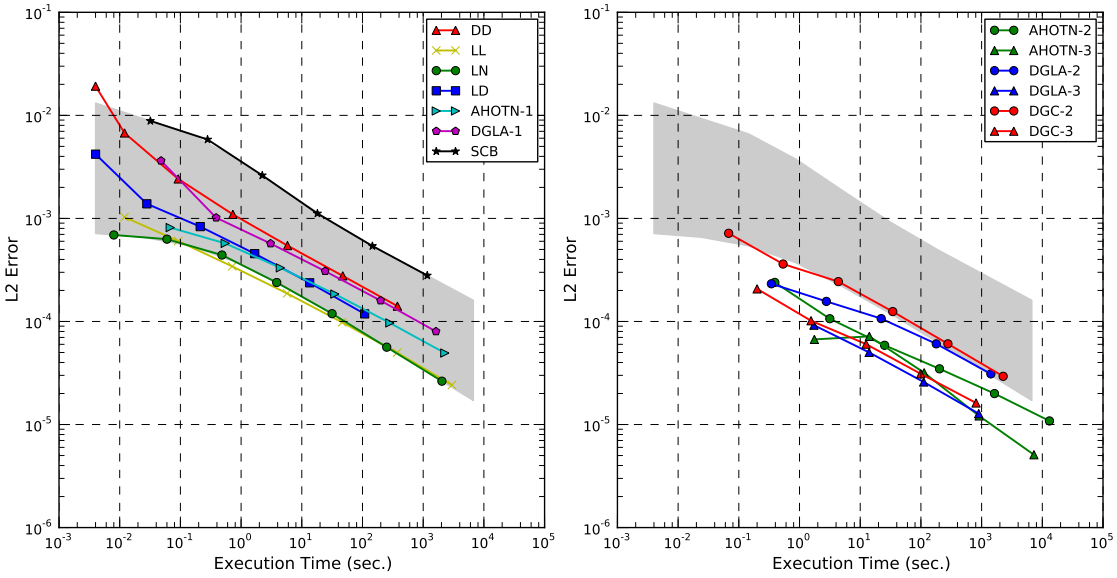


Figure 5.27: Discrete L_2 error norm versus execution time for various spatial discretization methods and orders for the C_1(VI) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

the method of choice.

5.1.10 Methods' Performance for C_0 Smoothness

Problems with discontinuous solutions referred to in this work as C_0 problems are very difficult to solve with standard numerical methods. Oscillations of the solution in the vicinity of the discontinuities leads to spurious pre-asymptotic behavior, very low observed convergence rates with mesh refinement and lack of cell-wise convergence prohibit obtaining high-fidelity solutions on any reasonable uniform mesh. Approaches to circumvent this problem can be found in Refs. [1] and [64]. Both references capitalize on adaptive mesh refinement to target cells featuring large errors, which are typically regions of low smoothness, to isolate them from the remainder of the mesh cells. Thus, resources are efficiently used to contain the influence of non-smoothness. Further [1] introduces the Singular-Characteristic Tracking algorithm for two-dimensional Cartesian meshes. This algorithm was extended to three-dimensional Cartesian meshes within this work forming the SCT-Step method.

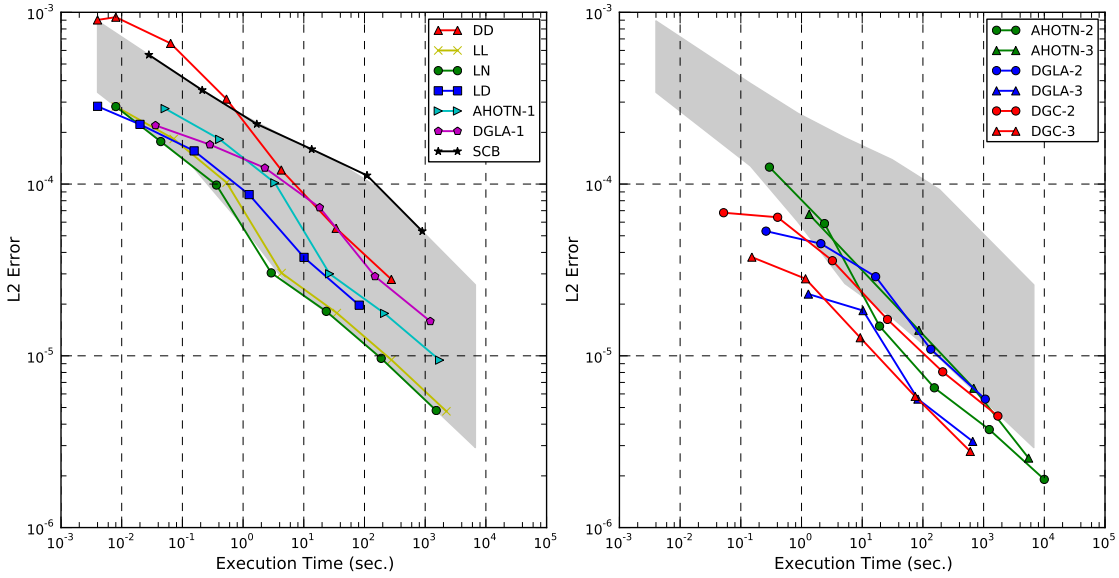


Figure 5.28: Discrete L_2 error norm versus execution time for various spatial discretization methods and orders for the C_1(VII) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

Discrete L_∞ error norm:

First, if the error is measured in the discrete L_∞ norm, then regular methods are non-convergent. The error increases with mesh refinement because the average flux of cells intersected by the SC/SPs does not converge to the true value. Therefore, the only resort currently available is to use the SCT algorithm. Cell-wise convergence may be an essential requirement in large-scale shielding problems where cells are physically large[1] and therefore many cells are intersected by the SC/SPs. It is essential to limit the error on the computed average flux within these cells, but further mesh refinement may be infeasible leading to a computation whose fidelity is not sufficient to enable decision-making on it.

Discrete L_2 error norm results:

If the error is measured in the discrete L_2 norm, as depicted in Figs. 5.30 to 5.32 for the C_0(I), C_0(II) and C_0(VII) test problems, solutions on coarse meshes are unreliable in their behavior with mesh refinement. The error may initially be large or small and subsequently may

[1]Cells representing air can be optically thin even though their physical dimensions are large due to the small density of air.

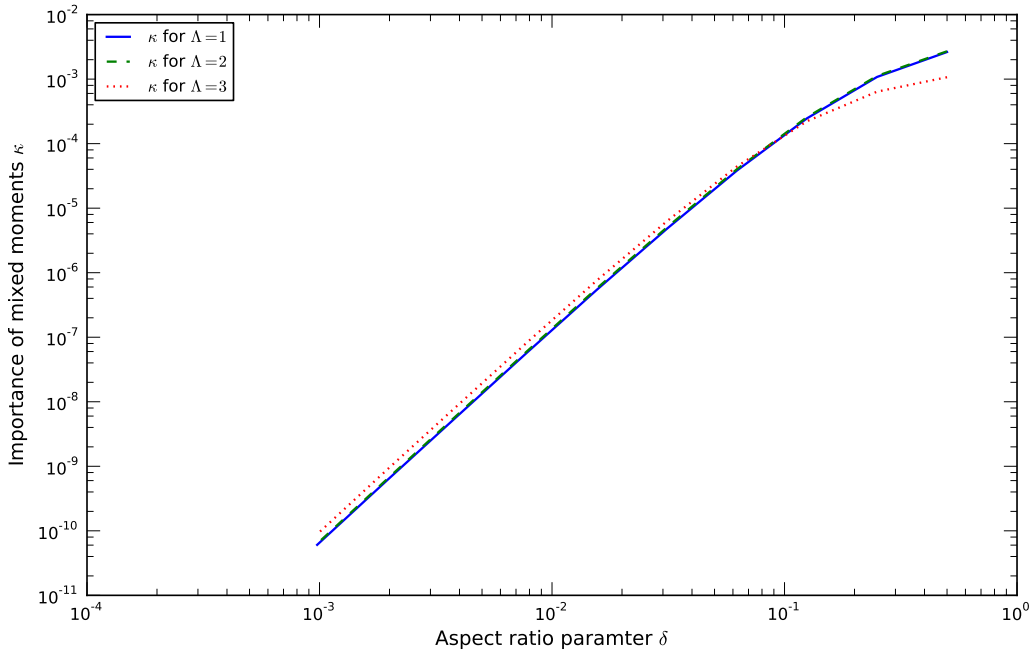


Figure 5.29: Plot of the importance of mixed flux expansion terms κ versus aspect ratio parameter δ for expansion orders $\Lambda = 1, 2, 3$.

increase or decrease with mesh refinement. Therefore, the results within this regime are not trustworthy. A sufficient number of mesh refinement steps is necessary to get into a regime where the error is decreasing monotonically albeit at a very small rate. Execution times corresponding to sufficient mesh refinement to obtain (for the most part) a monotonically decreasing error trend are indicated by a red line in plots 5.30 to 5.32. The necessary execution times range from two up to 200 seconds for most of the cases. For the high-order methods in test case C_0(II), the regime in which a decreasing error is observed is not reached within the admissible number of mesh refinement steps.

For all test problems with the exception of C_0(I) and high-order methods the SCT-Step method surpasses the ordinary discretization schemes' efficiency before reaching the red line. Therefore, the SCT-Step method is the most efficient discretization method when the error is measured in the discrete L_2 error norm. It should be pointed out that the SCT-Step method is far from an efficient method in itself because it utilizes the step discretization, which is grossly inaccurate due to its first order accuracy[2].

[2]In fact, the reason why it is so inaccurate is the inability to capture any slope in its solution. Linear methods perfectly reproduce large gradients as long as the curvature of the underlying solution is small.

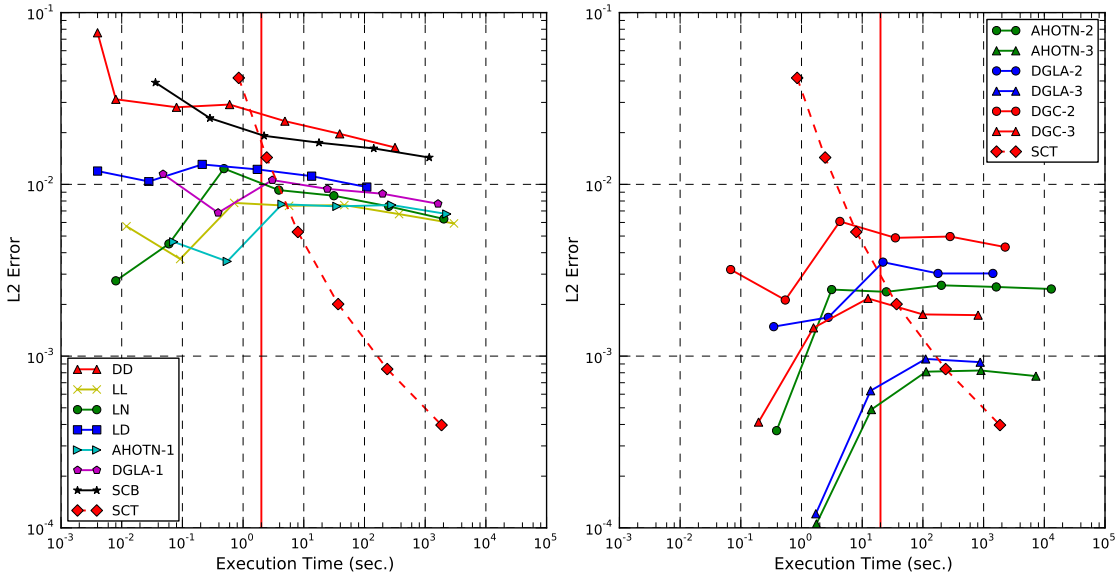


Figure 5.30: Discrete L_2 error norm versus execution time for various spatial discretization methods and orders for the C_0(I) test case. Red line indicates necessary level of mesh refinement (translated into execution time) from where on standard methods' results are trustworthy.

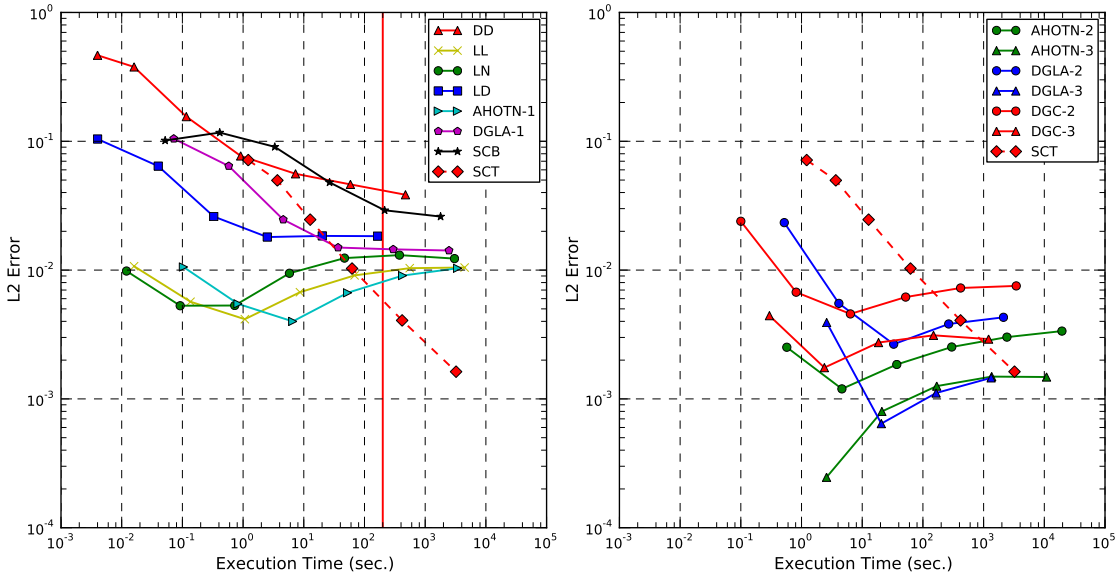


Figure 5.31: Discrete L_2 error norm versus execution time for various spatial discretization methods and orders for the C_0(II) test case. Red line indicates necessary level of mesh refinement (translated into execution time) from where on standard methods' results are trustworthy.

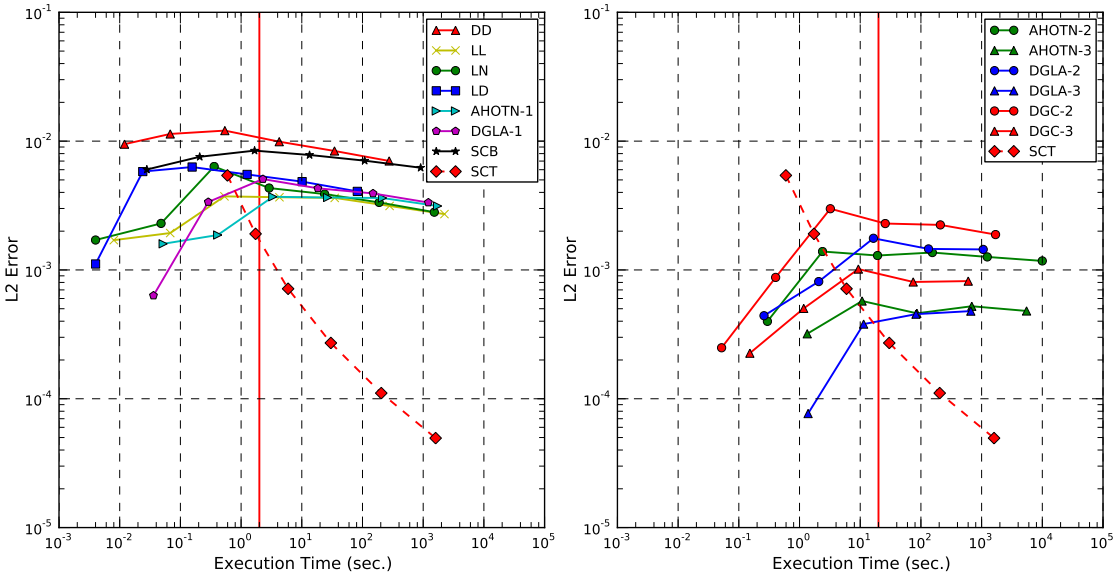


Figure 5.32: Discrete L_2 error norm versus execution time for various spatial discretization methods and orders for the C_1(VII) test case. Red line indicates necessary level of mesh refinement (translated into execution time) from where on standard methods' results are trustworthy.

The obvious deficiency of standard methods for the solution of C_0 problems logically leads to the creation of the SCT-Step method. However, its inherent deficiency is the poor quality of the utilized step method. Only because of this deficiency are the standard methods even competitive with the SCT-Step method. Combining a higher-order method with the SCT algorithm would combine the advantages of the SCT algorithm with a high quality discretization method. In addition, a high-order SCT method would have the potential to allow for spectral convergence, i.e. it restores the method's inherent convergence order.

It must be stressed here that the SCT method does not change the mesh and does not require a different mesh for each angular direction, but operates locally on the mesh cell of interest, separates it into the illumination segments, solves the pertinent equations on the subvolumes, and collapses them before finishing the solution in the said cell.

An algorithm for a higher-order SCT scheme should contain the following ingredients:

- An unstructured grid solver, e.g. DGFEM on tetrahedrons or for general polyhedra.
- A subroutine that divides the more complicated polyhedra illuminated by one boundary face into simple bodies like tetrahedra.
- A prolongation operator that computes the subcell (tetrahedra) source moments given

cell-wise source expansion without loss of accuracy order.

- A restriction operator that computes the cell-wide angular flux moment from the subcell solutions without loss of accuracy order.

Integral error norm results:

Integral error norm results are presented for the C_0(I) test problem in Fig. 5.33. These results are presented here as an example to demonstrate that integral error norm results for the C_0 test problems are remarkably similar to their C_1 counterparts. Therefore, the conclusions drawn earlier in the corresponding C_1 section carry over to the C_0 problems.

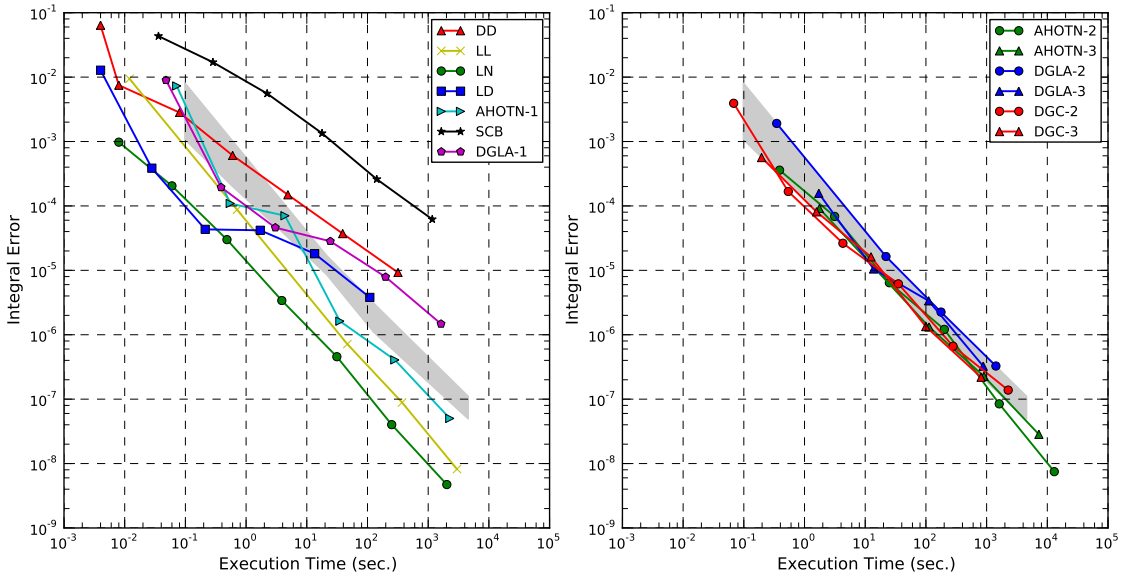


Figure 5.33: Integral error norm versus execution time for various spatial discretization methods and orders for the C_0(I) test case. The shaded area is identical in both plots to facilitate comparison between the two plots.

Summary of C_0 Results

With the exception of the integral error norm, C_0 results differ significantly from C_1 results. Cell-wise convergence (L_∞ norm) is attained only when using the SCT-Step method. For the discrete L_2 error norm, sufficient mesh refinement is necessary to obtain trustworthy results; however, for the presented test cases the SCT-Step method has already surpassed the efficiency

of the standard method at this point: the SCT-Step curve is below the other method's curve (smaller error for the same execution time). Within the set of considered methods, the SCT-Step method is the most suitable one for C_0 problems when cell-wise solutions are desired. However, its inherent flaw, the poorly performing step approximation, could be amended by combining the SCT with a higher-order method. Integral error norm performances do not differ from the C_1 test cases and therefore the same conclusions hold.

5.1.11 Summary of MMS Test Suite Results

Deciding which method is the most efficient for solving a given problem depends on the choice of the error norm and the smoothness of the exact solution. When accuracy is desired in the discrete L_p norms and the solution is at least C_1, then higher-order methods outperform low-order methods. For optically thick domains featuring moderate aspect ratios AHOTN-3 is the best performer, while for skewed aspect ratios the DGC-3 method emerges as the most efficient method. If the solution does not feature a differentiable flux (C_0) then the SCT-Step method is the best performer among the set of considered methods; in the author's opinion using the SCT algorithm in conjunction with a high-order method would create a superior method. Finally, if the error is measured in the integral error norm, the LL and LN methods are the best performers. By using the FOM defined in 5.5 we rationalized this behavior that is caused by a larger rate of convergence that is, however, identical for all methods.

5.2 Positivity: Lathrop's Test Problem

Negative fluxes are an undesirable feature of radiation transport solutions because they can inhibit convergence, pose problems for multiphysics coupling, and undermine user's confidence in the solution's validity. Negative solutions can occur for angular face fluxes, angular volume fluxes, scalar face fluxes, and scalar volume fluxes. In addition, by negative flux we mean that the average flux over a cell's volume or face is negative, or that the flux shape somewhere on the face or within the volume is negative.

Angular face fluxes are most prone to becoming negative in the sense that a cell whose source and inflow face fluxes are non-negative cannot have negative volume fluxes unless at least one outflow face flux is negative. Further, angular volume fluxes are more prone to becoming negative than scalar fluxes because at least one angular flux must be negative for the scalar flux to be negative. However, the scalar flux can still be positive even if the angular flux is not for all discrete ordinates. Within this work, we will concentrate on angular face fluxes and scalar volume fluxes. Angular face fluxes are the most sensitive indicator that the flux is not strictly positive, while volumetric scalar fluxes are used to compute reaction rates and are therefore an important quantities to be considered as the ultimate reason for performing a transport calculation, e.g. when multiphysics coupling of the radiation transport solver is desired.

The average flux is less prone to becoming negative than the flux shape that is perfectly acceptable to be negative. A negative flux shape will only cause a negative average flux if the negative contributions are not offset by positive contributions on a face or within a volume. Within this work, only average fluxes are considered for two reasons:

1. Flux shapes are hard to reconstruct and examine for general methods because the correct function space that ought to be used for the flux reconstruction is unknown for some methods.

2. For practical purposes, the flux average is the most important quantity and only in special applications may the flux shape be required to be positive.

Given non-negative inflow fluxes and distributed source, negative fluxes are, with a notable exception, a local phenomenon that can be explained by considering the solution process in a single cell. For the transport problem to be physically meaningful, the boundary conditions and distributed source are non-negative, so at least for the boundary cells, these conditions are true if the subject boundary condition is on an upstream face of the boundary cell. Once negative fluxes occur for the first cell, the conditions mentioned before are not valid anymore and cells that would under normal circumstances not produce negative fluxes might do so because the inflow and/or sources are negative.

This work will therefore look more closely at a single cell that satisfies the said conditions and examine under which circumstances the outgoing face fluxes can become negative. Within this work, single cell coupling coefficients are defined and used to explain the behavior of several methods that produce negative average fluxes in subsection 5.2.1. In subsection 5.2.2, Lathrop's test case is used to compare method's susceptibility to negative fluxes using the metrics defined in section 3.4.

5.2.1 Single Cell Coupling Coefficients

Following Eq. 4.22 and extending slightly to account for methods other than DGFEM, the within-cell set of equations is written as:

$$\mathbf{T}\vec{\psi}_n^{h,\vec{i}} = \vec{b}_S^{h,\vec{i}} + \sum_{F\in\mathcal{E}^I} \mathbf{A}_F \vec{\psi}_{n,F}^{h,\vec{i}}, \tag{5.43}$$

where matrices $\mathbf{A}_F$ are of size $(\Lambda+1)^3 \times (\Lambda+1)^2$, $\mathbf{T}$ is of size $(\Lambda+1)^3 \times (\Lambda+1)^3$, and $\vec{b}_S^{h,\vec{i}}$ contains the contributions from the distributed and scattering sources. The solution to Eq. 5.43 is then given by:

$$\vec{\psi}_n^{h,\vec{i}} = \mathbf{T}^{-1}\vec{b}_S^{h,\vec{i}} + \sum_{F\in\mathcal{E}^I} \mathbf{T}^{-1}\mathbf{A}_F \vec{\psi}_{n,F}^{h,\vec{i}}. \tag{5.44}$$

Within this section, the quantity of interest are the face fluxes. The face fluxes can be obtained from the volume flux moments/expansion coefficients by upstreaming (DGFEM) or substitution into the WDD equations (LL, LN, AHOTN, HODD). In general, the face flux moments/expansion coefficients are linear combinations of the volume flux moments/expansion coefficients such that the face fluxes can in general be written as:

$$\vec{\psi}_{n,F'}^{h,\vec{i}} = \mathbf{B}_{F'}\mathbf{T}^{-1}\vec{b}_S^{h,\vec{i}} + \sum_{F\in\mathcal{E}^I} \mathbf{B}_{F'}\mathbf{T}^{-1}\mathbf{A}_F \vec{\psi}_{n,F}^{h,\vec{i}}, \tag{5.45}$$

where $\mathbf{B}_{F'}$ is a matrix relating the volume flux moments/expansion coefficients to the face flux moments/expansion coefficients. Defining $\mathbf{C}_S = \mathbf{B}_{F'}\mathbf{T}^{-1}$ and $\mathbf{C}_{F,F'} = \mathbf{B}_{F'}\mathbf{T}^{-1}\mathbf{A}_F$, Eq. 5.45 yields:

$$\vec{\psi}_{n,F'}^{h,\vec{i}} = \mathbf{C}_S\vec{b}_S^{h,\vec{i}} + \sum_{F\in\mathcal{E}^I} \mathbf{C}_{F,F'} \vec{\psi}_{n,F}^{h,\vec{i}}. \tag{5.46}$$

The outflow face angular fluxes are a linear combination of the inflow face angular fluxes and the source. The single-cell coupling coefficients are the linear combination coefficients encoded in the matrices $\mathbf{C}_S$ and $\mathbf{C}_{F,F'}$. The coupling coefficient of outflow flux moment $\vec{m}^{F'}$ and inflow

flux moment $\vec{m}^F$ or source moment $\vec{m}$ can be retrieved by:

$$\begin{aligned} \text{Inflow/Outflow coupling} \quad &: \quad c_{F,F'}^{\vec{m}^F \to \vec{m}^{F'}} = \frac{\partial \psi_{n,\vec{m}^{F'}}^{h,\vec{i}}}{\partial \psi_{n,\vec{m}^F}^{h,\vec{i}}} \\ \text{Source coupling} \quad &: \quad c_S^{\vec{m} \to \vec{m}^{F'}} = \frac{\partial \psi_{n,\vec{m}^{F'}}^{h,\vec{i}}}{\partial S_{\vec{m}}^{h,\vec{i}}} \end{aligned} \tag{5.47}$$

In this subsection the focus shall be on the outflow face-averaged angular flux coupling coefficients denoted by:

$$\begin{aligned} \text{Inflow/Outflow coupling} \quad &: \quad \bar{c}_{F,F'}^{\vec{m}^F} = \frac{\partial \bar{\psi}_{n,F'}^{h,\vec{i}}}{\partial \psi_{n,\vec{m}^F}^{h,\vec{i}}} \\ \text{Source coupling} \quad &: \quad \bar{c}_{S,F}^{\vec{m}} = \frac{\partial \bar{\psi}_{n,F'}^{h,\vec{i}}}{\partial S_{\vec{m}}^{h,\vec{i}}}. \end{aligned} \tag{5.48}$$

The outflow face-averaged angular fluxes can now be computed exactly using the formula:

$$\bar{\psi}_{n,F'}^{h,\vec{i}} = \sum_{\vec{m}} \bar{c}_{S,F}^{\vec{m}} S_{\vec{m}}^{h,\vec{i}} + \sum_{F \in \mathcal{E}^I} \sum_{\vec{m}^F} \bar{c}_{F,F'}^{\vec{m}^F} \psi_{n,\vec{m}^F}^{h,\vec{i}}. \tag{5.49}$$

The signs and relative magnitudes of the coupling coefficients determine how prone a discretization method is towards developing negative face-averaged outflow fluxes. If the inflow and the source are flat, i.e. $\psi_{n,\vec{m}^F}^{h,\vec{i}} = 0$ except for $\vec{m}^F = (0,0)^T$ and $S_{\vec{m}}^{h,\vec{i}} = 0$ except for $\vec{m} = (0,0,0)^T$, then the particular outflow face flux average is related to the inflow and source averages by just four coupling coefficients.

In general, the distributed source and inflow face fluxes are not flat, but we shall restrict our discussion here to a "model" cell where they are. This would amount to ignoring the effect of higher-order face and volume source moments on the positivity of the averaged outflow fluxes. The reason for not considering the higher-order moments can be stated as follows:

- Clarity: The large number of coupling coefficients would not allow for a clear discussion if the influence of higher-order inflow/source moments was considered.
- Importance: The average inflow face fluxes are assumed to have the highest influence on the average outflow face fluxes.
- Positive definiteness: Averages are required to be non-negative, but higher-order moments can be either positive or negative[3]. Thus, a coupling coefficient relating inflow and outflow

[3]They might represent slopes for example.

averages is problematic if and only if it is negative. For higher-order coupling coefficients this simple rule does not hold. Rather, magnitude and sign of all coefficients must be considered in conjunction with all other coupling coefficients and with signs of higher-order flux moments. This substantially complicates the analysis.

Average-Average Single Cell Coupling Coefficients

Adopting the simplified analysis outlined above, the outflow average flux can be written as:

$$\text{Flat fluxes/source: } \bar{\psi}^{h,\vec{i}}_{n,F'} = \bar{c}_{S,F}\bar{S}^{h,\vec{i}} + \sum_{F\in\mathcal{E}^I} \bar{c}_{F,F'}\bar{\psi}^{h,\vec{i}}_{n}. \tag{5.50}$$

The exact solution of the mono-directional transport problem given flat inflow fluxes and distributed source can be obtained analytically. As a matter of fact, this analytical solution is implemented in the *Step Characteristic* method, i.e. the SC method obtains the exact answer to problems featuring flat inflow fluxes and distributed source.

It is noteworthy that not all inflow faces are coupled to all outflow faces, i.e. some $\bar{c}_{F,F'}$ are exactly zero. However, with the exception of characteristic discretization methods, all discretization methods couple face-averaged fluxes incorrectly; they feature a non-zero coupling coefficient where the exact coupling coefficient should be zero. The question that will be answered within this section is which of the coupling coefficients become negative when the optical cell thickness is increased raising the potential for negative cell-averaged outflow fluxes. In particular, the question examined here is whether correctly coupled but negative coupling coefficients, or negative incorrectly coupled coupling coefficients cause negative outflow fluxes.

The coupling coefficients, $\bar{c}_{F,F'}$ and $\bar{c}_{S,F}$, are obtained by using the finite difference representation of Eq. 5.48:

$$\begin{aligned}
\text{Inflow/Outflow coupling} \quad &: \quad \bar{c}_{F,F'} = \frac{\bar{\psi}^{h,\vec{i}}_{n,F'}\left(\left[\bar{\psi}^{h,\vec{i}}_{n,F}\right]_1\right) - \bar{\psi}^{h,\vec{i}}_{n,F'}\left(\left[\bar{\psi}^{h,\vec{i}}_{n,F}\right]_2\right)}{\left[\bar{\psi}^{h,\vec{i}}_{n,F}\right]_1 - \left[\bar{\psi}^{h,\vec{i}}_{n,F}\right]_2} \\
\text{Source coupling} \quad &: \quad \bar{c}_{S,F} = \frac{\bar{\psi}^{h,\vec{i}}_{n,F'}\left(\left[\bar{S}^{h,\vec{i}}\right]_1\right) - \bar{\psi}^{h,\vec{i}}_{n,F'}\left(\left[\bar{S}^{h,\vec{i}}\right]_2\right)}{\left[\bar{S}^{h,\vec{i}}\right]_1 - \left[\bar{S}^{h,\vec{i}}\right]_2},
\end{aligned} \tag{5.51}$$

which does not incur any approximation because of the linear relationship between inflow/source and outflow fluxes. In Eq. 5.51, the square brackets indicate a specific choice of the inflow face-averaged flux or the volume-averaged source. For obtaining all ten coupling coefficients, the within-cell equations have to be solved four times. For each of these within-cell solves, a unit inflow averaged flux on a single face or a unit source average is set, and all other inflow/source

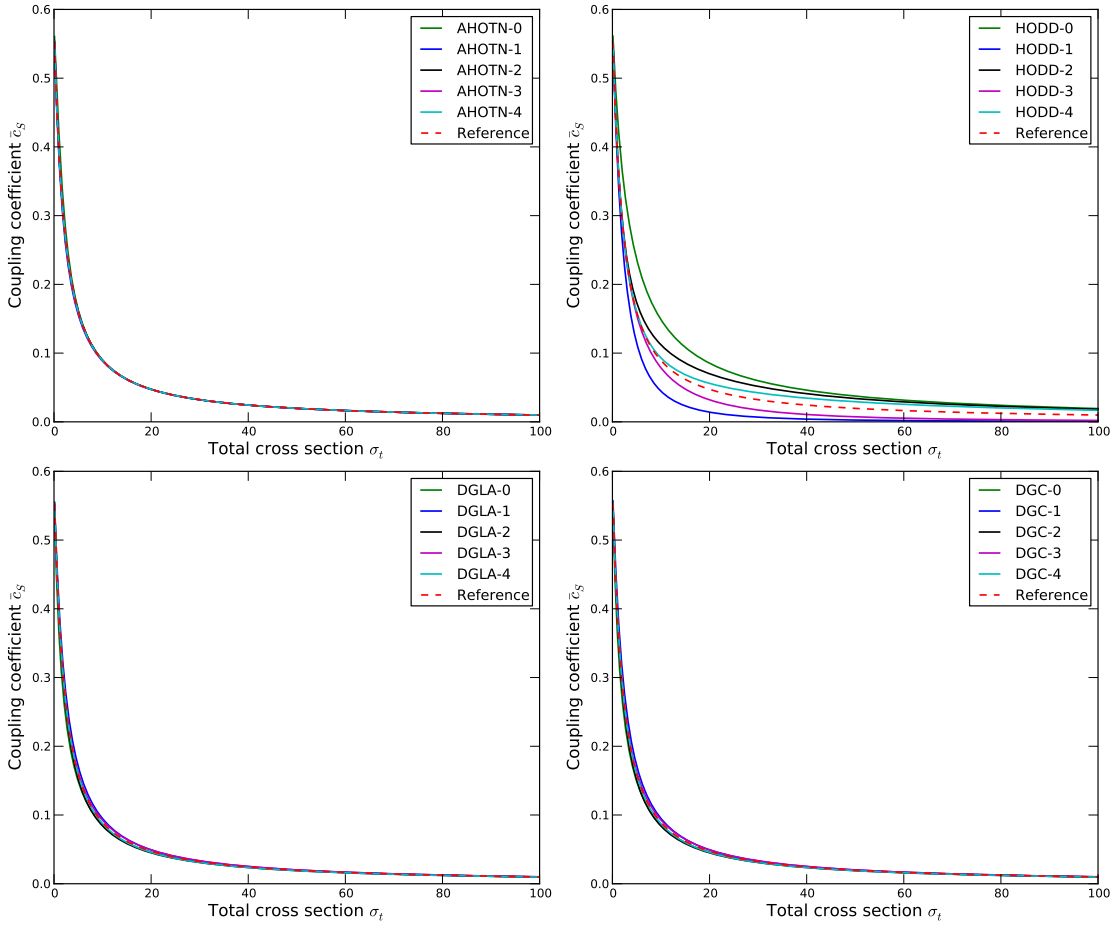


Figure 5.34: Coupling coefficients $\bar{c}_{S,F}$ versus σ_t for test case I (unity aspect ratio) and AHOTN, HODD, DGLA, and DGC of orders $\Lambda = 0, ...4$.

values are set to zero. Then, the resulting outflow flux averages are equal to the desired coupling coefficients.

The Source to Outflow Coupling Coefficients - Numerical Results

Two test cases are considered which vary in the cell's optical aspect ratio: Test case I has an optical aspect ratio of unity $\Delta x = \Delta y = \Delta z$ and $\hat{\Omega} = \left(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}\right)^T$, while test case II features $\Delta x = 2, \Delta y = 1, \Delta z = 1/2$ and the same $\hat{\Omega}$. For both cases the total cross section is varied from $\sigma_t = 0.01$ up to $\sigma_t = 100.0$. For test case I all $\bar{c}_{S,F}$ are identical, while for test

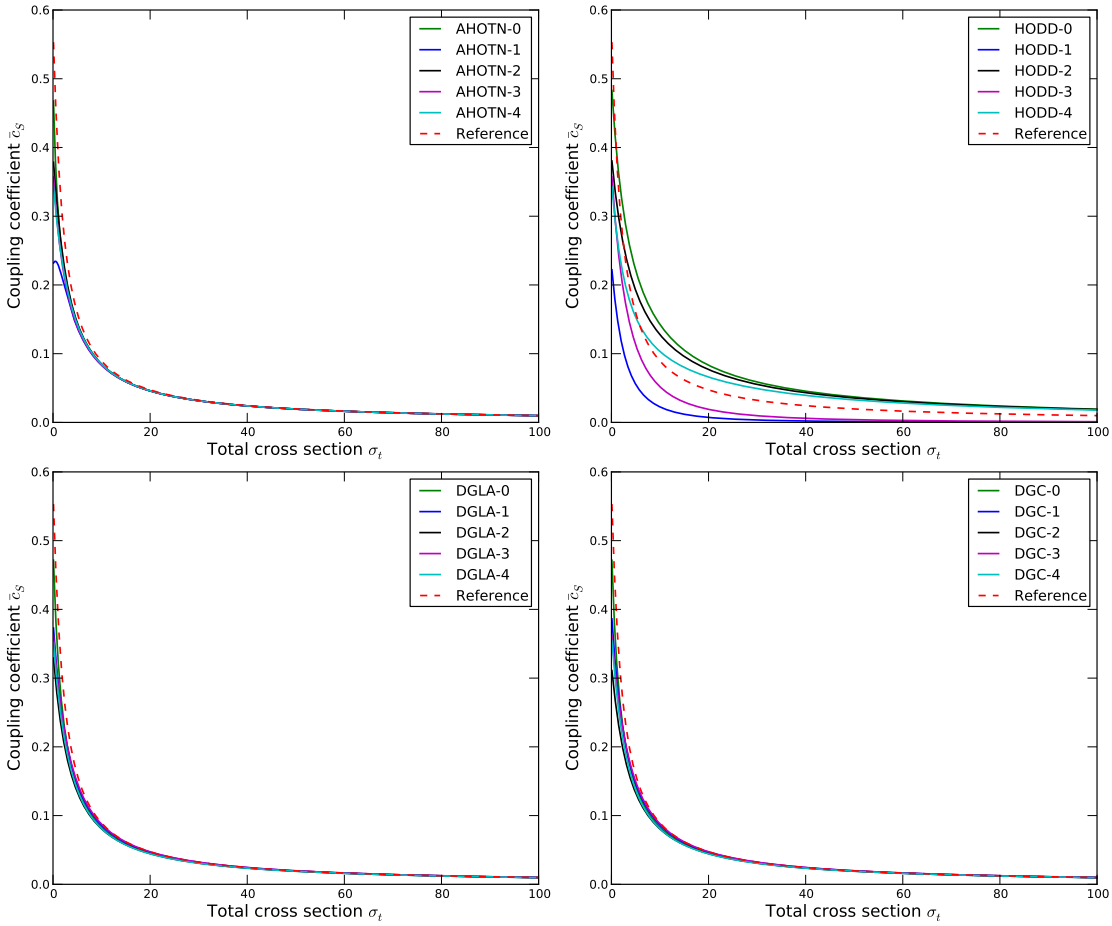


Figure 5.35: Coupling coefficients $\bar{c}_{S,E}$ (East outflow face) versus σ_t for test case II (non-unity aspect ratio) and AHOTN, HODD, DGLA, and DGC of orders $\Lambda = 0, ...4$.

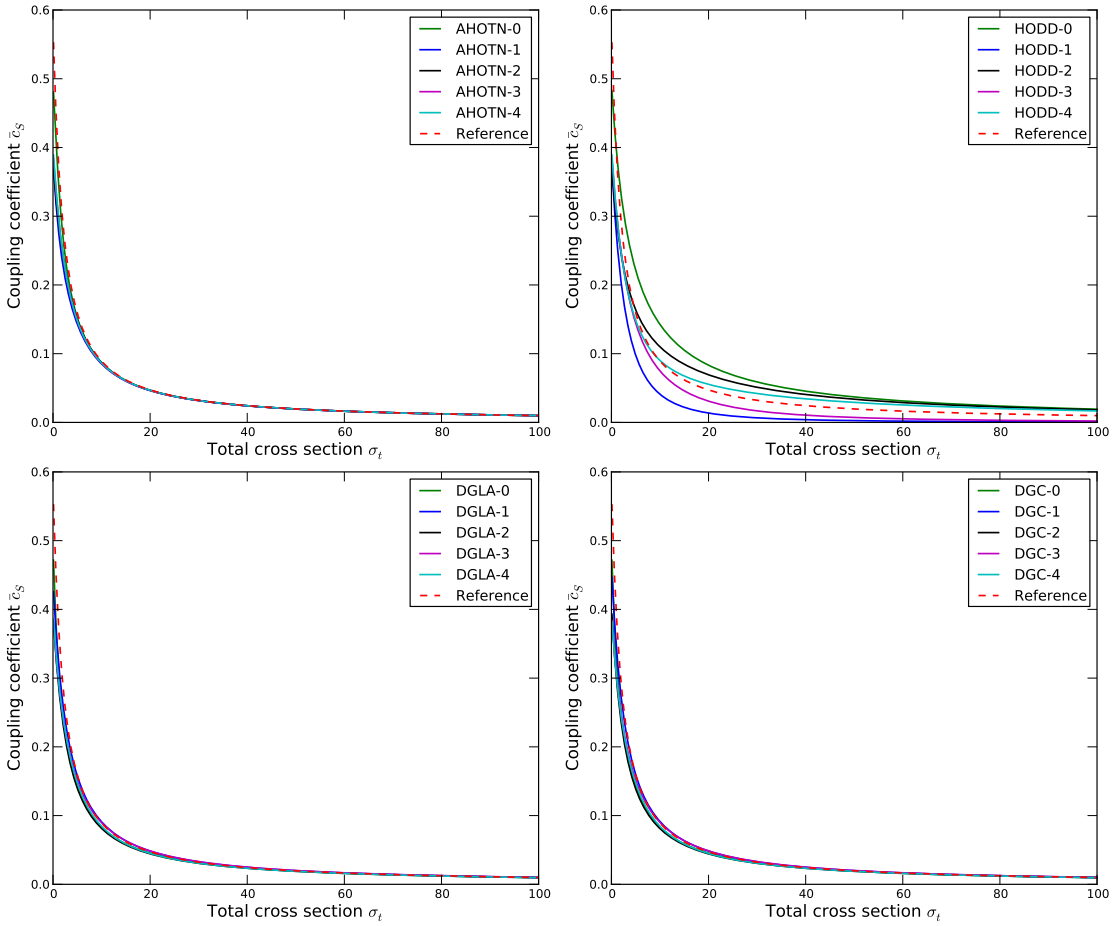


Figure 5.36: Coupling coefficients $\bar{c}_{S,N}$ (North outflow face) versus σ_t for test case II (non-unity aspect ratio) and AHOTN, HODD, DGLA, and DGC of orders $\Lambda = 0, ...4$.

case II the coupling coefficients differ between the different outflow faces. The said coupling coefficients are plotted versus σ_t in Figs. 5.34, 5.35, and 5.36 for test cases I (generic outflow face) and II (East and North outflow face), respectively.

The most important finding from Figs. 5.34 through 5.36 is that the coupling coefficient $\bar{c}_{S,F}$ is always positive. The immediate conclusion from this finding is that increasing the average source decreases the risk of negative outflow fluxes. In regions with strong distributed sources or large scattering cross sections, negative fluxes are less likely to occur. Second, the coupling with the volumetric cell source cannot cause negative outflow average fluxes because, by the conditions of a "healthy" mesh cell the cell, the averaged source is positive and so is the coupling coefficient, which ensures a positive contribution to the face averaged outflow fluxes.

The Inflow to Outflow Coupling Coefficients - Numerical Results

For discussion of the face-to-face average coupling coefficients the attention shall be restricted to test case II. In Figs. 5.37 through 5.39, the average coupling coefficients, $\bar{c}_{F,F'}$, are plotted for various discretization methods of order $\Lambda = 0, 1$, and 2, respectively.

The dashed, red line indicates the evolution of the exact coupling coefficients with increasing total cross section. Clearly, the West and East and South and North faces are uncoupled, while the Bottom and Top faces are coupled, but the coupling coefficient drops significantly when increasing σ_t to about unity. Therefore, one could consider the Bottom to Top coupling as weak. Note, all these face-to-face coupling combinations are of type $-k \rightarrow +k$, i.e. coupling of faces that are normal to the same coordinate axis (e.g. West (-x) to East (+x)).

It is in particular for the $-k \rightarrow +k,\ k = x, y, z$ face pairs that the numerical methods produce inaccurate coupling coefficients that do not even qualitatively reproduce their exact counterparts. For all other combinations, the numerical coupling coefficients reproduce the exact ones to a much higher fidelity. Negative face-averaged fluxes are caused by large magnitude negative coupling coefficients, which occur for the $-k \rightarrow +k,\ k = x, y, z$ combinations.

Looking at the West to East coupling coefficient obtained for the HODD method of orders $\Lambda = 0$, 1, and 2, an even-odd pattern emerges: for all expansion orders $\bar{c}_{W,E}$ is large in magnitude but for $\Lambda = 0$, and $\Lambda = 2$ it is negative, while for $\Lambda = 1$ it is positive. The opposite observation holds true for the DGLA and DGC method: for $\Lambda = 0$ and $\Lambda = 2$, $\bar{c}_{W,E}$ is positive, while it is negative for $\Lambda = 1$. Since the highest interpolation order for the DGLA/DGC-$(\Lambda + 1)$ method is identical to the interpolation order of the HODD-Λ method, the results suggest that for even interpolation orders, the $-k \rightarrow +k,\ k = x, y, z$ coupling coefficients are positive, and for odd interpolation orders they are negative. This behavior is expected to be visible in an even-odd pattern when comparing the performance of discretization methods with respect to their resilience against negative fluxes.

The inadequacy of the HODD method for meshes with large cell optical thicknesses discussed in subsection 5.1.6 is visible in Figs. 5.37 through 5.39 in that the HODD method's $-k \rightarrow +k,\ k = x, y, z$ coupling coefficients are the only ones that limit to ± 1 when increasing the total cross sections. This is an unphysical behavior because it defies the increasing attenuation that is expected when increasing the cell optical thickness. For the particular case of a negative coupling coefficient it translates into the occurrence of large negative outflow face-averaged fluxes for optically thick cells. In contrast, the DGLA and DGC methods may feature moderately large magnitude, negative coupling coefficients for intermediate optical thicknesses, $\sigma_t \approx 10$, but these coupling coefficients limit to zero as $\sigma_t \rightarrow \infty$.

In summary, looking at the face-to-face average coupling we learned a lot about the cause of negative outflow face-averaged fluxes. Neglecting the influence of higher-order flux moments, we found that faces that should be uncoupled or only weakly coupled, in particular the $W \rightarrow E$, $S \rightarrow N$, and $B \rightarrow T$, the exact coupling is not adequately represented by equations comprised in the numerical methods. For odd interpolation orders, these coupling coefficients become negative, leading to negative outflow face-averaged fluxes. For even interpolation orders the numerical coupling coefficients tend to be positive, thus not causing problems for the positivity of the discretization method.

5.2.2 Numerical Results from Lathrop's Test Problem

In this subsection, results from numerical experiments utilizing Lathrop's test problem, described in section 3.4, are presented to investigate the resilience of spatial discretization methods against negative fluxes for an S_N problem for which obtaining strictly positive flux proved to be difficult. First, the dependence of the methods' resilience against negative fluxes on the interpolation order is discussed. Subsequently, based on parameters listed in Table 3.4, methods are compared to one another to rank performance with respect to flux positivity.

The negativity measures τ_ψ^w and τ_ϕ^w, defined in section 3.5, are used to measure the extent of negative fluxes. The magnitude of τ_ψ^w and τ_ϕ^w should vary from zero to one with one being the worst and zero being an entirely positive solution. However, cases are possible where the magnitude of τ_ψ^w and τ_ϕ^w may be greater than one because the numerator and denominator in Eq. 3.47 are not taken from the same solution. The numerator comes from the solution obtained on a typically coarse mesh, while the denominator ideally comes from the exact solution which is replaced by a fine mesh solution in practice.

In Fig. 5.40 the τ_ψ^w measure is plotted versus the interpolation order for various discretization methods. The presented results are obtained from the Lathrop-III-1 test case. From the obtained data, a clear pattern emerges that confirms the expectation from the results presented in the preceding subsection: odd interpolation orders are more prone to developing negative

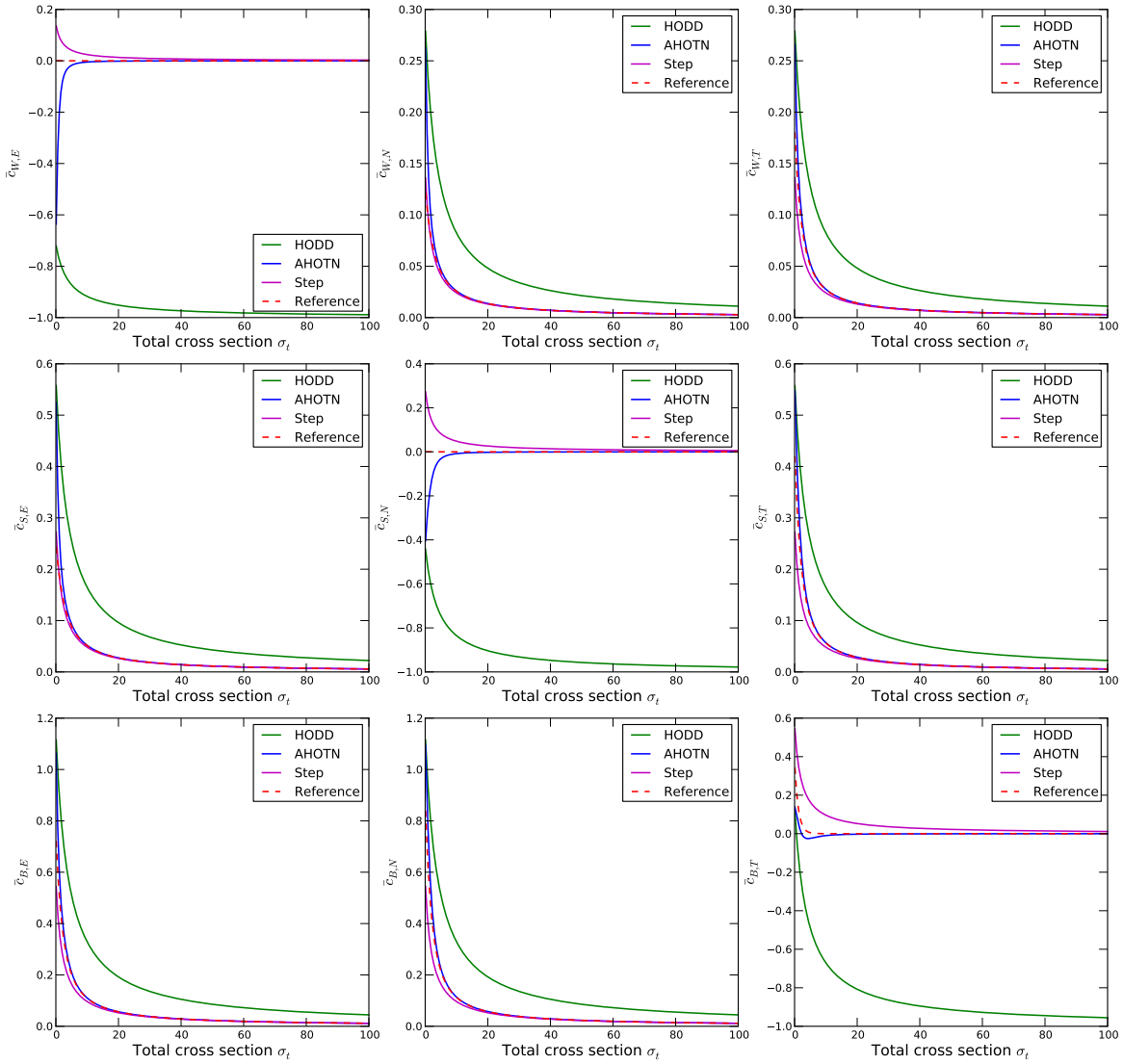


Figure 5.37: Coupling coefficients $\bar{c}_{F,F'}$ for $F = W, S, B$ and $F = E, N, T$ versus σ_t for test case II and AHOTN, HODD, DGLA and DGC of order $\Lambda = 0$. Note, DGLA and DGC are essentially the same method, the *Step Method*.

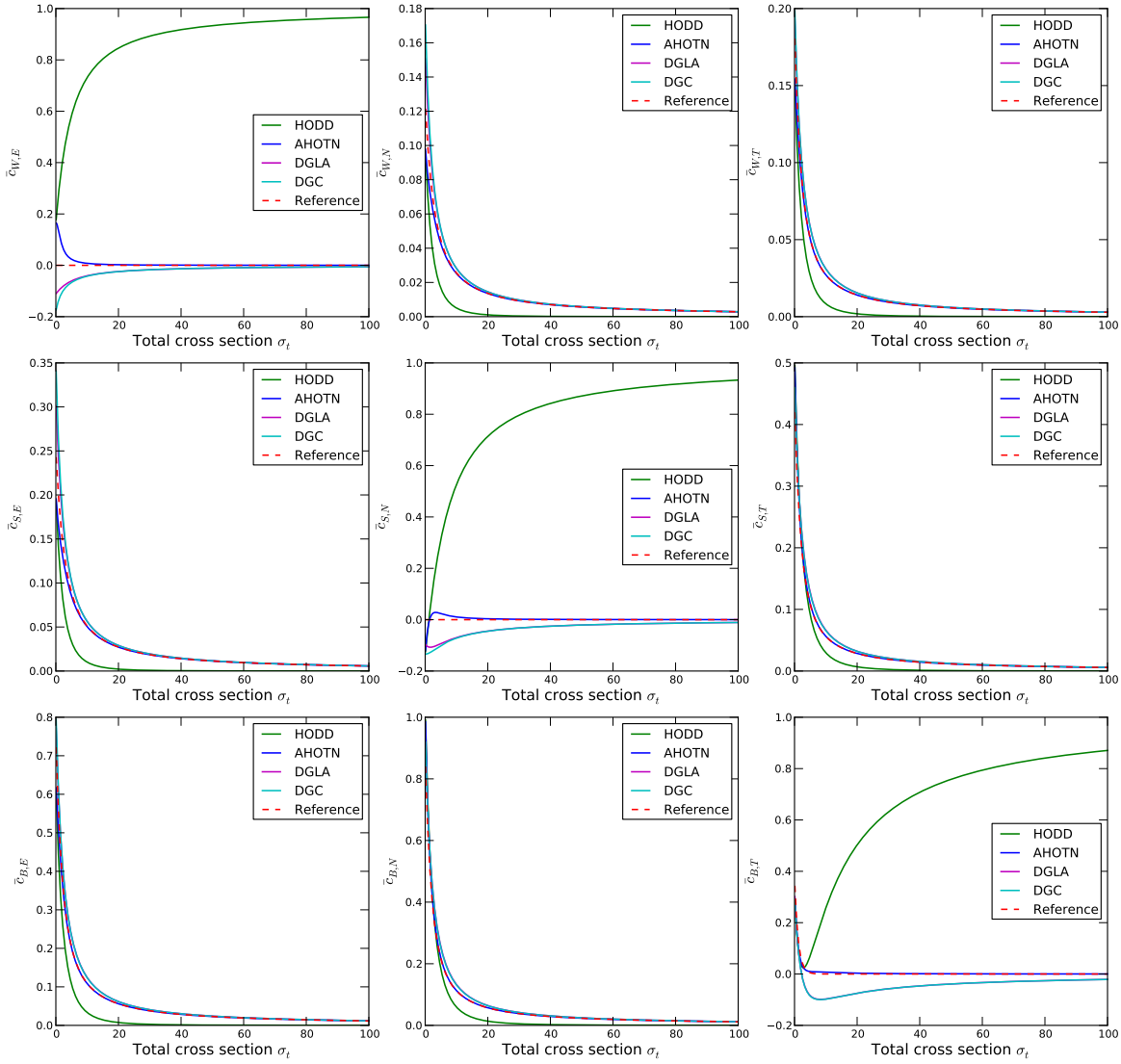


Figure 5.38: Coupling coefficients $\bar{c}_{F,F'}$ for $F = W, S, B$ and $F' = E, N, T$ versus σ_t for test case II and AHOTN, HODD, DGLA and DGC of order $\Lambda = 1$.

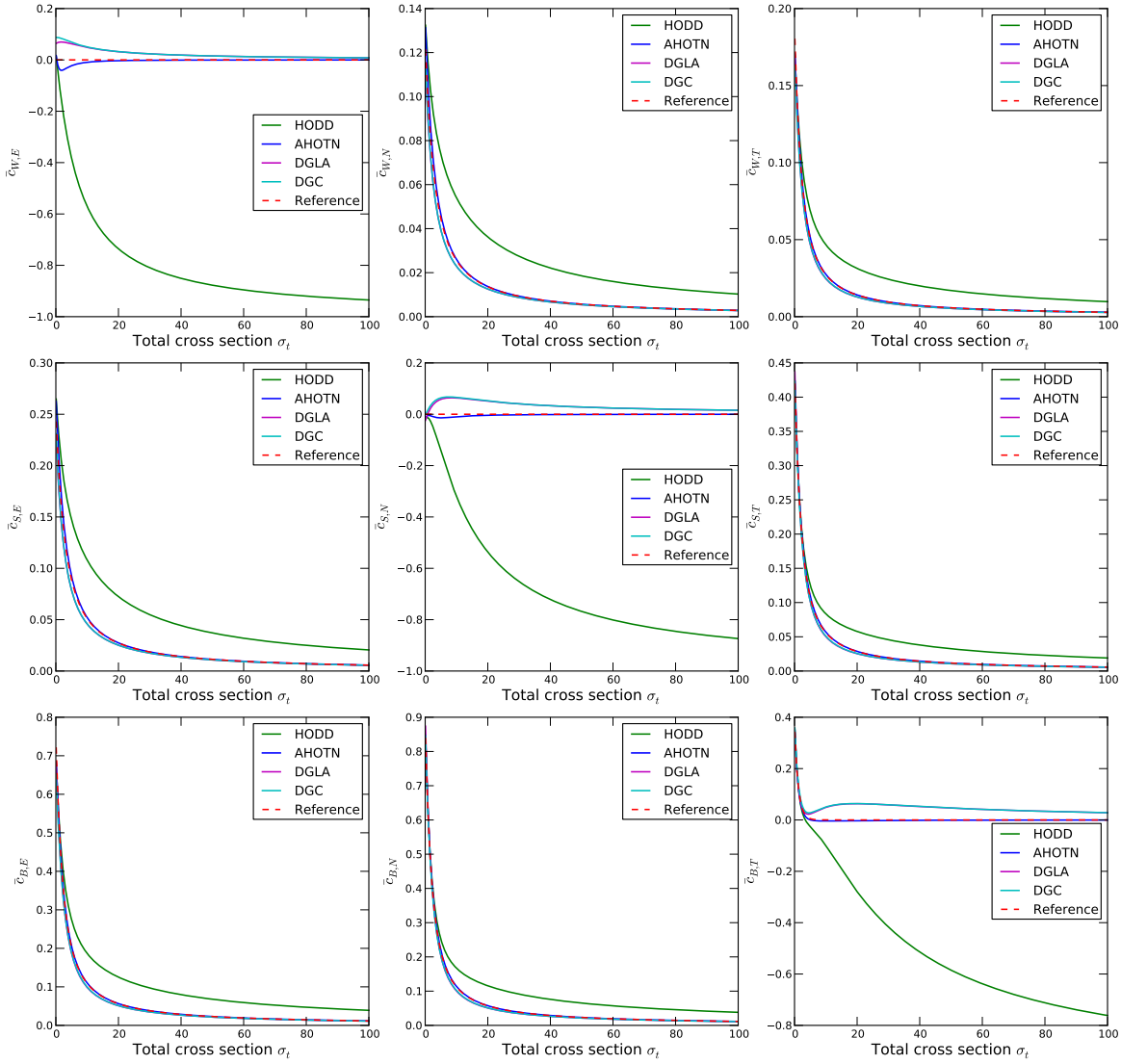


Figure 5.39: Coupling coefficients $\bar{c}_{F,F'}$ for $F = W, S, B$ and $F = E, N, T$ versus σ_t for test case II and AHOTN, HODD, DGLA and DGC of order $\Lambda = 2$.

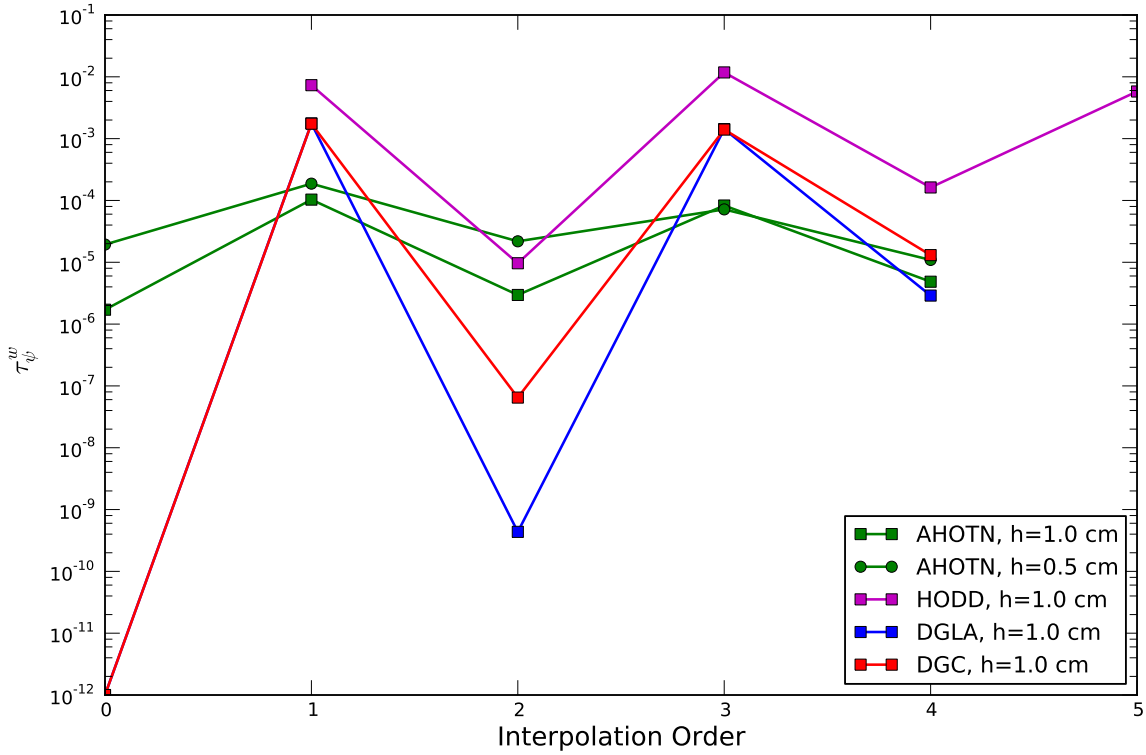


Figure 5.40: Negativity measure τ_ψ^w versus interpolation order for AHOTN, HODD, DGLA, and DGC methods for test case Lathrop-III-1.

fluxes than even expansion orders. This behavior is most pronounced for the DGLA and DGC methods, but also visible for the HODD and AHOTN methods. The AHOTN method features the least fluctuations with variation of Λ. The immediate consequence of this finding is that even interpolation orders should be preferred over odd interpolation orders if positivity of the solution is desired.

In Fig. 5.41, τ_ψ^w and τ_ϕ^w are plotted versus the cell optical thickness for Lathrop-I-1, Lathrop-II-1, and Lathrop-III-1 test cases. As τ_ψ^w and τ_ϕ^w can both become exactly zero, causing problems for the utilized loglog plots, we set:

$$\tau_k^w \leftarrow \min\left(\tau_k^w, 10^{-12}\right).$$

Hence, values for τ_ψ^w and τ_ϕ^w of 10^{-12} should be considered effectively zero.

For sufficiently fine meshes, both τ_ψ^w and τ_ϕ^w decrease with further mesh refinement demonstrating that positivity of the solution can be restored with mesh refinement. The scalar flux

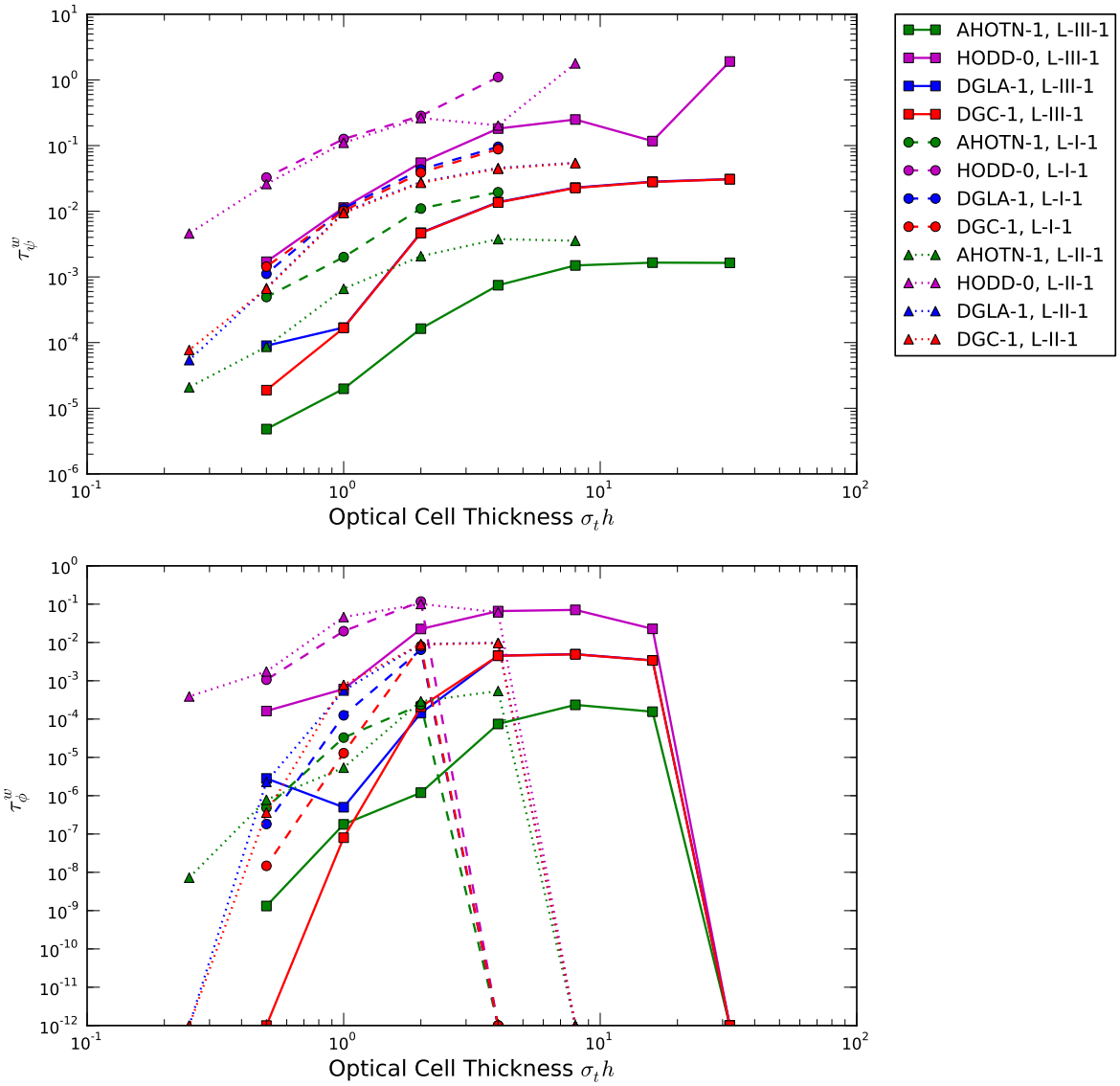


Figure 5.41: Evolution of negativity measures τ_ψ^w and τ_ϕ^w with refinement for Lathrop-I-1, Lathrop-II-1, and Lathrop-III-1 test cases.

measure τ_ϕ^w even completely vanishes for reasonable mesh refinement levels when using the DGLA or DGC methods of order one. However, for the hardest test case, Lathrop-III-1, the finest utilized mesh features $192^3 \approx 7 \times 10^6$ cells and a positive definite angular flux is not attained. From this perspective, today's computational resources may not always suffice to solve challenging problems on meshes that are resolved enough to guarantee positive definite solutions.

From the discussion of the coupling coefficients and the mesh refinement study Fig. 5.41, we inferred that negative fluxes occur on meshes that tend to feature optically thick cells on the order of about one to several mean free path. However, increasing the optical thickness further may decrease the negativity measure τ_ψ^w, as seen in Fig. 5.41 where most curves develop maxima for cells featuring in between one and ten mean-free paths. This observation is corroborated by the coupling coefficients described in the preceding subsection that cause negative cell outflow fluxes, some of which peak around cell optical thicknesses of this order, while others may start as large and negative and then limit to zero for $\sigma_t \to \infty$.

However, in the latter case, the positive coupling coefficients drop significantly from large positive values to almost zero when increasing the cell optical thickness to about one mfp. The large, positive coupling coefficients prevented the occurrence of negative outflow face-averaged fluxes, even with a single sizable negative coupling coefficient on optically thin meshes. The exception are the Diamond Difference results, which exhibit monotonically decreasing coupling coefficients that limit to -1. Even those, however, develop a maximum in their τ_ψ^w curves between one and ten mfp cell thicknesses. Additionally, further increasing the cell optical thickness leads to a sharp increase in τ_ψ^w for DD, which is not observed for the other depicted methods.

Comparing the evolution of τ_ψ^w and τ_ϕ^w curves for the same discretization method but different problem parameters (I-1, II-1, and III-3) shows differences up to about one order of magnitude. Therefore, the presence of negative fluxes in a solution measured via τ_ψ^w and τ_ϕ^w depends on the problem at hand; there is no universal curve that holds true for all test problems. However, the normalization to the problem's flux magnitude leads to $\tau_k^w, k = \psi, \phi$ curves that are of the same order of magnitude for different problem parameters. This leads to the hope that results obtained from Lathrop's test problem can be used to judge a solution's resilience against negative fluxes for other problems.

In Figs. 5.42 and 5.43, τ_ψ^w and τ_ϕ^w obtained for the L-III-1 problem are plotted versus the mesh spacing for various discretization methods in order to compare their resilience against negative flux solutions. Clearly, as conjectured, odd interpolation orders should be avoided as the performance is always worse than for the same discretization method employing an even interpolation order. Similarly, the HODD method's solution for all expansion orders is worse compared to the DGLA, DGC, and AHOTN methods of the equivalent interpolation

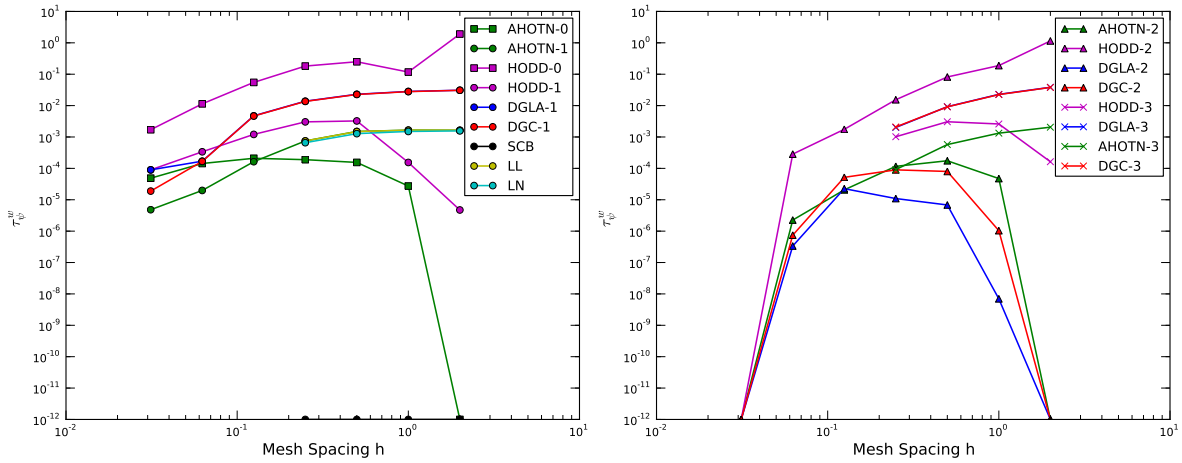


Figure 5.42: Negativity measure τ_{ψ}^{w} versus mesh spacing for test case Lathrop-III-1 and various spatial discretization methods.

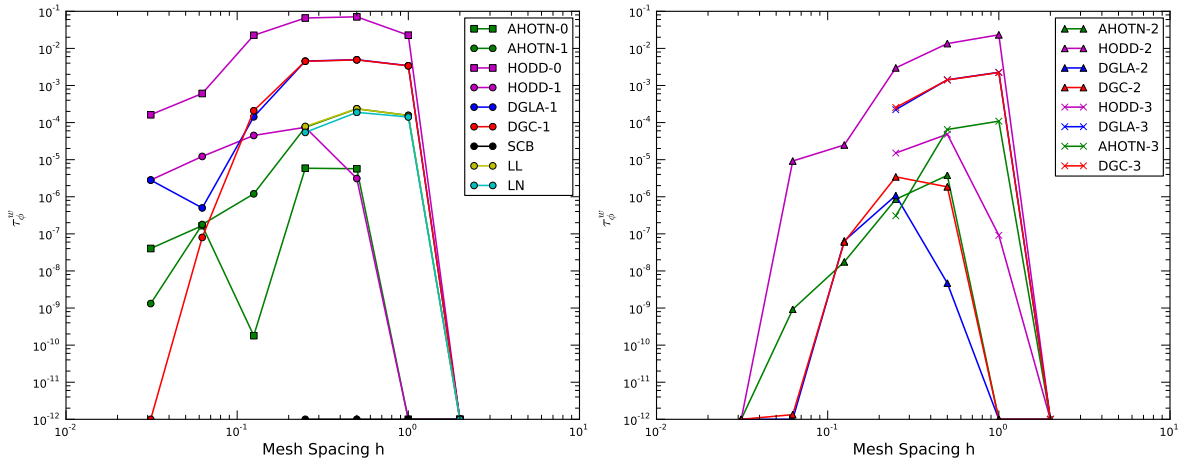


Figure 5.43: Negativity measure τ_{ϕ}^{w} versus mesh spacing for test case Lathrop-III-1 and various spatial discretization methods.

order. Therefore, utilizing HODD is also not recommended if resilience against negative fluxes is desired, i.e. small τ_ψ^w or τ_ϕ^w.

The choices that guarantee positive solutions are the *Step Method* (DGLA-0/DGC-0) and the *Step Characteristic Method*. However, those may be undesirable due to their poor accuracy. Therefore, in the remainder of this chapter the best alternatives to these methods shall be investigated.

In particular, four methods perform exceptionally well for the Lathrop-III-1 test case: AHOTN, DGLA, and DGC of order $\Lambda = 2$, and the *Simple Corner Balance Method* (SCB). While the first three methods' solutions exhibit negative fluxes, the SCB method's solution is positive for the Lathrop-III-1 test problem. This observation holds for both τ_ψ^w and τ_ϕ^w, depicted in Figs. 5.42 and 5.43, respectively. In contrast, AHOTN-2, DGLA-2, and DGC-2 obtain positive solutions only for the last mesh refinement level ($196^3 \approx 7 \times 10^6$ mesh cells). Among these three methods, the DGLA-2 method performs slightly better than the other two.

While performing poorly for the τ_ψ^w indicator, the AHOTN-0 method is among the better performing methods for the τ_ϕ^w measure depicted in Fig. 5.43, competing with the results obtained by the AHOTN, DGLA and DGC methods of order $\Lambda = 2$. Finally, the LL and LN methods perform almost identical to the AHOTN-1 method, and therefore do not belong to the set of well-performing methods with regards to solution positivity.

In Fig. 5.44 τ_ψ^w obtained for the Lathrop-III-2 test case is depicted. The results for the τ_ϕ^w measure yield the same conclusions as stated before and are therefore omitted. Changing the scattering ratio from 0.1 to 0.5 leaves many of the conclusions from the Lathrop-III-1 test problem unchanged. Remarkably, the SCB method still features a strictly positive solution; AHOTN-2, DGLA-2, and DGC-2 remain the runner-up methods. In contrast to the Lathrop-III-1 test problem, the DGLA-2 method performs significantly better than the DGC-2 method. Comparing AHOTN-2 and DGLA-2, the obtained τ_ψ^w values are comparable at larger mesh sizes, but DGLA returns a positive definite solution for the finest two meshes while AHOTN-2 does not.

Surprisingly, the AHOTN-0 method obtains a positive definite solution for all but the last mesh refinement level. This behavior differs significantly from the AHOTN-0 performance for the Lathrop-III-1 test case, where AHOTN-0 belongs to the set of poorly performing methods.

Finally, in Fig. 5.45 τ_ψ^w results are plotted for the Lathrop-III-3 test problem (scattering ratio of 0.9). With the exception of the LL, LN, DD, HODD-1, HODD-2 and all third-order methods, the obtained solutions for Lathrop-III-3 are strictly positive. That demonstrates that with increasing scattering ratio, the difficulty of obtaining a positive solution decreases. The HODD-0,1,2 methods' results are particular in that the solutions start out positive on coarse meshes and develop negative solutions with mesh refinement. Starting from these intermediate meshes, further mesh refinement is expected to decrease the measure τ_ψ^w, and strictly positive

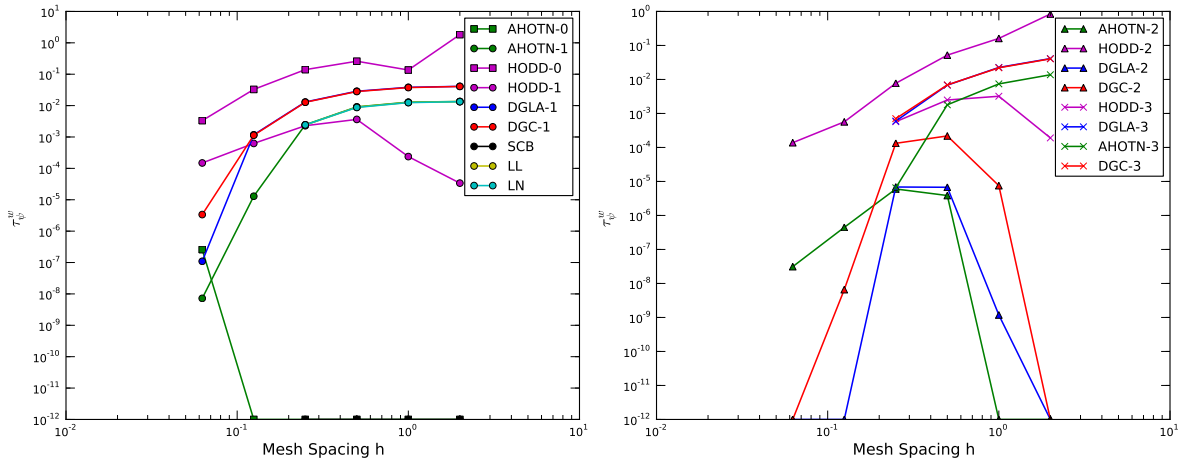


Figure 5.44: Negativity measure τ_ψ^w versus mesh spacing for test case Lathrop-III-2 and various spatial discretization methods.

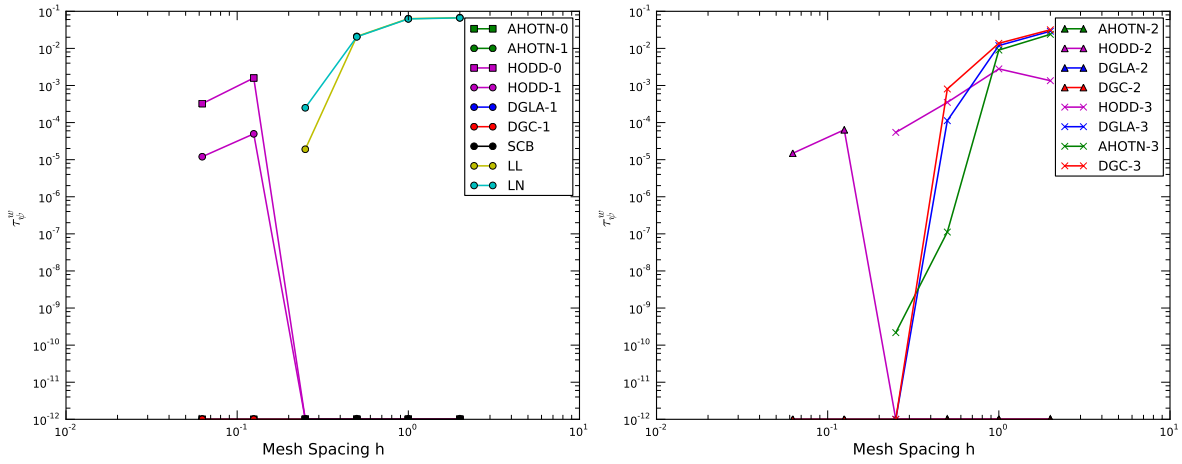


Figure 5.45: Negativity measure τ_ψ^w versus mesh spacing for test case Lathrop-III-3 and various spatial discretization methods.

solutions are expected on very fine meshes. In contrast, the LL, LN, and all third order methods exhibit τ_ψ^w that decrease with mesh refinement. The largest τ_ψ^w are obtained for the LL and LN methods. These results underscore several facts that were already pointed out before:

- Odd order methods are more prone to negative fluxes than even orders.
- The HODD method is more prone to negative fluxes than any of the other family of methods.
- The LL/LN methods are prone to negative fluxes, even for problems for which other methods produce strictly positive solutions.
- While worse in its resilience against negative fluxes than AHOTN-0 or AHOTN-2, the AHOTN-1 methods is still much better than the HODD-1 methods for example.

5.2.3 Negative Solutions and First Collision Source

Negative fluxes develop predominantly in regions with small distributed sources and/or scattering ratios. This is because larger cell-averaged sources always increase the flux levels, and reduce the likelihood of negative coupling from the inflow face fluxes to cause negative outflow average fluxes. In the particular case of Lathrop's test problem, negative fluxes occur in the outlying regions that feature no distributed source. The flux in the outlying regions comes from neutrons streaming from the central source region into the outlying region. In the discretization methods, streaming is represented by coupling of the inflow and outflow faces. As discussed in subsection 5.2.1, the coupling coefficients of the face averages can become negative, finally leading to negative volumetric fluxes in the source-free region.

Another problem associated with S_N solutions in large, source-free, low-scattering regions are ray effects[23]. While ray effects can produce negative fluxes, their predominant detriment to the S_N solution is that cells that are not intersected by a discrete ordinate drawn from the localized source to the said cell will feature a nonphysically small flux, or a zero flux in cases where the medium is non-scattering.

One well-known and effective remedy for ray effects is using a first collision source[23], [65], [66]. The solution is decomposed into the uncollided flux and the collided flux. The uncollided flux can be computed by a process called ray-tracing: Draw lines from the point source to the considered mesh cell and compute the optical length of these lines. The contribution from the point source to the flux within the mesh cell can be computed by simple exponential attenuation[4]. The collided flux is then computed by setting up the transport equation in a way

[4]This simplifies the process implemented in GRTUNCL[66] significantly. In fact, contributions from the source to the cell's scalar flux need to be accumulated for all angular directions that have intersections with the said mesh cell.

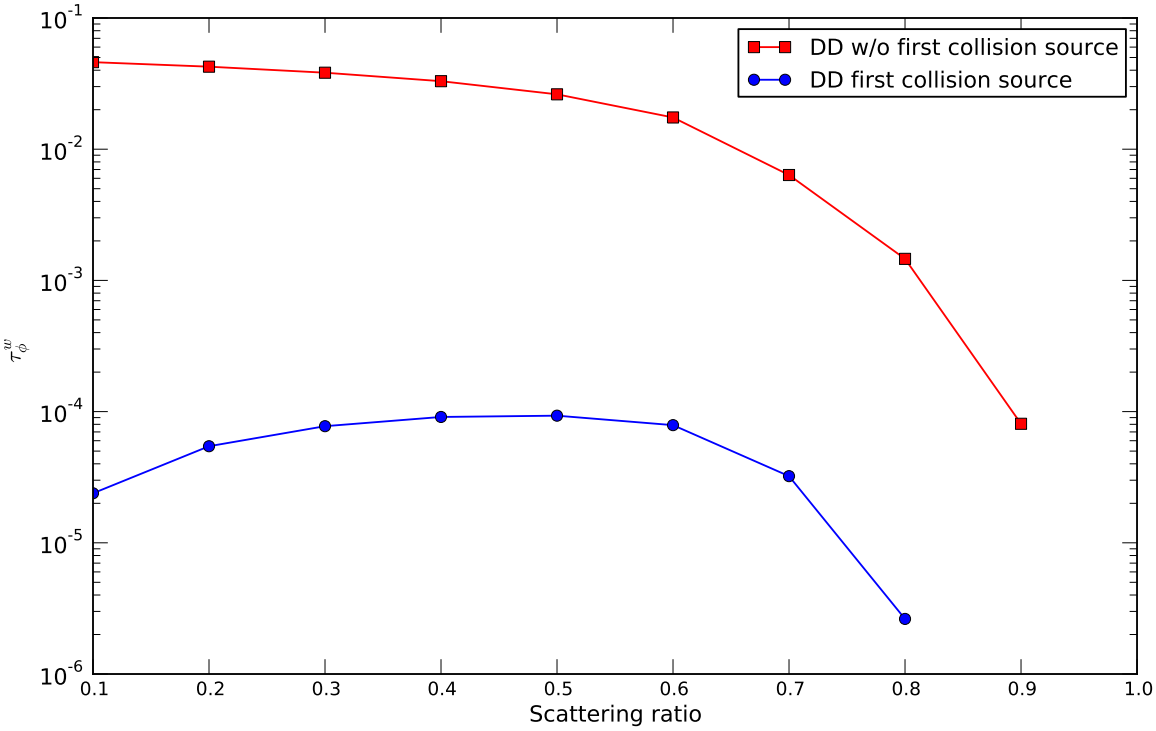


Figure 5.46: Negativity measure τ_ψ^w versus scattering ratio for DD solutions with and without using the first collision source.

that the scattering source spawned by the uncollided flux is used as external distributed source.

The key item that is of interest within this subsection is that the ray-tracing procedure creates a positive uncollided flux and hence a positive first-collision source. Therefore, the hope is that the first collision source may alleviate or even effectively eliminate negative fluxes in source-free, low-scattering regions.

For test purposes, the Lathrop-III test problem is solved for various scattering ratios on a 24^3 mesh (optical cell thickness: 2 mfp) with and without using the first collision source, computed by the first collision source algorithm implemented in DENOVO[24]. The negative-flux prone Diamond Difference discretization method is utilized for this test. The results in the τ_ϕ^w measure are depicted in Fig. 5.46. While using the first collision source reduces τ_ϕ^w by up to three orders of magnitude, it does not eliminate negative fluxes.

In summary, using a first collision source for the solution of the S_N transport problem even if ray-effects are not present, alleviates the problem of negative fluxes significantly. However, it does not eliminate negative fluxes, and thus falls short of the optimistic expectations.

5.3 Thick Diffusion Limit Problems

In this section certain properties of selected discretization schemes in the thick diffusion limit defined in section 3.6.1 are examined. This section augments findings in Ref. [7]: for several low-order methods, Diamond Difference, HODD-1, AHOTN-0, AHOTN-1, LL, and LN analysis will be presented that either shows that the respective methods do not have a diffusion limit or that they are candidates for having it. As the analysis in [7] already comprises the DGC-0,1 and DGLA-0,1 methods, these results shall only be tested numerically and no further analysis is performed on these methods in this work.

Numerical experiments based on the test case presented in sec. 3.6.1 are performed for all competing methods to support and extend the analysis performed for Diamond Difference, HODD-1, AHOTN-0, AHOTN-1, LL, and LN methods. This is only the first part of the analysis done in Ref. [7], which in addition looks at the robustness of methods in the thick diffusion limit and the quality of the boundary conditions. This extended analysis is not performed within this work.

Note, the notation is slightly modified in the following analysis in order to avoid excessively lengthy spatial indices. The discrete ordinates index becomes a superscript, while the spatial index becomes a subscript. The superscript h indicating approximate solution is dropped because all considered fluxes within this section are approximate fluxes.

5.3.1 Review of Adams Analysis

Adam's analysis is performed for the first order Discontinuous Finite Element Methods labeled DGLA-1 and DGC-1 within this work. In general, the method can be written as:

$$\hat{\Omega}\cdot\left[\mathbf{L}^{s}_{\vec{i}}\vec{\psi}^{n,s}_{\vec{i}}+\mathbf{L}_{\vec{i}}\vec{\psi}^{n}_{\vec{i}}\right]+\frac{1}{\epsilon}\mathbf{T}_{\vec{i}}\vec{\psi}^{n}_{\vec{i}}=\frac{1}{4\pi}\left[\frac{1}{\epsilon}\mathbf{T}_{\vec{i}}-\epsilon\mathbf{A}_{\vec{i}}\right]\vec{\phi}_{\vec{i}}+\frac{\epsilon}{4\pi}\vec{Q}_{\vec{i}}, \tag{5.52}$$

where $\mathbf{L}^{s}_{\vec{i}}$ is a surface matrix, $\mathbf{L}_{\vec{i}}$ is a stiffness matrix, $\mathbf{T}_{\vec{i}}=\sigma_t\mathbf{M}_{\vec{i}}$ is a mass matrix times the total cross section, $\mathbf{A}_{\vec{i}}=\sigma_a\mathbf{M}_{\vec{i}}$ is a mass matrix times the absorption cross section, $\vec{\psi}^{n,s}_{\vec{i}}$ collects all the surface unknowns and $\vec{\psi}^{n}_{\vec{i}}$ the volume unknowns. The cell index is denoted, as usual, by $\vec{i}$.

The following asymptotic expansions for the volume and surface unknowns are introduced:

$$\vec{\psi}^{n}_{\vec{i}}\left(\vec{r}\right) \;=\; \sum_{p=0}^{\infty}\epsilon^{p}\vec{\psi}^{n,[p]}_{\vec{i}}\left(\vec{r}\right). \tag{5.53}$$

Substitution of Eq. 5.53 into 5.52 and collecting powers of ϵ gives:

$$\mathcal{O}\left(\epsilon^{-1}\right) \quad : \quad \vec{\psi}_{\vec{i}}^{n,[0]} = \frac{1}{4\pi}\vec{\phi}_{\vec{i}}^{[0]} \tag{5.54}$$

$$\mathcal{O}\left(1\right) \quad : \quad \hat{\Omega}\cdot\left[\mathbf{L}_{\vec{i}}^{s}\vec{\psi}_{\vec{i}}^{n,s,[0]} + \mathbf{L}_{\vec{i}}\vec{\psi}_{n,\vec{i}}^{[0]}\right] + \mathbf{T}_{\vec{i}}\vec{\psi}_{\vec{i}}^{s,[1]} = \frac{1}{4\pi}\mathbf{T}_{\vec{i}}\vec{\phi}_{\vec{i}}^{s,[1]}. \tag{5.55}$$

From Eq. 5.54 it can be inferred that the leading order solution is isotropic, i.e. it is independent of direction index n.

Applying the quadrature operator $\sum_{n=1}^{N} w_n \cdot$ to Eq. 5.55 leads to:

$$\sum_{n=1}^{N} w_n\left[\hat{\Omega}\cdot\mathbf{L}_{\vec{i}}^{s}\vec{\psi}_{\vec{i}}^{n,s,[0]}\right] + \left[\sum_{n=1}^{N} w_n\hat{\Omega}\right]\cdot\mathbf{L}_{\vec{i}}\vec{\psi}_{\vec{i}}^{n,[0]} = -\mathbf{T}_{\vec{i}}\left[\sum_{n=1}^{N} w_n\vec{\psi}_{\vec{i}}^{s,[1]}\right] + \left[\sum_{n=1}^{N} w_n\right]\frac{1}{4\pi}\mathbf{T}_{\vec{i}}\vec{\phi}_{\vec{i}}^{s,[1]}. \tag{5.56}$$

Typical quadrature rules satisfy the following two conditions:

$$\begin{aligned}\sum_{n=1}^{N} w_n &= 4\pi \\ \sum_{n=1}^{N} w_n\hat{\Omega} &= \vec{0}.\end{aligned} \tag{5.57}$$

Using Eq. 5.57 in Eq. 5.56 results in:

$$\sum_{n=1}^{N} w_n\left[\hat{\Omega}\cdot\mathbf{L}_{\vec{i}}^{s}\vec{\psi}_{\vec{i}}^{n,s,[0]}\right] = 0. \tag{5.58}$$

Labeling the surfaces with index l and expanding the expression Eq. 5.58 according to the definition in [7] gives:

$$\sum_{l=1}^{L_{\vec{i}}}\int_{\mathcal{E}_l} dS v_{\vec{i},m}\left\{\left[\sum_{\hat{n}\cdot\hat{\Omega}>0} w_n\left(\hat{n}_{\vec{i},k}\cdot\hat{\Omega}\right)\psi_n^{n,[0]}\left(\vec{r}^{+}\right)\right] + \left[\sum_{\hat{n}\cdot\hat{\Omega}<0} w_n\left(\hat{n}_{\vec{i},k}\cdot\hat{\Omega}\right)\psi_n^{n,[0]}\left(\vec{r}^{-}\right)\right]\right\} = 0, \tag{5.59}$$

where $v_{\vec{i},m}$ is the m^{th} test function, and $\vec{r}^{+}$ and $\vec{r}^{-}$ are points on the faces just inside (interior trace) and outside (exterior trace) of the cell, respectively. Note that while DGFEM equations are local to each mesh cell, one could assemble a global system of equations by numbering test functions consecutively throughout all cells.

Now, since the leading order solution is isotropic, Eq. 5.59 can be manipulated to:

$$\sum_{l=1}^{L_{\vec{i}}} \int_{\mathcal{E}_l} dS v_{\vec{i},m} \sum_{\hat{n}\cdot\hat{\Omega}>0} w_n \left(\hat{n}_{\vec{i},k}\cdot\hat{\Omega}\right) \left[\phi^{n,[0]}\left(\vec{r}^{+}\right) - \phi^{n,[0]}\left(\vec{r}^{-}\right)\right] = 0, \tag{5.60}$$

Equation 5.60 relates the scalar flux within a cell to the upstream scalar fluxes and the boundary conditions. Thus, it can be used to assemble the global system of equations with the unknowns being the leading order scalar fluxes on the faces:

$$\mathbf{B}\vec{\phi} = \vec{\beta}, \tag{5.61}$$

where $\vec{\phi}$ collects all the interior face scalar flux unknowns and $\vec{\beta}$ is a vector that depends only on the boundary conditions and, in case of vacuum boundary conditions, is zero.

If the coefficient matrix $\mathbf{B}$ has full rank, then the problem's solution is determined solely by the boundary conditions, and in case of vacuum boundary conditions, is zero everywhere. This is unphysical because the solution of the S_N problem must also depend on the distributed source and material properties, which are not present in Eq. 5.61. Therefore, the conclusion is that the method cannot have a diffusion limit if the matrix $\mathbf{B}$ in Eq. 5.61 has full rank, but it may have a diffusion limit if $\mathbf{B}$ is rank-deficient.

5.3.2 Application of Adams' Analysis to First Order HODD and TMB Methods

In this section, the first stage of Adam's analysis is adapted to Balance-WDD style methods and applied to the Diamond Difference, HODD-1, AHOTN-0, AHOTN-1, LL, and LN methods to infer whether they have the potential to possess the thick diffusion limit. As all these methods share the same structure of equations, i.e. balance relations augmented with WDD relations, the balance relations shall be first analyzed, followed by the analysis of the six methods.

Analysis of the Balance Equations

The scaled balance equation of order $\vec{m}$ is given by:

$$\begin{aligned} &\sum_{k=x,y,z} \frac{\mu_{n,k}}{\Delta_k} \left[\psi^n_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - (-1)^{m_k}\, \psi^n_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} - 2 \sum_{l=0}^{[(m_k-1)/2]} (2m_k - 4l - 1)\psi^n_{\vec{i},\vec{m}-(2l+1)\hat{e}_k} \right] \\ + \quad &\frac{\sigma_t}{\epsilon}\psi^n_{\vec{i},\vec{m}} = \frac{1}{4\pi}\left(\frac{\sigma_t}{\epsilon} - \epsilon\sigma_a\right)\phi_{\vec{i},\vec{m}} + \frac{\epsilon}{4\pi}Q_{\vec{i},\vec{m}}, \end{aligned} \tag{5.62}$$

where position indices $\vec{i} \pm \hat{e}_k/2$ denote face flux-moments. Introducing a power series expansion in terms of ϵ for $\psi^n_{\vec{i},\vec{m}}$:

$$\begin{aligned} \psi^n_{\vec{i},\vec{m}} &= \sum_{p=0}^{\infty} \epsilon^p \psi^{n,[p]}_{\vec{i},\vec{m}} \\ \psi^n_{\vec{i}\pm\hat{e}_k/2,\vec{m}} &= \sum_{p=0}^{\infty} \epsilon^p \psi^{n,[p]}_{\vec{i}\pm\hat{e}_k/2,\vec{m}}, \end{aligned} \tag{5.63}$$

substituting it into Eq. 5.62, and then collecting equal powers of ϵ leads to:

$$\mathcal{O}\left(\epsilon^{-1}\right) \quad : \quad \psi^{n,[0]}_{\vec{i},\vec{m}} = \frac{1}{4\pi} \phi^{[0]}_{\vec{i},\vec{m}} \tag{5.64}$$

$$\begin{aligned} \mathcal{O}\left(1\right) \quad : \quad & \sum_{k=x,y,z} \frac{\mu_{n,k}}{\Delta_k} \left[\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - (-1)^{m_k} \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} \right] \\ - \quad & \sum_{k=x,y,z} \frac{2\mu_{n,k}}{\Delta_k} \left[\sum_{l=0}^{[(m_k-1)/2]} (2m_k - 4l - 1) \psi^{n,[0]}_{\vec{i},\vec{m}-(2l+1)\hat{e}_k} \right] \\ + \quad & \sigma_t \psi^{n,[1]}_{\vec{i},\vec{m}} = \frac{\sigma_t}{4\pi} \phi^{[1]}_{\vec{i},\vec{m}}. \end{aligned} \tag{5.65}$$

From Eq. 5.64 we conclude that $\psi^{n,[0]}_{\vec{i},\vec{m}}$ is isotropic, i.e. does not depend on the n index. Applying the quadrature $\sum_{n=1}^{N} w_n$ to Eq. 5.65 and noting that the quadrature rule satisfies Eq. 5.57 yields:

$$\sum_{n=1}^{N} w_n \left\{ \sum_{k=x,y,z} \frac{\mu_{n,k}}{\Delta_k} \left[\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - (-1)^{m_k} \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} \right] \right\} = 0 \tag{5.66}$$

Note that Eq. 5.66 evaluates for all possible combinations of m_x, m_y and m_z such that for HODD-1, for example, there would be 8 different instances of this equation. From here, asymptotic analysis of the WDD relations that are particular for each of the methods: HODD-1, AHOTN-1, LL, and LL are used to assemble the global matrix $\mathbf{B}$ by relating the face fluxes in cell $\vec{i}$ to the volumetric angular flux moments in this and the neighboring cells.

Analysis of the HODD-1 Method

The HODD-1 WDD relations are given by:

$$\frac{\mathrm{sgn}\left(\mu_{k,n}\right)}{2} \left(\psi^n_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - \psi^n_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} \right) = 3\psi^n_{\vec{i},\vec{g}_k}, \tag{5.67}$$

with $\vec{g}_k$ being defined as:

$$\vec{g}_k = \begin{cases} (1, m_y, m_z)^T & \text{if } k = x \\ (m_x, 1, m_z)^T & \text{if } k = y \\ (m_x, m_y, 1)^T & \text{if } k = z. \end{cases} \tag{5.68}$$

Substituting Eq. 5.63 and retaining only the $\mathcal{O}(1)$ terms gives:

$$\frac{\text{sgn}(\mu_{k,n})}{2}\left(\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k}\right) = 3\psi^{n,[0]}_{\vec{i},\vec{g}_k}. \tag{5.69}$$

From Eq. 5.69 we want to develop relations to eliminate the face fluxes from Eq. 5.66. If m_k is even, then we can just multiply by $2/\text{sgn}(\mu_{k,n})$:

$$\left(\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k}\right) = \frac{6}{\text{sgn}(\mu_{k,n})}\psi^{n,[0]}_{\vec{i},\vec{g}_k}. \tag{5.70}$$

If m_k is odd, then we need to find an expression for $\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k}$. This can be accomplished by adding twice the WDD relation owned by the neighboring upstream cell:

$$\begin{aligned} &\underline{\mu_{k,n} > 0:} \\ &\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}-3/2\hat{e}_k,\vec{m}^k} \\ &\underline{\mu_{k,n} < 0:} \\ &\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}+3/2\hat{e}_k,\vec{m}^k}. \end{aligned} \tag{5.71}$$

Note that additional face terms located on the $\vec{i} \pm 3/2\hat{e}_k$ face appeared in Eq. 5.71, which need to be eliminated. Thereto, the WDD relations for the $\vec{i} \pm 2\hat{e}_k$ cell can be utilized:

$$\begin{aligned} &\underline{\mu_{k,n} > 0:} \\ &\psi^{n,[0]}_{\vec{i}-3/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i}-2\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}-5/2\hat{e}_k,\vec{m}^k} \\ &\underline{\mu_{k,n} < 0:} \\ &\psi^{n,[0]}_{\vec{i}+3/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i}+2\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}+5/2\hat{e}_k,\vec{m}^k}. \end{aligned} \tag{5.72}$$

Substitution into Eq. 5.71 gives:

$$\begin{aligned} &\underline{\mu_{k,n} > 0:} \\ &\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}-2\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}-5/2\hat{e}_k,\vec{m}^k} \\ &\underline{\mu_{k,n} < 0:} \\ &\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} + 12\psi^{n,[0]}_{\vec{i}+2\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{\vec{i}+5/2\hat{e}_k,\vec{m}^k}. \end{aligned} \tag{5.73}$$

We can continue using the WDD relations of neighboring cells in this manner until we reach the boundary at which point we obtain the following equation:

$$\begin{aligned}
&\underline{\mu_{k,n} > 0:} \\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{i_k-1}\psi^{n,[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{1/2,\vec{m}^k} \\
&\underline{\mu_{k,n} < 0:} \\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = 6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{I_k-i_k}\psi^{n,[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k}, \qquad (5.74)
\end{aligned}$$

where dimension k features I_k linear intervals. Substituting Eq. 5.74 into Eq. 5.66 results in:

$$\begin{aligned}
&6\sum_{n=1}^{N} w_n \left\{ \sum_{k:m_k \text{ even}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{\vec{i},\vec{g}_k} \right\} \\
+ &\sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \left[6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{i_k-1}\psi^{n,[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{1/2,\vec{m}^k} \right] \right\} \\
- &\sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \left[6\psi^{n,[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{I_k-i_k}\psi^{n,[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} + 2\psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k} \right] \right\} = 0, \qquad (5.75)
\end{aligned}$$

where the summation over angle for the odd m_k is broken into two partial summations for $\mu_{n,k}$ greater and smaller than zero. The notation $k : m_k$even/odd in the summation denotes that that the summation over k is only performed if m_k is even or odd, respectively.

Now, using the result that the volume moments are isotropic, Eq. 5.75 can be written in terms of scalar flux volume moments:

$$\begin{aligned}
&\sum_{k:m_k \text{ even}} 6\phi^{[0]}_{\vec{i},\vec{g}_k} \left\{ \sum_{n=1}^{N} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right\} + \sum_{k:m_k \text{ odd}} \left[6\phi^{[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{i_k-1}\phi^{[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} \right] \left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right\} \\
- &\sum_{k:m_k \text{ odd}} \left[6\phi^{[0]}_{\vec{i},\vec{g}_k} + 12\sum_{l=1}^{I_k-i_k}\phi^{[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} \right] \left\{ \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right\} \\
= &-8\pi \sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{1/2,\vec{m}^k} \right\} + 8\pi \sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k} \right\}. \qquad (5.76)
\end{aligned}$$

Typically, for angular quadratures, it holds that

$$w_n = w_{n'} \text{ if } \mu_{n,k} = -\mu_{n',k}. \tag{5.77}$$

At this point it is convenient to define the quantity ρ_k:

$$\rho_k = 2\left\{\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\right\} = 2\left\{\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\right\} = \left\{\sum_{n=1}^{N} w_n \frac{|\mu_{n,k}|}{\Delta_k}\right\}, \tag{5.78}$$

to rewrite Eq. 5.76 as:

$$\begin{aligned}
& 6 \sum_{k:m_k \text{ even}} \rho_k \phi^{[0]}_{\vec{i},\vec{g}_k} \\
+ & \sum_{k:m_k \text{ odd}} \left\{ \frac{1}{2}\rho_k \left[6\phi^{[0]}_{\vec{i},\vec{g}_k} + 12 \sum_{l=1}^{i_k-1} \phi^{[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} \right] - \frac{1}{2}\rho_k \left[6\phi^{[0]}_{\vec{i},\vec{g}_k} + 12 \sum_{l=1}^{I_k-i_k} \phi^{[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} \right] \right\} \\
= & -8\pi \sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{1/2,\vec{m}^k} \right\} + 8\pi \sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k} \right\}.
\end{aligned} \tag{5.79}$$

The terms involving odd m_k can further be simplified by combining the two sums over k:

$$\begin{aligned}
& \sum_{k:m_k \text{ even}} \rho_k \phi^{[0]}_{\vec{i},\vec{g}_k} + \sum_{k:m_k \text{ odd}} \rho_k \left[\sum_{l=1}^{i_k-1} \phi^{[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} - \sum_{l=1}^{I_k-i_k} \phi^{[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} \right] \\
= & -\frac{4\pi}{3} \sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{1/2,\vec{m}^k} \right\} \\
+ & \frac{4\pi}{3}\pi \sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k} \right\}.
\end{aligned} \tag{5.80}$$

In the further development, a uniform mesh with $\Delta x = \Delta y = \Delta z$ and level symmetric quadrature is assumed. Under these conditions it holds that

$$\rho = \rho_x = \rho_y = \rho_z. \tag{5.81}$$

Then, Eqs. 5.79 and 5.80 can be written in an even simpler form:

$$\begin{aligned}
&\sum_{k:m_k \text{ even}} \phi^{[0]}_{\vec{i},\vec{g}_k} + \sum_{k:m_k \text{ odd}} \left[\sum_{l=1}^{i_k-1} \phi^{[0]}_{\vec{i}-l\hat{e}_k,\vec{g}_k} - \sum_{l=1}^{I_k-i_k} \phi^{[0]}_{\vec{i}+l\hat{e}_k,\vec{g}_k} \right] \\
= \; & -\frac{4\pi}{3\rho} \sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{1/2,\vec{m}^k} \right\} \\
+ \; & \frac{4\pi}{3\rho}\pi \sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k:m_k \text{ odd}} \frac{|\mu_{n,k}|}{\Delta_k} \psi^{n,[0]}_{I_k+1/2\hat{e}_k,\vec{m}^k} \right\}. \qquad (5.82)
\end{aligned}$$

From Eq. 5.82, the **B** matrix can be constructed. A sparsity pattern plot of the HODD-1 method's **B** matrix is depicted in Fig. 5.47 for a 4^3 mesh. The matrix is of size 512×512 and has full rank as determined by the Mathematica notebook listed in Appendix E.1. Therefore, we conclude that the HODD-1 methods does not possess the thick diffusion limit.

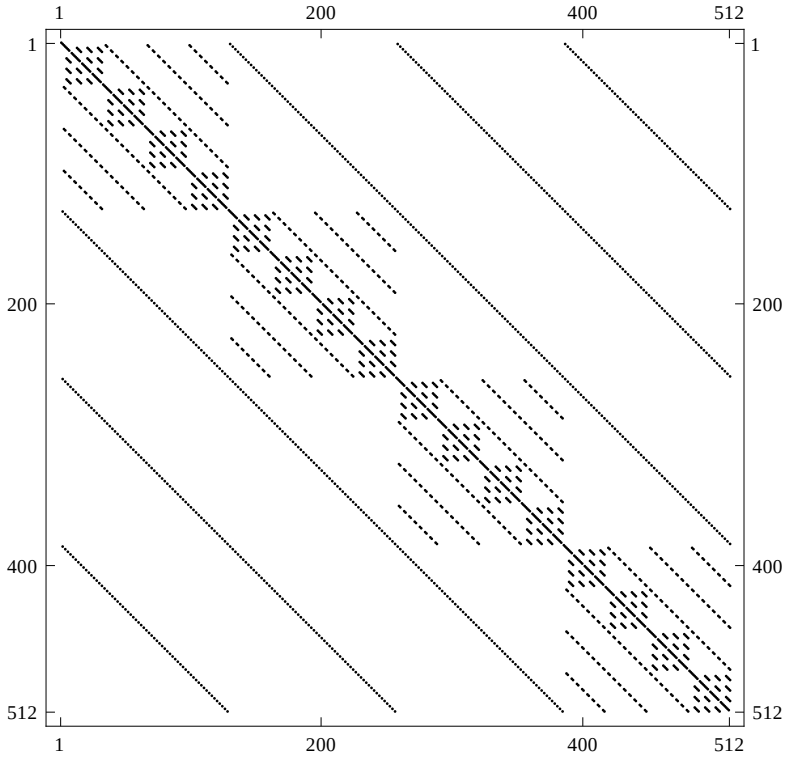


Figure 5.47: Sparsity pattern plot of the HODD-1 **B** matrix. The matrix has full rank.

Analysis of the Diamond Difference Method

The Diamond Difference method has a thick diffusion limit in one-dimensional slab geometry. However, it will be shown in this analysis that it does not possess the thick diffusion limit in three-dimensional Cartesian geometry. The Diamond Diamond Difference relations are given by:

$$\frac{1}{2}\left(\psi^{n}_{\vec{i}+1/2\hat{e}_k}+\psi^{n}_{\vec{i}-1/2\hat{e}_k}\right)=\psi^{n}_{\vec{i}}, \tag{5.83}$$

where the moment indices are omitted because it is clear that for the Diamond Difference method only the averages appear in the equations. Substituting the power series expansions, Eq. 5.63, and retaining only the $\mathcal{O}(1)$ term leads to:

$$\frac{1}{2}\left(\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}+\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}\right)=\psi^{n,[0]}_{\vec{i}}. \tag{5.84}$$

In the balance equation we have terms of the form $\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}-\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}$, which shall be written in terms of the cell-averaged fluxes. In order to obtain the desired differences of face-averaged flux values, we subtract twice the DD relation, Eq. 5.84, of the upstream cell:

$$\begin{aligned}
&\underline{\mu_{k,n}>0:}\\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}-\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}=2\psi^{n,[0]}_{\vec{i}}-4\psi^{n,[0]}_{\vec{i}-\hat{e}_k}+2\psi^{n,[0]}_{\vec{i}-3/2\hat{e}_k}\\
&\underline{\mu_{k,n}<0:}\\
&\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}-\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}=2\psi^{n,[0]}_{\vec{i}}-4\psi^{n,[0]}_{\vec{i}+\hat{e}_k}+2\psi^{n,[0]}_{\vec{i}+3/2\hat{e}_k}.
\end{aligned} \tag{5.85}$$

The average on the $\vec{i}\pm 3/2\hat{e}_k$ face can be replaced by using the DD equation of the $\vec{i}\pm 2\hat{e}_k$ cell:

$$\begin{aligned}
&\underline{\mu_{k,n}>0:}\\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}-\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}=2\psi^{n,[0]}_{\vec{i}}-4\psi^{n,[0]}_{\vec{i}-\hat{e}_k}+4\psi^{n,[0]}_{\vec{i}-2\hat{e}_k}-2\psi^{n,[0]}_{\vec{i}-5/2\hat{e}_k}\\
&\underline{\mu_{k,n}<0:}\\
&\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k}-\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k}=2\psi^{n,[0]}_{\vec{i}}-4\psi^{n,[0]}_{\vec{i}+\hat{e}_k}+4\psi^{n,[0]}_{\vec{i}+2\hat{e}_k}-2\psi^{n,[0]}_{\vec{i}+5/2\hat{e}_k}.
\end{aligned} \tag{5.86}$$

The face average on the $\vec{i}\pm(l+1/2)$ can generally be removed by using the $\vec{i}\pm(l+1)$ DD

equation such that the final result for Eq. 5.86 is:

$$\begin{aligned}
&\underline{\mu_{k,n} > 0:} \\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k} - \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k} = 2\psi^{n,[0]}_{\vec{i}} + 4\sum_{l=1}^{i_k-1}(-1)^l\psi^{n,[0]}_{\vec{i}-l\hat{e}_k} - 2(-1)^{i_k}\psi^{n,[0]}_{1/2} \\
&\underline{\mu_{k,n} < 0:} \\
&\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k} - \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k} = 2\psi^{n,[0]}_{\vec{i}} + 4\sum_{l=1}^{I_k-i_k}(-1)^l\psi^{n,[0]}_{\vec{i}-l\hat{e}_k} - 2(-1)^{i_k-I_k-1}\psi^{n,[0]}_{I_k+1/2\hat{e}_k}. \quad (5.87)
\end{aligned}$$

Substitution of Eq. 5.87 into Eq. 5.66 yields:

$$\begin{aligned}
&\sum_{\mu_{n,k}>0} w_n \left\{ \sum_{k=x,y,z} \frac{|\mu_{n,k}|}{\Delta_k} \left[2\psi^{n,[0]}_{\vec{i}} + 4\sum_{l=1}^{i_k-1}(-1)^l\psi^{n,[0]}_{\vec{i}-l\hat{e}_k} - 2(-1)^{i_k}\psi^{n,[0]}_{1/2} \right] \right\} \\
+ &\sum_{\mu_{n,k}<0} w_n \left\{ \sum_{k=x,y,z} \frac{|\mu_{n,k}|}{\Delta_k} \left[2\psi^{n,[0]}_{\vec{i}} + 4\sum_{l=1}^{I_k-i_k}(-1)^l\psi^{n,[0]}_{\vec{i}-l\hat{e}_k} - 2(-1)^{i_k-I_k-1}\psi^{n,[0]}_{I_k+1/2\hat{e}_k} \right] \right\} = 0. \quad (5.88)
\end{aligned}$$

The leading order cell-averaged angular fluxes are isotropic and can be replaced by scalar fluxes using Eq. 5.64:

$$\begin{aligned}
&\left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right\} \left\{ \sum_{k=x,y,z} \left[2\phi^{[0]}_{\vec{i}} + 4\sum_{l=1}^{i_k-1}(-1)^l\phi^{[0]}_{\vec{i}-l\hat{e}_k} \right] \right\} \\
+ &\left\{ \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right\} \left\{ \sum_{k=x,y,z} \left[2\phi^{[0]}_{\vec{i}} + 4\sum_{l=1}^{I_k-i_k}(-1)^l\phi^{[0]}_{\vec{i}-l\hat{e}_k} \right] \right\} \\
= &\ 4\pi \left[\sum_{k=x,y,z} \left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[2(-1)^{i_k}\psi^{n,[0]}_{1/2} \right] \right\} \right. \\
- &\left. \left\{ \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[2(-1)^{i_k-I_k-1}\psi^{n,[0]}_{I_k+1/2\hat{e}_k} \right] \right\} \right]. \quad (5.89)
\end{aligned}$$

Using definition Eq. 5.78 and its properties, Eq. 5.81, we can manipulate Eq. 5.89 into:

$$\begin{aligned}
&\left\{ \sum_{k=x,y,z} \left[\phi_{\vec{i}}^{[0]} + \sum_{l=1}^{i_k-1} (-1)^l \phi_{\vec{i}-l\hat{e}_k}^{[0]} - \sum_{l=1}^{I_k-i_k} (-1)^l \phi_{\vec{i}-l\hat{e}_k}^{[0]} \right] \right\} \\
= \quad & \frac{2\pi}{\rho} \left[\sum_{k=x,y,z} \left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[2(-1)^{i_k} \psi_{1/2}^{n,[0]} \right] \right\} \right. \\
- \quad & \left. \left\{ \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[2(-1)^{i_k - I_k - 1} \psi_{I_k+1/2\hat{e}_k}^{n,[0]} \right] \right\} \right] . \qquad (5.90)
\end{aligned}$$

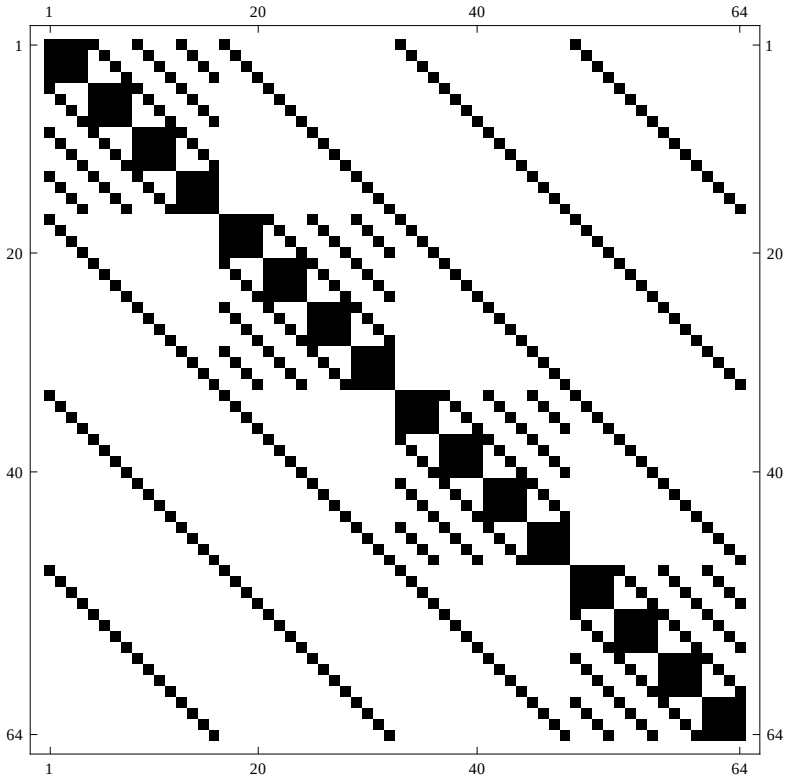


Figure 5.48: Sparsity pattern plot of the DD **B** matrix. The matrix has full rank.

From Eq. 5.90 we can construct the **B** matrix for the DD method. A sparsity pattern plot of the **B** matrix for the DD method is presented in Fig 5.48 for the case of 4^3 cells. The dimension of the **B** matrix is 64×64 and it is full rank. Therefore, we conclude that the DD method does not possess the thick diffusion limit in contrast to slab geometry, where it does

possess the thick diffusion limit. A listing of the Mathematica notebook to create the DD **B** matrix can be found in Appendix E.2.

Analysis of the AHOTN-1 Method

The AHOTN-1 WDD relations along direction $k = x, y, z$ are given by:

$$\frac{1+\alpha_{n,k}}{2}\psi^n_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} + \frac{1-\alpha_{n,k}}{2}\psi^n_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = \psi^n_{\vec{i},\vec{b}_k} + 3\alpha_{n,k}\psi^n_{\vec{i},\vec{g}_k}, \tag{5.91}$$

where $\vec{b}_k$ is given by:

$$\vec{b}_k = \begin{cases} (0, m_y, m_z)^T & \text{if } k = x \\ (m_x, 0, m_z)^T & \text{if } k = y \\ (m_x, m_y, 0)^T & \text{if } k = z. \end{cases} \tag{5.92}$$

Noting that the spatial weights, $\alpha_{n,k}$, asymptotically behave like:

$$\alpha_{n,k} = \text{sgn}\left(\mu_{k,n}\right)\left[1 + \mathcal{O}\left(\epsilon\right)\right], \tag{5.93}$$

the AHOTN-1 WDD relations for the leading order angular face and volume moments can be derived to be:

$$\begin{aligned} \underline{\mu_{k,n} > 0:} \quad & \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} = \psi^{n,[0]}_{\vec{i},\vec{b}_k} + 3\psi^{n,[0]}_{\vec{i},\vec{g}_k} \\ \underline{\mu_{k,n} < 0:} \quad & \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = \psi^{n,[0]}_{\vec{i},\vec{b}_k} - 3\psi^{n,[0]}_{\vec{i},\vec{g}_k}. \end{aligned} \tag{5.94}$$

Adding/Subtracting the WDD relation for the upstream cell gives:

$$\begin{aligned} \underline{\mu_{k,n} > 0:} \quad & \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} \pm \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} = \left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} \pm \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k}\right) + 3\left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} \pm \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k}\right) \\ \underline{\mu_{k,n} < 0:} \quad & \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} \pm \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} = \left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} \pm \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k}\right) - 3\left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} \pm \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k}\right) \end{aligned} \tag{5.95}$$

Now Eq. 5.95 is substituted into Eq. 5.66:

$$\begin{aligned}
&\sum_{k:m_k \text{ even}} \left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[\left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} - \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} - \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \right. \\
&+ \left. \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[\left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} - \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} - \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \right\} \\
&+ \sum_{k:m_k \text{ odd}} \left\{ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[\left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} + \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} + \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \right. \\
&- \left. \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \left[\left(\psi^{n,[0]}_{\vec{i},\vec{b}_k} + \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\psi^{n,[0]}_{\vec{i},\vec{g}_k} + \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \right\} = 0. \qquad (5.96)
\end{aligned}$$

As the leading order angular flux moments are isotropic, they can be replaced by the scalar fluxes:

$$\begin{aligned}
&\sum_{k:m_k \text{ even}} \left\{ \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \left[\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right] \right. \\
&+ \left. \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \left[\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right] \right\} \\
&+ \sum_{k:m_k \text{ odd}} \left\{ \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \left[\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right] \right. \\
&- \left. \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \left[\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k} \right] \right\} = 0. \qquad (5.97)
\end{aligned}$$

Now Eq. 5.78 is used and, considering property Eq. 5.81, one obtains:

$$\begin{aligned}
&\sum_{k:m_k \text{ even}} \rho \left\{ \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \right. \\
&+ \left. \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \right\} \\
&+ \sum_{k:m_k \text{ odd}} \rho \left\{ \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} \right) + 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right] \right. \\
&- \left. \left[\left(\phi^{[0]}_{\vec{i},\vec{b}_k} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) - 3 \left(\phi^{[0]}_{\vec{i},\vec{g}_k} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] \right\} = 0. \qquad (5.98)
\end{aligned}$$

The final equation is obtained by noting that ρ can be pulled out of the summation over k:

$$\sum_{k:m_k \text{ even}} \left[\left(2\phi^{[0]}_{\vec{i},\vec{b}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) + 3 \left(\phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} \right) \right]$$
$$+ \sum_{k:m_k \text{ odd}} \left[\left(\phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{b}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{b}_k} \right) + 3 \left(2\phi^{[0]}_{\vec{i},\vec{g}_k} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{g}_k} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{g}_k} \right) \right] = 0. \tag{5.99}$$

Note, that Eq. 5.99 only holds for interior cells, while for boundary cells Eq. 5.66 can be manipulated:

$$\sum_{n=1}^{N} w_n \left\{ \sum_{k=x,y,z\neq k'} \frac{\mu_{n,k}}{\Delta_k} \left[\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k,\vec{m}^k} - (-1)^{m_k} \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k,\vec{m}^k} \right] \right\}$$
$$+ \sum_{\mu_{n,k'}\cdot\hat{n}>0} w_n \frac{\mu_{n,k'}}{\Delta_{k'}} \left[\psi^{n,[0]}_{\vec{i}+1/2\hat{e}'_k,\vec{m}^{k'}} - (-1)^{m_{k'}} \psi^{n,[0]}_{\vec{i}-1/2\hat{e}'_k,\vec{m}^{k'}} \right]$$
$$+ \sum_{\mu_{n,k'}\cdot\hat{n}<0} w_n \frac{\mu_{n,k'}}{\Delta_{k'}} \psi^{n,[0]}_{\vec{i}+1/2\hat{e}'_k,\vec{m}^{k'}} = \sum_{\mu_{n,k'}\cdot\hat{n}<0} w_n \frac{\mu_{n,k'}}{\Delta_{k'}} (-1)^{m_{k'}} \psi^{n,[0]}_{BC,\vec{m}^{k'}}, \tag{5.100}$$

where it is assumed that $\vec{i} - 1/2\hat{e}'_k$ is the boundary face, but equivalent expressions could be derived otherwise. The rest of the analysis would then proceed as demonstrated before.

Equation 5.99 is now used to construct the **B** matrix of the AHOTN-1 method for the case of 4^3 cells. The sparsity pattern is plotted in Fig. 5.49: the matrix has dimensions 512×512 and rank 485. Thus it has 27 redundant rows. By determining the rank for varying number of cells I^3 we find that **B** has $(I-1)^3$ redundant rows, i.e. one redundancy for each interior vertex[5]. This property is referred to by Adams[7] as characterizing a *full resolution* method. The AHOTN-1 method possesses the thick diffusion limit, and is a full resolution method. A listing of the Mathematica notebook performing the said operation can be found in Appendix E.3.

Analysis of the AHOTN-0 Method

The WDD relation for the AHOTN-0 method is:

$$\frac{1+\alpha_{n,k}}{2} \psi^n_{\vec{i}+1/2\hat{e}_k} + \frac{1-\alpha_{n,k}}{2} \psi^n_{\vec{i}-1/2\hat{e}_k} = \psi^n_{\vec{i}}, \tag{5.101}$$

[5] A vertex is a point $(x_i, y_j, z_k)^T$

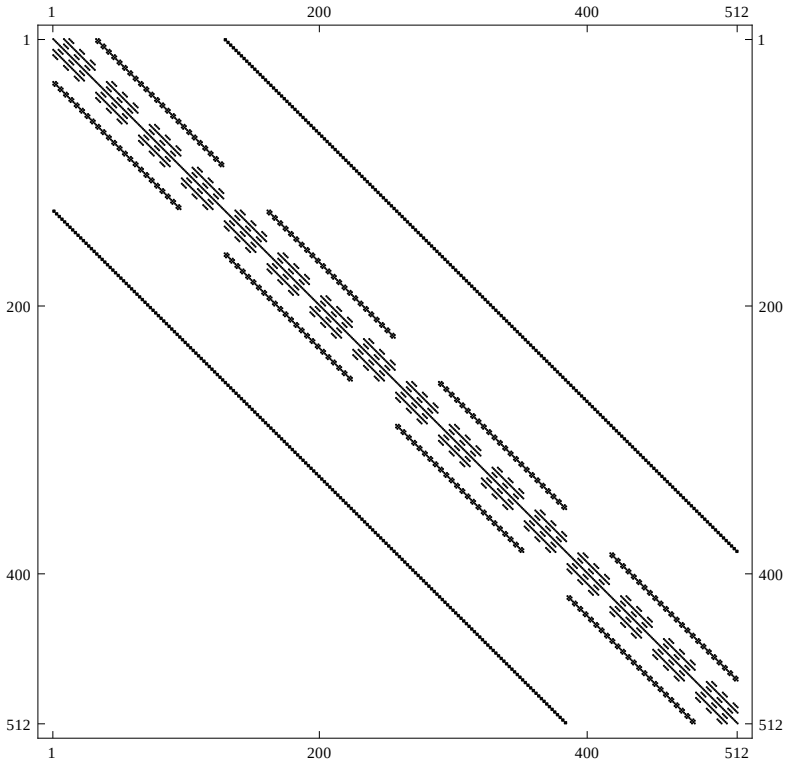


Figure 5.49: Sparsity pattern plot of the AHOTN-1 **B** matrix. The matrix has rank 485.

where indices indicating the spatial moment order are omitted because it is understood that all appearing face/volume fluxes are averages. To leading order, Eq. 5.101 becomes:

$$\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k} = \psi^{n,[0]}_{\vec{i}}. \tag{5.102}$$

In the thick diffusion limit the AHOTN-0 method looks exactly like the *Step* method (identical to DGC-0, DGLA-0) and [7] already reported that the *Step* method does not possess the diffusion limit. However, we shall continue as it may be instructive to complete the analysis. For positive and negative $\mu_{n,k}$ we can derive the difference of the two opposing face fluxes in cell $\vec{i}$ as:

$$\begin{aligned}
&\underline{\mu_{n,k} > 0:} \\
&\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k} - \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k} = \psi^{n,[0]}_{\vec{i}} - \psi^{n,[0]}_{\vec{i}-\hat{e}_k} \\
&\underline{\mu_{n,k} < 0:} \\
&\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_k} - \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_k} = \psi^{n,[0]}_{\vec{i}} - \psi^{n,[0]}_{\vec{i}+\hat{e}_k}.
\end{aligned} \tag{5.103}$$

Substitution into Eq. 5.66 and separation of the sums into a positive and negative $\mu_{n,k}$ contribution gives:

$$\sum_{k=x,y,z}\left[\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\psi_{\vec{i}}^{n,[0]}-\psi_{\vec{i}-\hat{e}_k}^{n,[0]}\right)+\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\psi_{\vec{i}}^{n,[0]}-\psi_{\vec{i}+\hat{e}_k}^{n,[0]}\right)\right]=0. \tag{5.104}$$

Using the definition of ρ and the isotropy of the leading order angular fluxes gives:

$$\rho\sum_{k=x,y,z}\left(2\phi_{\vec{i}}^{[0]}-\phi_{\vec{i}-\hat{e}_k}^{[0]}-\phi_{\vec{i}+\hat{e}_k}^{[0]}\right)=0. \tag{5.105}$$

Equation 5.105 defines a $\mathbf{B}$ matrix that is well known from discretization of the negative Laplacian operator using a central difference. The resulting equation is not rank deficient and therefore AHOTN-0 cannot have the thick diffusion limit.

Analysis of the LN and LL Methods

The analysis of the LN and LL methods shall start with looking at the asymptotic limit of the WDD relations. It will be shown that LL and LN, which only differ in the WDD relations, are equivalent in the thick diffusion limit. The WDD may be stated as follows:

$$\begin{aligned}
&\underline{\text{LN :}}\\
&\quad\frac{1+\alpha_{n,0}}{2}\bar{\psi}_{\vec{i}+1/2\hat{e}_k}^{n}+\frac{1-\alpha_{n,0}}{2}\bar{\psi}_{\vec{i}-1/2\hat{e}_k}^{n}=\bar{\psi}_{\vec{i}}^{n}+3s_{\mu_{n,k}}\left|\alpha_{n,0}\right|\psi_{\vec{i},\hat{e}_k}^{n}\\
&\quad\frac{1+\alpha_{n,1}}{2}\psi_{\vec{i}+1/2\hat{e}_k,m^k}^{n}+\frac{1-\alpha_{n,0}}{2}\psi_{\vec{i}-1/2\hat{e}_k,m^k}^{n}=\psi_{\vec{i},\vec{m}}^{n}\\
&\underline{\text{LL :}}\\
&\quad\frac{1+\alpha_{n,0}}{2}\bar{\psi}_{\vec{i}+1/2\hat{e}_k}^{n}+\frac{1-\alpha_{n,0}}{2}\bar{\psi}_{\vec{i}-1/2\hat{e}_k}^{n}=\bar{\psi}_{\vec{i}}^{n}+3s_{\mu_{n,k}}\left|\alpha_{n,0}\right|\psi_{\vec{i},\hat{e}_k}^{n}\\
&\quad\frac{1+\alpha_{n,1}}{2}\psi_{\vec{i}+1/2\hat{e}_k,m^k}^{n}+\frac{1-\alpha_{n,0}}{2}\psi_{\vec{i}-1/2\hat{e}_k,m^k}^{n}=\psi_{\vec{i},\vec{m}}^{n}+\mathcal{O}\left(\epsilon\right),
\end{aligned} \tag{5.106}$$

where $\bar{\psi}$ indicates the face/volume average. The first (i.e. the average) LL and LN WDD equations are identical, while the two first order moment equations differ by a term that is $\mathcal{O}\left(\epsilon\right)$.

The ϵ multiplying the second summand in Eq. 5.106 originates from the inverse of the optical thickness in its scaled version:

$$t_{n,k}=\frac{\sigma_t\Delta_k}{\mu_{n,k}\epsilon}. \tag{5.107}$$

Using the asymptotic limits of the weights $\alpha_{n,0}$ and $\alpha_{n,1}$:

$$\alpha_{n,l} = \mathrm{sgn}\left(\mu_{k,n}\right)\left[1 + \mathcal{O}\left(\epsilon\right)\right] \text{ for } l = 0, 1, \tag{5.108}$$

the WDD relations of the LL and LN methods are identical to leading order and can be stated as:

$$\begin{aligned} \bar{\psi}^{n,[0]}_{\vec{i}\pm 1/2\hat{e}_k} &= \bar{\psi}^{n,[0]}_{\vec{i}} \pm 3\psi^{n}_{\vec{i},\hat{e}_k} \\ \psi^{n,[0]}_{\vec{i}\pm 1/2\hat{e}_k,m^k} &= \psi^{n,[0]}_{\vec{i},\vec{m}}, \end{aligned} \tag{5.109}$$

where $\pm \to +$ if $\mu_{n,k} > 0$ and $\pm \to -$ if $\mu_{n,k} < 0$.

For the sake of clarity, the derivation of the $\mathbf{B}$ matrix entries originating from the balance equation $\vec{m} = (0,0,0)^T$ and first moment equations $\vec{m} = \hat{e}_k$ starting all from Eq. 5.66 is discussed separated. This results from the WDD relations, which differ significantly in form for the face averages and face moments.

Balance Equation:

The leading order balance equation can be written as:

$$\sum_{k=x,y,z}\sum_{n=1}^{N} w_n \frac{\mu_{n,k}}{\Delta_k}\left(\bar{\psi}^{n,[0]}_{\vec{i}+1/2\hat{e}_k} - \bar{\psi}^{n,[0]}_{\vec{i}-1/2\hat{e}_k}\right) = 0, \tag{5.110}$$

and the sum over angles n can furthermore be split into two parts:

$$\begin{aligned} &\sum_{k=x,y,z}\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\bar{\psi}^{n,[0]}_{\vec{i}+1/2\hat{e}_k} - \bar{\psi}^{n,[0]}_{\vec{i}-1/2\hat{e}_k}\right) \\ + &\sum_{k=x,y,z}\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\bar{\psi}^{n,[0]}_{\vec{i}-1/2\hat{e}_k} - \bar{\psi}^{n,[0]}_{\vec{i}+1/2\hat{e}_k}\right) = 0 \end{aligned} \tag{5.111}$$

The WDD equation for the face averages, Eq. 5.109, and the corresponding equation for the upstream cell are now used to obtain:

$$\bar{\psi}^{n,[0]}_{\vec{i}\pm 1/2\hat{e}_k} - \bar{\psi}^{n,[0]}_{\vec{i}\mp 1/2\hat{e}_k} = \left(\bar{\psi}^{n,[0]}_{\vec{i}} - \bar{\psi}^{n,[0]}_{\vec{i}\mp\hat{e}_k}\right) \pm 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} - \psi^{n,[0]}_{\vec{i}\mp\hat{e}_k,\hat{e}_k}\right). \tag{5.112}$$

Substitution of Eq. 5.112 into Eq. 5.111 gives:

$$\sum_{k=x,y,z}\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left[\left(\bar{\psi}^{n,[0]}_{\vec{i}} - \bar{\psi}^{n,[0]}_{\vec{i}-\hat{e}_k}\right) + 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} - \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\hat{e}_k}\right)\right]$$
$$+ \sum_{k=x,y,z}\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left[\left(\bar{\psi}^{n,[0]}_{\vec{i}} - \bar{\psi}^{n,[0]}_{\vec{i}+\hat{e}_k}\right) - 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} - \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\hat{e}_k}\right)\right] = 0. \quad (5.113)$$

Since the leading order volume fluxes are isotropic, the angular flux moments can be replaced by scalar flux moments:

$$\left[\sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\right]\left[\sum_{k=x,y,z}\left(\bar{\phi}^{[0]}_{\vec{i}} - \bar{\phi}^{[0]}_{\vec{i}-\hat{e}_k}\right) + 3\left(\phi^{[0]}_{\vec{i},\hat{e}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\hat{e}_k}\right)\right]$$
$$+ \left[\sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\right]\left[\sum_{k=x,y,z}\left(\bar{\phi}^{[0]}_{\vec{i}} - \bar{\phi}^{[0]}_{\vec{i}+\hat{e}_k}\right) - 3\left(\phi^{[0]}_{\vec{i},\hat{e}_k} - \phi^{[0]}_{\vec{i}+\hat{e}_k,\hat{e}_k}\right)\right] = 0. \quad (5.114)$$

Using the definition of ρ_k, Eq. 5.78, and its independence from k for the particular case considered, Eq. 5.81, Eq. 5.114 is rewritten as:

$$\sum_{k=x,y,z}\left[\left(2\bar{\phi}^{[0]}_{\vec{i}} - \bar{\phi}^{[0]}_{\vec{i}-\hat{e}_k} - \bar{\phi}^{[0]}_{\vec{i}+\hat{e}_k}\right) + 3\left(\phi^{[0]}_{\vec{i}+\hat{e}_k,\hat{e}_k} - \phi^{[0]}_{\vec{i}-\hat{e}_k,\hat{e}_k}\right)\right] = 0. \quad (5.115)$$

From expression Eq. 5.115, all rows of matrix $\mathbf{B}$ can be constructed that correspond to $\vec{m} = (0,0,0)^T$.

First Order Moment Equation:

The first order moment equations up to leading order for $\vec{m} = \hat{e}_k$ are given by:

$$\sum_{l:l\neq k}\left[\sum_{\mu_{n,l}>0} w_n \frac{|\mu_{n,l}|}{\Delta_l}\left(\psi^{n,[0]}_{\vec{i}+1/2\hat{e}_l,\vec{m}^k} - \psi^{n,[0]}_{\vec{i}-1/2\hat{e}_l,\vec{m}^k}\right)\right.$$
$$\left. + \sum_{\mu_{n,l}<0} w_n \frac{|\mu_{n,l}|}{\Delta_l}\left(\psi^{n,[0]}_{\vec{i}-1/2\hat{e}_l,\vec{m}^k} - \psi^{n,[0]}_{\vec{i}+1/2\hat{e}_l,\vec{m}^k}\right)\right]$$
$$+ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\bar{\psi}^{n,[0]}_{\vec{i}+1/2\hat{e}_k} + \bar{\psi}^{n,[0]}_{\vec{i}-1/2\hat{e}_k}\right)$$
$$- \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left(\bar{\psi}^{n,[0]}_{\vec{i}-1/2\hat{e}_k} + \bar{\psi}^{n,[0]}_{\vec{i}+1/2\hat{e}_k}\right) = 0. \quad (5.116)$$

For replacing the face fluxes in Eq. 5.116 with volume fluxes, Eqs. 5.109 applied for cell $\vec{i}$ and the corresponding upstream cell are used:

$$\begin{aligned}\psi^{n,[0]}_{\vec{i}\pm 1/2\hat{e}_l,\vec{m}^k} - \psi^{n,[0]}_{\vec{i}\mp 1/2\hat{e}_l,\vec{m}^k} &= \psi^{n,[0]}_{\vec{i},\vec{m}} - \psi^{n,[0]}_{\vec{i}\mp\hat{e}_l,\vec{m}} \\ \bar{\psi}^{n,[0]}_{\vec{i}\pm 1/2\hat{e}_k} + \bar{\psi}^{n,[0]}_{\vec{i}\mp 1/2\hat{e}_k} &= \left(\bar{\psi}^{n,[0]}_{\vec{i}} + \bar{\psi}^{n,[0]}_{\vec{i}\mp\hat{e}_k}\right) \pm 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} + \psi^{n,[0]}_{\vec{i}\mp\hat{e}_k,\hat{e}_k}\right), \end{aligned} \tag{5.117}$$

where the sign convention is identical to the one introduced in Eq. 5.109. Substituting Eq. 5.117 into Eq. 5.116 then yields:

$$\begin{aligned} &\sum_{l:l\neq k}\left[\sum_{\mu_{n,l}>0} w_n \frac{|\mu_{n,l}|}{\Delta_l}\left(\psi^{n,[0]}_{\vec{i},\vec{m}} - \psi^{n,[0]}_{\vec{i}-\hat{e}_l,\vec{m}}\right)\right. \\ &+ \left.\sum_{\mu_{n,l}<0} w_n \frac{|\mu_{n,l}|}{\Delta_l}\left(\psi^{n,[0]}_{\vec{i},\vec{m}^k} - \psi^{n,[0]}_{\vec{i}+\hat{e}_l,\vec{m}^k}\right)\right] \\ &+ \sum_{\mu_{n,k}>0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left[\left(\bar{\psi}^{n,[0]}_{\vec{i}} + \bar{\psi}^{n,[0]}_{\vec{i}-\hat{e}_k}\right) + 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} + \psi^{n,[0]}_{\vec{i}-\hat{e}_k,\hat{e}_k}\right)\right] \\ &- \sum_{\mu_{n,k}<0} w_n \frac{|\mu_{n,k}|}{\Delta_k}\left[\left(\bar{\psi}^{n,[0]}_{\vec{i}} + \bar{\psi}^{n,[0]}_{\vec{i}+\hat{e}_k}\right) - 3\left(\psi^{n,[0]}_{\vec{i},\hat{e}_k} + \psi^{n,[0]}_{\vec{i}+\hat{e}_k,\hat{e}_k}\right)\right] = 0. \end{aligned} \tag{5.118}$$

Now the leading order angular flux moments are replaced by scalar flux moments, which can be pulled out of the summation over angular direction; then the summation over angles is performed using the definition of $\rho_k = \rho$. Noting that ρ for the considered case is independent of k the, it can be canceled and the final expression for Eq. 5.119 becomes:

$$\begin{aligned} &\sum_{l:l\neq k}\left(2\phi^{[0]}_{\vec{i},\vec{m}} - \phi^{[0]}_{\vec{i}+\hat{e}_l,\vec{m}} - \phi^{[0]}_{\vec{i}-\hat{e}_l,\vec{m}}\right) \\ &+ \left[\left(\bar{\phi}^{[0]}_{\vec{i}-\hat{e}_k} - \bar{\phi}^{[0]}_{\vec{i}+\hat{e}_k}\right) + 3\left(2\phi^{[0]}_{\vec{i},\vec{m}} + \phi^{[0]}_{\vec{i}-\hat{e}_k,\vec{m}} + \phi^{[0]}_{\vec{i}+\hat{e}_k,\vec{m}}\right)\right] = 0. \end{aligned} \tag{5.119}$$

From Eq. 5.115 and 5.119, the **B** matrix for the LL/LN method can be constructed. Again, the test case features 4^3 spatial cells and the matrix **B** is of size 256×256. The sparsity pattern is depicted in Fig. 5.50 and the Mathematica notebook that constructs **B** is listed in Appendix E.4. The matrix has has full rank for the LL/LN method such that LL/LN are concluded to not have the thick diffusion limit.

5.3.3 Numerical Experiments using Thick Diffusion Limit Test Case

The test problem described in section 3.6 is used to determine, by numerical experiment, whether the set of numerical methods considered within this work possess the thick diffusion

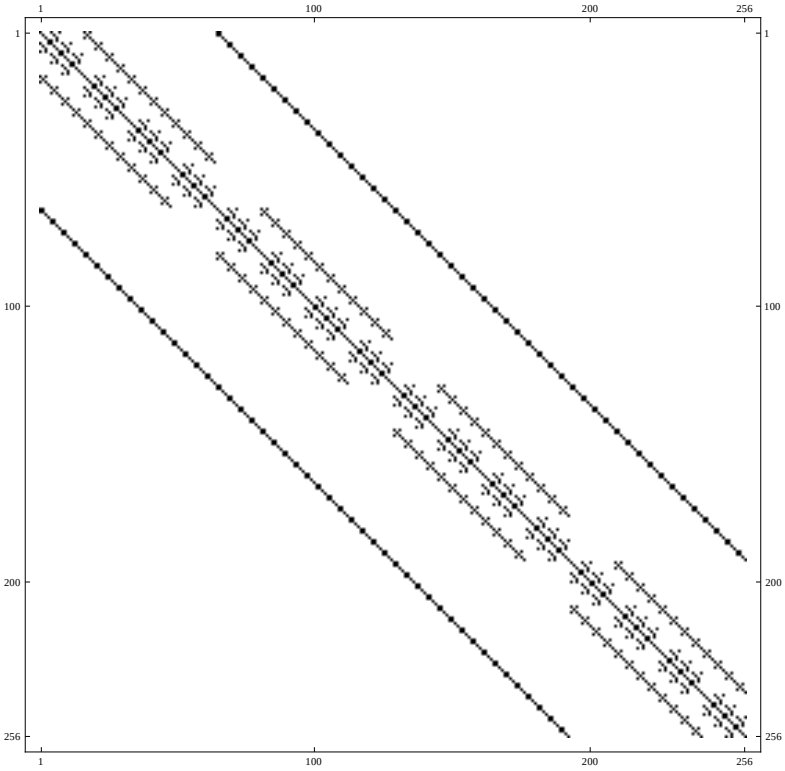


Figure 5.50: Sparsity pattern plot of the LL/LN **B** matrix. The matrix has full rank.

limit. The numerical experiment proceeds by solving the transport problem on a homogeneous cube with cross sections and uniform source:

$$\begin{aligned} \sigma_t &= \frac{1}{\epsilon} \\ \sigma_s &= \frac{1}{\epsilon} - \epsilon \\ Q &= \epsilon. \end{aligned} \tag{5.120}$$

for $\epsilon = 0.1^l,\ l = 1, 2, ..., 5$. Vacuum boundary conditions are applied for the transport solution on all inflow boundaries. For all levels of ϵ, the same mesh spacing along all coordinate axes is used: $\Delta x = \Delta y = \Delta z = 1/25$ cm, and the physical domain thickness is fixed at 1 cm. The solution is obtained using the *GMRES* method described in section 1.2.3 instead of using the source iteration described in 1.2.2 because of the faster iterative convergence when the scattering ratio approaches unity. The source iteration's spectral radius in an infinite medium is the scattering ratio: $c = 1 - \epsilon^2$. GMRES converges faster than SI for the presented test problem, but still suffers from an increase in iteration count. The iterations are stopped when

the residual norm reduces by a factor of 1.0×10^{-12}.

The reference solution for $\epsilon \to 0$ is obtained by solving Eq. 3.52 using a simple finite difference solver for the Diffusion problem. In order to ensure accuracy of the numerical solution of Eq. 3.52, the solution was obtained on a mesh featuring 125 mesh cells per dimensions.

With decreasing ϵ, a method that possesses the diffusion limit will approach the reference solution from above and converge towards it. A method that does not possess the thick diffusion limit will produce a solution that converges to zero as $\epsilon \to 0$. An early indicator that a method does not have the thick diffusion limit is that its scalar flux falls below the reference diffusion solution.

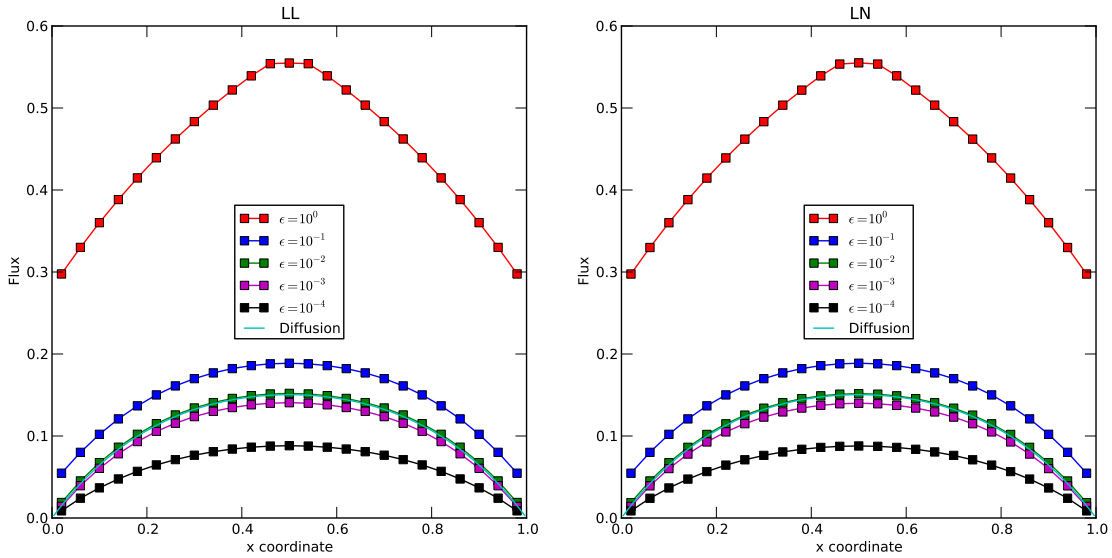


Figure 5.51: Results for thick diffusion limit numerical experiment for the LL and LN method. Neither of these two discretization methods has the thick diffusion limit.

In Figs. 5.51 through 5.55 results for the thick diffusion limit test problem for LL and LN, AHOTN, HODD, DGLA and DGC are presented, respectively. The numerical results corroborate the conclusions made for those methods whose analysis is performed in [7] or within this work:

- DGC-0,1 and DGLA-0 do not have the diffusion limit.
- AHOTN-1and DGLA-1 have the diffusion limit.
- AHOTN-0, HODD-0, HODD-1, LL and LN do not possess the diffusion limit.

In addition, numerical results are obtained for higher-order methods for which analysis is not available. The AHOTN and DGLA methods possess the thick diffusion limit for all orders $\Lambda > 0$, while none of the other methods possesses the thick diffusion limit. Note, that no results are presented for the SCB method, because from [7] it is known that it does feature the thick diffusion limit. As a corollary of the lack of the diffusion limit for both the AHOTN-0 and the *Step* method the SCT-Step method is concluded to not possess a diffusion limit.

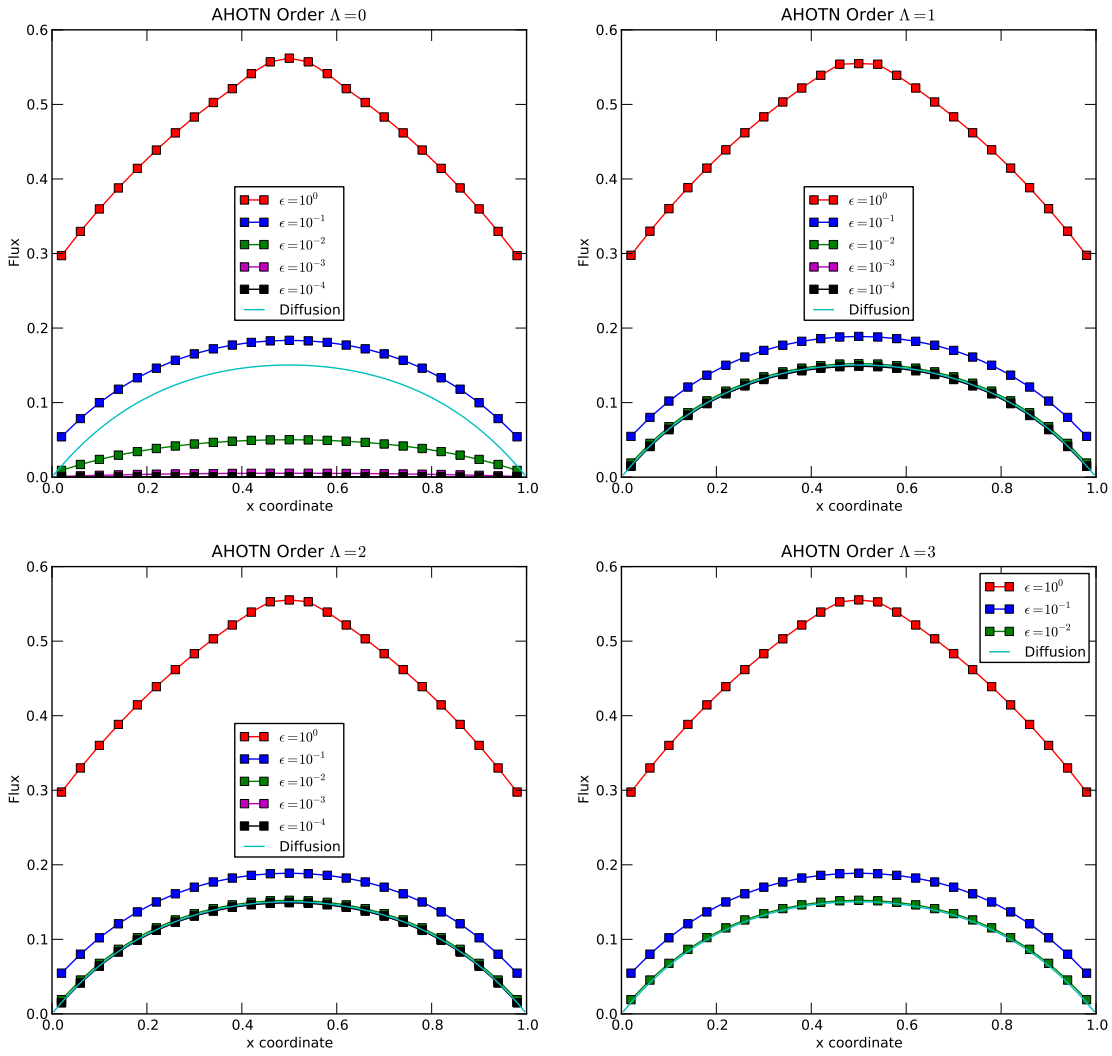


Figure 5.52: Results for thick diffusion limit numerical experiment for AHOTN of orders zero through three. Except for AHOTN-0, all other methods possess the diffusion limit.

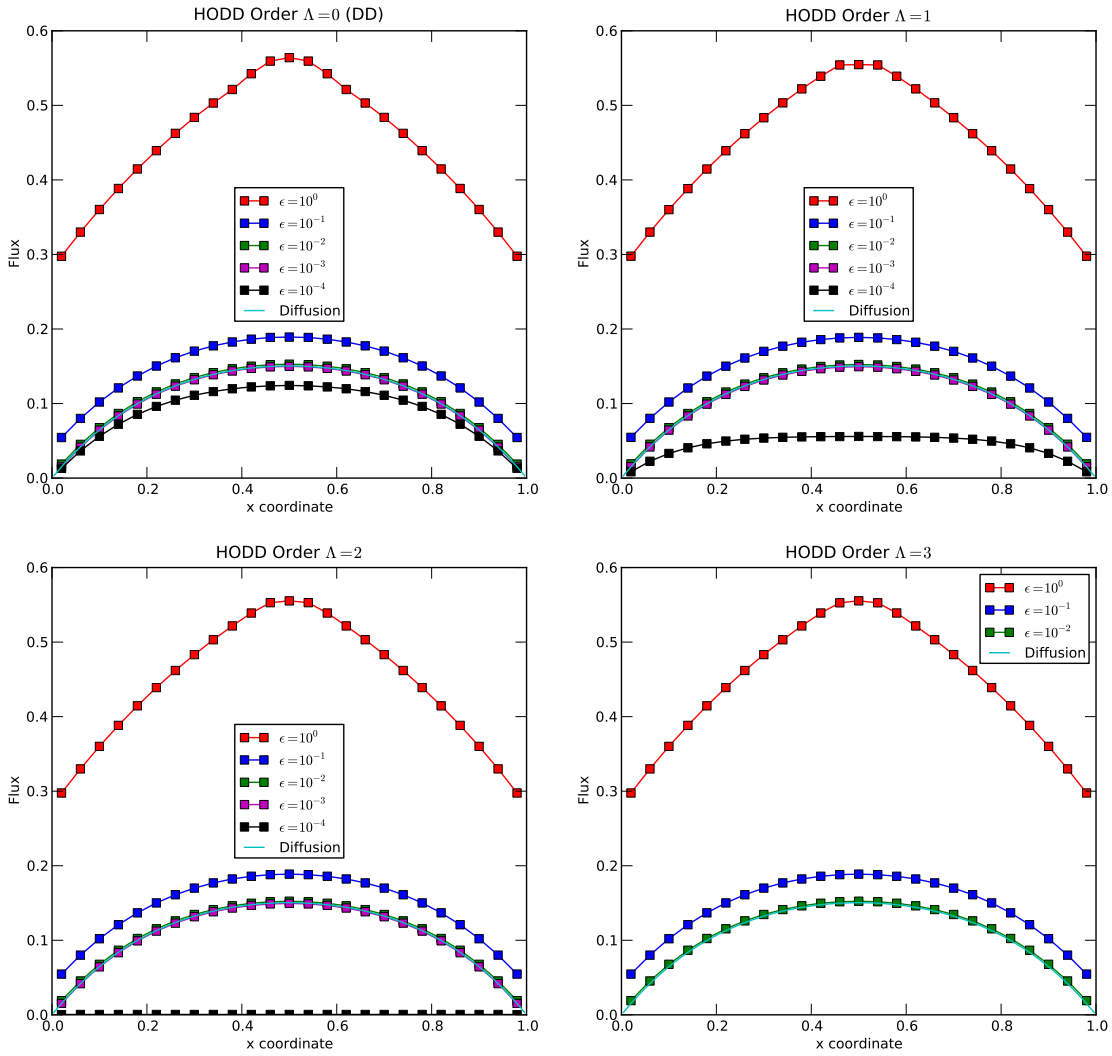


Figure 5.53: Results for thick diffusion limit numerical experiment for HODD of orders zero through three. Clearly HODD-0,1,2 do not possess the thick diffusion limit, but for HODD-3 ϵ, could not be decreased far enough to make a definite conclusions.

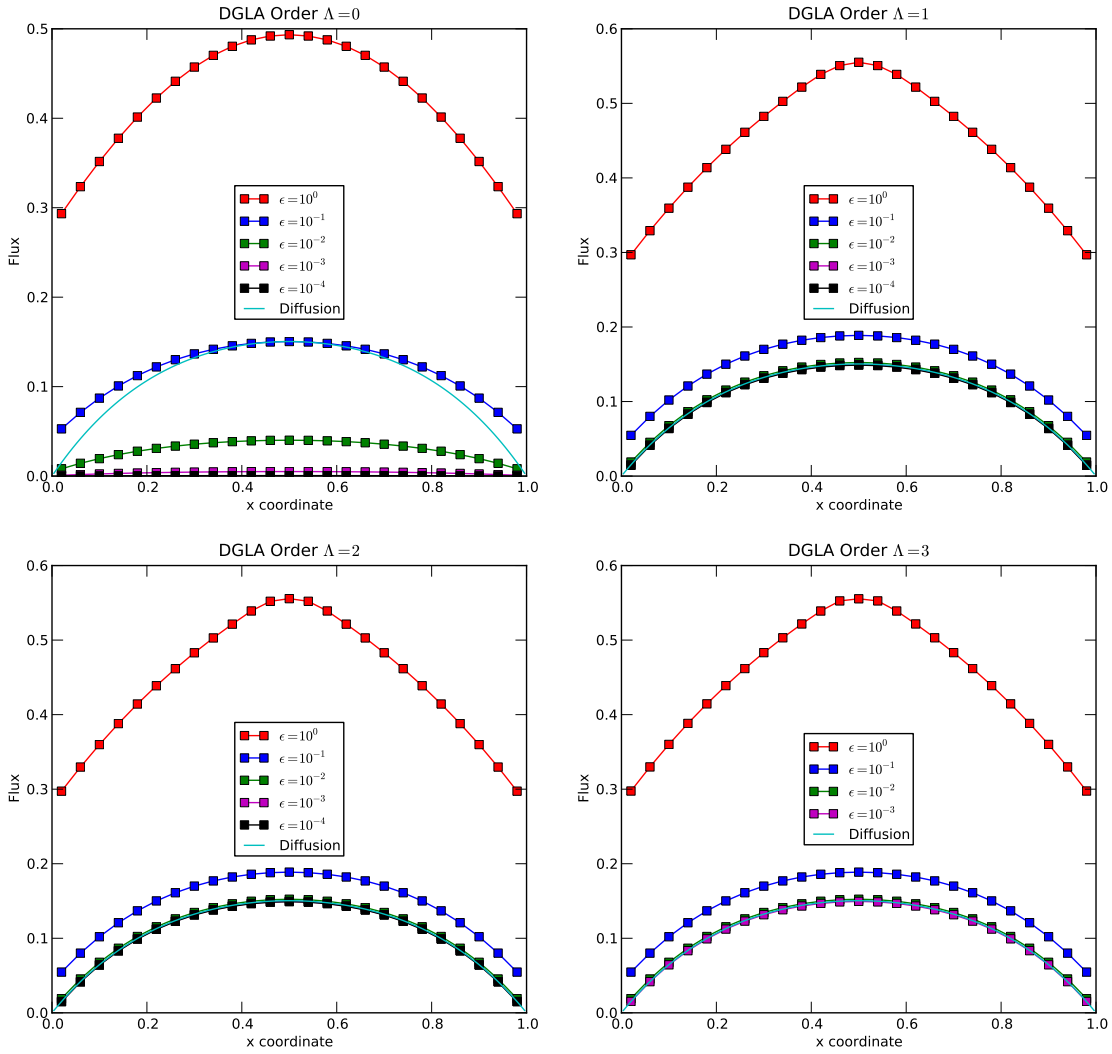


Figure 5.54: Results for thick diffusion limit numerical experiment for DGLA of orders zero through three. Except for DGLA-0 (*Step* method), all DGLA methods feature the thick diffusion limit.

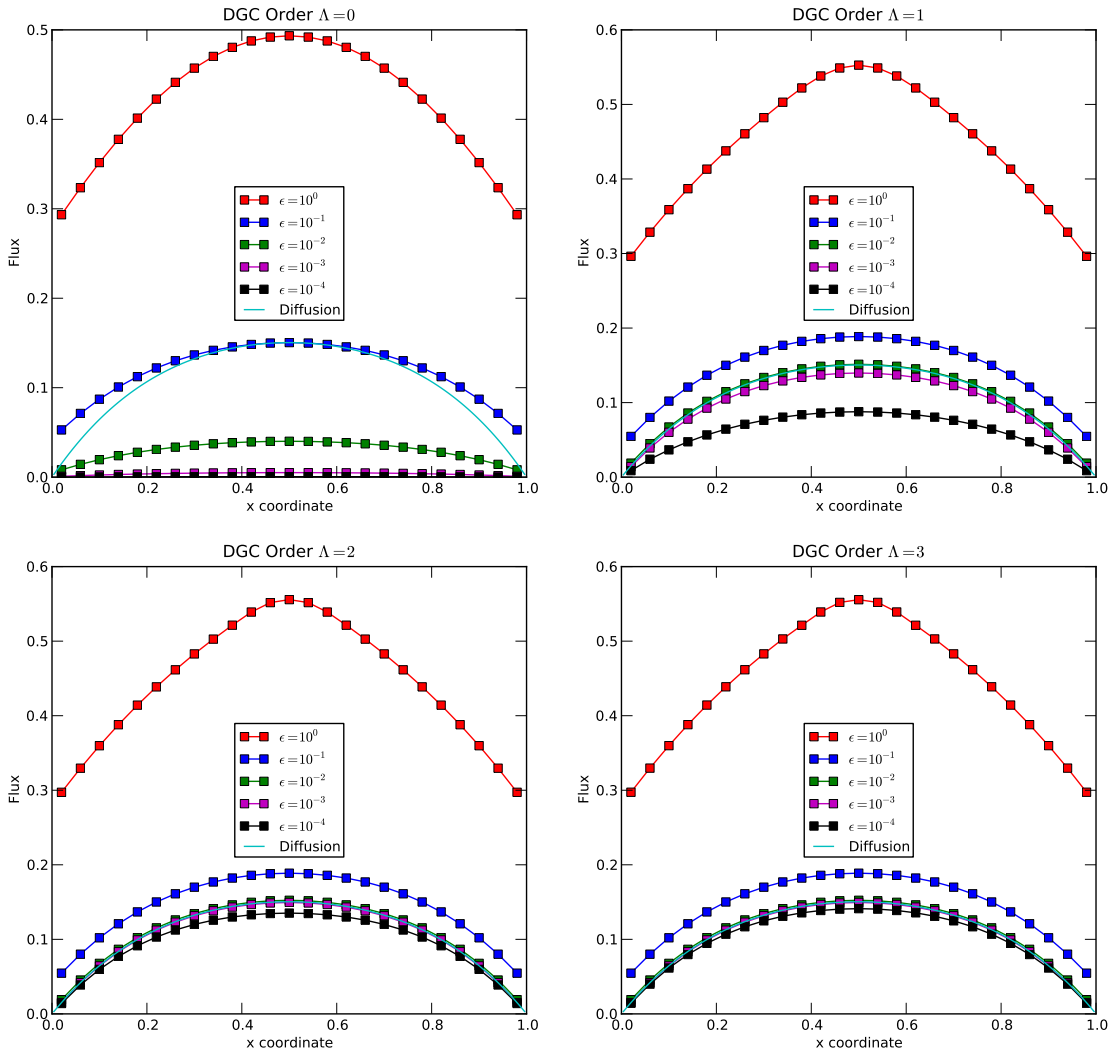


Figure 5.55: Results for thick diffusion limit numerical experiment for DGC of orders zero through three. None of the DGC orders possesses the thick diffusion limit.

5.4 Summary of the Numerical Results

Within chapter 5, numerical results were presented for ranking the properties of the contending spatial discretization methods: AHOTN, DGLA, DGC, and HODD of orders $\Lambda = 0, .., 3$; and SCB, LL and LN (all of order $\Lambda = 1$). In addition, the properties of the new SCT-Step method were investigated. First, accuracy and efficiency were tested based on the MMS test problem described in section 3.2. Second, the resilience of the contending methods against the occurrence of negative angular flux face-averaged and negative scalar flux volume-averaged fluxes was investigated based on Lathrop's test problem, introduced within this work in section 3.4. Finally, analysis motivated by Ref. [7] was performed and supporting numerical experiments based on the test problem described in section 3.4 were conducted to determine the methods that possess the thick diffusive limit.

This section shall summarize the findings of the preceding sections and organize them for the reader in a convenient fashion. In Table 5.5 the efficiency/accuracy results are summarized for the two solution-smoothness classes encountered in applications, namely C_0 and C_1. The utilized symbols have the following meaning:

- $\oplus$: Best performer/performs much better than other methods.
- $+$: Good performance, applicable
- $\odot$: Fair performance/other methods are superior but no detrimental failure.
- $-$: Poor performance, application not recommended.
- $\ominus$: Catastrophic failure precludes method's application.

Further, in Table 5.6, results from Lathrop's test problems for determining resilience of the discretization methods against negative fluxes are summarized. Finally, in Table 5.7, results from section 5.3 pertaining to the thick diffusion limit are summarized.

Table 5.5: Summary of the efficiency/accuracy results obtained within this work using the MMS suite for various discretization methods.

Efficiency/Accuracy					
	C0 Smoothness			C1 Smoothness	
	L_p	L_∞	Integral	L_p [1]	Integral
AHOTN-1	$-$	$\ominus$	$\odot$	$\odot$	$\odot$
AHOTN-2	$-$	$\ominus$	$\odot$	$+$	$\odot$
AHOTN-3	$-$	$\ominus$	$\odot$	$\oplus$	$\odot$
DD	$-$	$\ominus$	$-$	$-$	$-$
HODD-1	$-$	$\ominus$	$-$	$-$	$-$
HODD-2	$-$	$\ominus$	$-$	$-$	$-$
HODD-3	$-$	$\ominus$	$-$	$-$	$-$
DGLA-1	$-$	$\ominus$	$\odot$	$-$	$\odot$
DGLA-2	$-$	$\ominus$	$\odot$	$+$	$\odot$
DGLA-3	$-$	$\ominus$	$\odot$	$\oplus$	$\odot$
DGC-1	$-$	$\ominus$	$\odot$	$\odot$	$\odot$
DGC-2	$-$	$\ominus$	$\odot$	$+$	$\odot$
DGC-3	$-$	$\ominus$	$\odot$	$\oplus$	$\odot$
LL	$-$	$\ominus$	$+$	$\odot$	$+$
LN	$-$	$\ominus$	$+$	$\odot$	$+$
SCB	$-$	$\ominus$	$-$	$-$	$-$
SCT-Step	$\oplus$	$\oplus$	$-$	$-$	$-$

[1] Includes $p = 1, 2, \infty$

Table 5.6: Summary of the resilience against negative fluxes for various discretization methods determined in this work using Lathrop's test problem.

Resilience against negative fluxes		
	τ_ψ^w	τ_ϕ^w
AHOTN-0	$\odot$	$+$
AHOTN-1	$-$	$-$
AHOTN-2	$+$	$+$
AHOTN-3	$-$	$-$
HODD-0	$\ominus$	$\ominus$
HODD-1	$-$	$-$
HODD-2	$-$	$-$
HODD-3	$-$	$-$
DGLA-0	$\oplus$	$\oplus$
DGLA-1	$-$	$-$
DGLA-2	$+$	$+$
DGLA-3	$-$	$-$
DGC-1	$-$	$-$
DGC-2	$+$	$+$
DGC-3	$-$	$-$
LL	$-$	$-$
LN	$-$	$-$
SCB	$\oplus$	$\oplus$

Table 5.7: Synopsis of the analysis and numerical experiments related to the possession of the thick diffusion limit. For the SCT-Step method the results are marked as extrapolated because neither of the basic discretization methods possesses the diffusion limit.

Method	Order	Analysis	Numerical Results
DGLA	0	no[7]	no
	1	yes[7]	yes
	2	n.a.	yes
	3	n.a.	yes
DGC	0	no[7]	no
	1	no[7]	no
	2	n.a.	no
	3	n.a.	no
SCB	1	yes[7]	n.A.
HODD	0	no	no
	1	no	no
	2	n.a.	no
	3	n.a.	no
AHOTN	0	no	no
	1	yes	yes
	2	n.a.	yes
	3	n.a.	yes
LL	1	no	no
LN	1	no	no
SCT-Step	0	no (extr.)	no(extr.)

Chapter 6

Development of a Quantitative Decision Metric

This chapter is dedicated to the development of a quantitative decision metric that is based on the data obtained from the MMS test problem described in 3.2 and Lathrop's test problem described in 3.4. This is followed by a validation exercise for the resulting metric that is derived from the one energy group NEA box-in-box benchmark problem described in [67] and [68]. The approach taken in this chapter is to first predict the contending methods' performance utilizing the developed decision metric and the data obtained in chapter 5, then to use the contending methods to solve selected cases adapted from the NEA box-in-box benchmark suite. The validation is then completed by comparing the predicted and computed decision metric's results.

It should be stressed that the computed values of the decision metric are only meaningful in relation to the values associated with other contending methods, i.e. relative to one another. The thick diffusion limit property shall be excluded from the presented validation exercise because the selected NEA box-in-box suite does not include any challenge relating to the thick diffusive regime. However, we will present some ideas on how to add the diffusion limit property to the decision metric, if desired.

Within this chapter, the NEA box-in-box benchmark suite is described in section 6.1, subsequently the decision metric is developed in section 6.2, and finally the decision metric is exercised and validated in section 6.3.

6.1 NEA Box-In-Box Benchmark Suite

The single energy group NEA box-in-box benchmark problem is depicted in Fig. 6.1. It consists of a box of dimensions $1 \times 1 \times L$ cm completely enclosing a smaller box of dimension $\gamma \times \gamma \times \gamma L$ cm. The volume inside the smaller box is referred to as region II, while the volume inside the

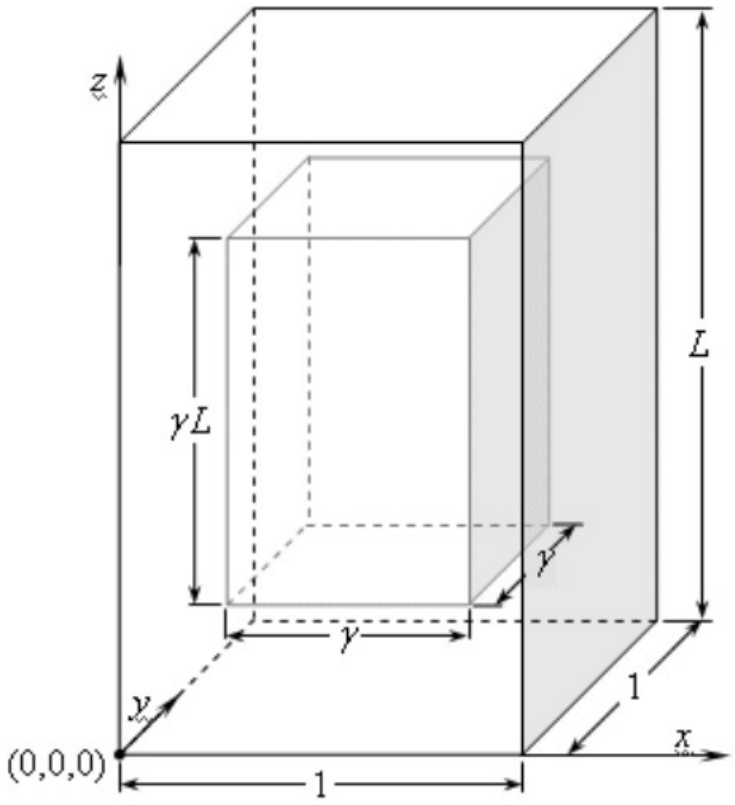


Figure 6.1: Schematic of the NEA box-in-box benchmark suite. Image courtesy of Y. Azmy[67].

enclosing box but outside the smaller box is referred to as region I. The geometric and material properties of regions I and II are allowed to differ independently.

The length L controls the height of the spatial domain and since the domain's size in the x and y direction is constant, it also controls the domain's aspect ratio. The second parameter, γ, controls the size of region II compared to the size of region I. Finally, for each region, the total cross section and scattering ratio can be varied independently. Thus, six parameters can be varied independently of one another, which creates a suite of benchmark problems rather than a single test problem. In [67], 729 variations of the box-in-box suite are considered, created by all possible combinations of the parameter values listed in Table 6.1.

The flux in the box-in-box suite is driven by a source located in a box ranging from $\vec{r} = (0,0,0)^T$ to $\vec{r} = ((1-\gamma)/2, (1-\gamma)/2, L(1-\gamma)/2)$: the box is completely encompassed in region I and extends from the origin to the lower left corner of the smaller box defining region II. The

Table 6.1: Parameter variations of NEA box-in-box benchmark suite utilized in [67].

Parameter	Range		
L	0.1	1.0	5.0
γ	0.1	0.5	0.9
$\sigma_{t,I}$	0.1	1.0	5.0
c_I	0.5	0.8	1.0
$\sigma_{t,II}$	0.1	1.0	5.0
c_{II}	0.5	0.8	1.0

source is spatially uniform and satisfies the normalization condition:

$$\int_0^{(1-\gamma)/2} dx \int_0^{(1-\gamma)/2} dy \int_0^{L(1-\gamma)/2)} dz \; S\left(\vec{r}\right) = \frac{\bar{S}L(1-\gamma)^3}{8} = 1, \tag{6.1}$$

where $\bar{S}$ is the average source within the defined source volume.

The NEA box-in-box benchmark suite is designed to test the accuracy of integral quantities computed by participating discretization methods[67]. These integral quantities can be divided into three categories: (1) the average scalar fluxes in regions I and II, (2) currents over several surfaces in the suite domain, and (3) finally average scalar fluxes over several subvolumes in the domain. The NEA box-in-box benchmark suite is an excellent benchmark problem to examine the performance of the decision metric over a range in parameter space and not just a single benchmark configuration.

Within this work, only quantity types one and three are considered, while currents are excluded from the discussion. The reason for this omission is that the MMS test suite was used to obtain discretization errors only related to angular and scalar fluxes because of the limitation of quantities that the MMS3D code can compute. This data is instrumental for the prediction of the method's accuracy and efficiency in the framework of the decision metric. Therefore, predictions of errors associated with currents cannot be made because the necessary data is not readily available. However, this is not a fundamental limitation of our approach and may well constitute future research.

Further, both considered types of quantities, the region and subvolume average fluxes, are integral quantities and their accuracy is therefore naturally measured in an integral error norm. The NEA box-in-box benchmark suite does not provide an analytical or mesh-cell wise reference solution as the MMS test suite does, because, in contrast to the MMS test problem, these are unknown and virtually impossible to obtain. Therefore, the validation of the performance metric will be restricted to accuracy in computing integral quantities. In many ways, this

reflects the typical interest of radiation transport code practitioners, e.g. in reactor design where pin-averaged fission rates are desired. A total of 15 integral (average) target quantities listed in Table 6.2 are computed when solving each considered instance of the NEA benchmark suite.

Table 6.2: Target subvolumes for which averaged scalar fluxes are computed the framework of the NEA box-in-box suite.

Quantity ID	Lower left corner	Upper right corner
1.a	Region I	
1.b	Region II	
3.a	$\left(\frac{1-\gamma}{4}, \frac{1-\gamma}{4}, L\frac{1-\gamma}{4}\right)^T$	$\left(\frac{1-\gamma}{2}, \frac{1-\gamma}{2}, L\frac{1-\gamma}{2}\right)^T$
3.b	$\left(\frac{1-\gamma}{2}, \frac{1-\gamma}{2}, L\frac{1-\gamma}{2}\right)^T$	$\left(\frac{1}{2}, \frac{1}{2}, \frac{L}{2}\right)^T$
3.c	$\left(\frac{1}{2}, \frac{1}{2}, \frac{L}{2}\right)^T$	$\left(\frac{1+\gamma}{2}, \frac{1+\gamma}{2}, L\frac{1+\gamma}{2}\right)^T$
3.d	$\left(\frac{1+\gamma}{2}, \frac{1+\gamma}{2}, L\frac{1+\gamma}{2}\right)^T$	$\left(\frac{3+\gamma}{4}, \frac{3+\gamma}{4}, L\frac{3+\gamma}{4}\right)^T$
3.e	$\left(\frac{3+\gamma}{4}, 0, 0\right)^T$	$\left(1, \frac{1-\gamma}{4}, L\frac{1-\gamma}{4}\right)^T$
3.f	$\left(\frac{3+\gamma}{4}, \frac{3+\gamma}{4}, 0\right)^T$	$\left(1, 1, L\frac{1-\gamma}{4}\right)^T$
3.g	$\left(0, 0, L\frac{3+\gamma}{4}\right)^T$	$\left(\frac{1-\gamma}{4}, \frac{1-\gamma}{4}, L\right)^T$
3.h	$\left(\frac{3+\gamma}{4}, 0, L\frac{3+\gamma}{4}\right)^T$	$\left(1, \frac{1-\gamma}{4}, L\right)^T$
3.i	$\left(\frac{3+\gamma}{4}, \frac{3+\gamma}{4}, L\frac{3+\gamma}{4}\right)^T$	$(1, 1, L)^T$
3.j	$\left(\frac{1}{2}, \frac{1-\gamma}{2}, L\frac{1-\gamma}{2}\right)^T$	$\left(\frac{1+\gamma}{2}, \frac{1}{2}, \frac{L}{2}\right)^T$
3.k	$\left(\frac{1}{2}, \frac{1}{2}, L\frac{1-\gamma}{2}\right)^T$	$\left(\frac{1+\gamma}{2}, \frac{1+\gamma}{2}, \frac{L}{2}\right)^T$
3.l	$\left(\frac{1-\gamma}{2}, \frac{1-\gamma}{2}, \frac{L}{2}\right)^T$	$\left(\frac{1}{2}, \frac{1}{2}, L\frac{1+\gamma}{2}\right)^T$
3.m	$\left(\frac{1}{2}, \frac{1-\gamma}{2}, \frac{L}{2}\right)^T$	$\left(\frac{1+\gamma}{2}, \frac{1}{2}, L\frac{1+\gamma}{2}\right)^T$

Reference solutions for the box-in-box test problem were obtained by two means: fine mesh TORT[65] and MCNP reference solutions[68]. Thus, the discretization error in the problem posed by Azmy consists of the spatial discretization error and the angular discretization error. The Monte-Carlo MCNP solution only comprises statistical uncertainty and beyond this

limitation is free of error, while TORT solutions comprise a discretization error that encompasses some contribution from the angular quadrature and some spatial discretization error. For comparison of spatial discretization methods, the angular discretization error is an unwanted artifact because it contaminates the quantity of interest for our purposes: the spatial discretization error.

However, the error from the angular quadrature is important beyond being just an annoyance for the applicability of the decision metric. Consider an error norm applied to the numerically computed scalar flux, $\phi^{h,N}$:

$$\|\epsilon\| = \|\phi^{h,N} - \phi\|, \tag{6.2}$$

where ϕ is the exact scalar flux. Now, consider the scalar flux ϕ^N, which comprises only an angular discretization error. It can, for example, be obtained by letting $h \to 0$ while fixing the angular quadrature. Then Eq. 6.2 can be written as:

$$\|\epsilon\| = \|(\phi^N - \phi) + (\phi^{h,N} - \phi^N)\| \leq \|\phi^N - \phi\| + \|\phi^{h,N} - \phi^N\|. \tag{6.3}$$

The first summand in Eq. 6.3 is the angular discretization error, while the second summand is the spatial discretization error. In order to obtain the correct solution $\|\epsilon\| \to 0$ both the spatial mesh and angular quadrature need to be refined. In case only the spatial mesh is refined the angular discretization error will dominate the total discretization error and, at least from the solution accuracy perspective, it is inconsequential which spatial discretization method is utilized.

The discussion of the decision metric that will be introduced in section 6.2 is based on the assumption that the spatial discretization error is much larger than the angular discretization error. It will briefly be discussed how the decision metric can still be of utility if the angular discretization error dominates the spatial discretization error. However, for the remainder of the discussion, it shall be assumed that the spatial discretization error dominates the angular discretization error.

The reference solution employed to quantify the spatial discretization error must reflect the necessity of a dominating spatial discretization error. Two options are available to enable realizing this objective: First, a quadrature that is accurate enough could be chosen for the S_N runs, and the MCNP solutions could be used as reference solutions; or second, an S_N solution that is spatially converged for a fixed angular quadrature is used as reference solution and all contending methods' solutions share that same quadrature. Then in Eq. 6.3 $\phi \leftarrow \phi^N$ and

$$\|\epsilon\| = \|\phi^{h,N} - \phi^N\|.$$

Within this work, the second option is utilized, because level symmetric quadratures that

are sufficiently fine to observe the first option would prolong execution times to impractical levels. In fact, the level-symmetric quadrature might not even be convergent for the NEA box-in-box test problem. The more accurate Legendre-Chebychev quadrature would provide more rapidly converging results, but these were not readily available in the research codes used within this work.

Instead of using TORT computed reference solutions, the AHOTN-1 code used throughout this work is utilized for generating the reference solutions. The reason for this is that the discretization error of several contending methods will be smaller than the TORT reference solutions on fine meshes. This results from the higher expansion order and quality of approximation used by AHOTN-1 compared to the TORT code. In addition, tight stopping criteria of 1.0×10^{-10} have to be set to avoid contamination of the reference solutions with iterative convergence error. Therefore, TORT references created within [68] cannot be used for this work.

The method used for creating high-fidelity reference solutions employed within this work is using AHOTN-1 for a set of five meshes featuring up to 256^3 mesh cells. The finest mesh from the corresponding validation exercise runs will be at least a factor of four coarser. In addition, Richardson extrapolation[69] is used to improve the accuracy of the reference solution. For the described case, Richardson extrapolation takes the following form:

0. Mesh spacing values: $h_j, j = 1, ...5$
1. Computed region averages: $\bar{\phi}_j, j = 1, ...5$
2. Compute errors: $e_j = \left|\bar{\phi}_j - \bar{\phi}_5\right|, j = 1, ..., 4$
3. Least-squares fit of $\{h_j,\ e_j\}_{j=1,...,4}$ to $e(h) = Ch^p \Rightarrow \log(e) = \log(C) + p\log(h)$
4. Richardson extrapolation: $\bar{\phi}^R = \dfrac{2^p\bar{\phi}_5 - \bar{\phi}_4}{2^p - 1},$ (6.4)

where $\bar{\phi}^R$ is the Richardson extrapolate that is used as reference value for the corresponding benchmark quantity.

The final limitation of the NEA box-in-box benchmark suite is that it only allows for C_1 smoothness while the discussion of accuracy and efficiency in chapter 5 also presented results for discontinuous solutions of C_0 problems. For the latter class of problems, no validation will be performed in this work because of the limitation of the validation exercise. In addition, the SCT-Step method is removed from the list of contending methods due to its poor performance for problems featuring C_1 smoothness.

6.2 Development and Implementation of Decision Metric

The utility of the developed decision metric is to guide the practitioner's choice of which spatial discretization method to select for a specific application. As pointed out in chapter 1, various requirements may exist, and their relative importance can drastically vary. The basic assumption that the decision metric relies on is that performance indicators collected from simple test problems that may be archived in a data base can be used to predict a method's performance for different, more complicated problems.

The decision metric assigns a score to a method's expected performance that distinguishes its performance with respect to alternate methods. However, a single value of that score by itself does not have significant meaning because it does not relate to any observable quantity, e.g. solution accuracy. In that regard, the decision metric score is similar to the FOM employed in Monte-Carlo Methods[62]. In contrast to the Monte-Carlo FOM, the decision metric's score will be normalized such that it always varies in-between 0 and 1, where 0 is the lowest value (failure) and 1 is a perfect score.

The general form of the decision metric's score chosen within this work is a generalized geometric mean of scores, each associated with a single property that the practitioner deems important:

$$\Gamma = \left(\prod_{p} \Gamma_p^{\beta_p} \right)^{\frac{1}{\sum_{p=1}^{P} \beta_p}} . \tag{6.5}$$

In Eq. 6.5, p stands for accuracy, execution time, efficiency (FOM as in Eq. 5.6), or positivity (e.g. measured by τ_ψ^w) within this work. However, it is not limited to these properties of the employed numerical method. Possession of a thick diffusion limit could be incorporated with a score Γ_p that is either 0 or 1 for methods that possess or do not possess the thick diffusion limit if possession of the thick diffusion limit is an essential requirement. The same approach could be taken to enforce positive definiteness and/or cell-wise convergence for non-smooth C_0 problems. In addition, properties that are not discussed within this work but listed in chapter 1 can be added with little effort if a norm is chosen to quantify them and performance data from test cases are available.

The exponents β_p in Eq. 6.5 are interpreted as weights set by the user to control the importance of the several properties that are aggregated into the final score, Γ. The higher the value of β_p, the higher the importance of the single property p, as measured by its contribution to the total score Γ. The standard geometric mean is obtained for $\beta_p = 1,\ p = 1, ..., P$.

In order to enforce Γ to vary between 0 and 1, each of the individual scores is required to also vary between 0 and 1. Further, all quantities considered within this work, such as discretization error, τ_ψ^w and execution time, vary by orders of magnitude when the mesh is refined. Under

these circumstances, two pathological situations may arise: (1) Γ_p may vary abruptly by orders of magnitude for different refinement levels and (2) a single $\Gamma_{p'}$ may completely dominate all other Γ_p if it is much smaller/larger; thus a single property p' dominates the decision metric score that is designed to be an aggregation of scores. As these scenarios are undesirable for the anticipated objectives of the decision metric, a logarithmic scale is used to assign a score Γ_p to the quantities of interest.

For the four properties of interest: execution time T, error norm $\|\epsilon\|$, FOM and τ_ψ^w the corresponding Γ_p are given by

$$\begin{aligned}
\Gamma_T &= \frac{\log\left(\frac{T}{T_{max}}\right)}{\log\left(\frac{T_{min}}{T_{max}}\right)} \\
\Gamma_{\|\epsilon\|} &= \frac{\log\left(\frac{\|\epsilon\|}{\|\epsilon\|_{max}}\right)}{\log\left(\frac{\|\epsilon\|_{min}}{\|\epsilon\|_{max}}\right)} \\
\Gamma_{\mathrm{FOM}} &= \frac{\log\left(\frac{\mathrm{FOM}}{\mathrm{FOM}_{min}}\right)}{\log\left(\frac{\mathrm{FOM}_{max}}{\mathrm{FOM}_{min}}\right)} \\
\Gamma_{\tau_\psi^w} &= \frac{\log\left(\frac{\tau_\psi^w}{\tau_{\psi,min}^w}\right)}{\log\left(\frac{\tau_{\psi,max}^w}{\tau_{\psi,min}^w}\right)},
\end{aligned} \tag{6.6}$$

where the max and min subscripts refer to the largest and smallest occurring value of the respective quantity. These minimum and maximum values can also be used to set thresholds on the acceptable/desired value of the respective quantity if we assign:

$$x \leftarrow \begin{cases} x_{min} & \text{if } x < x_{min} \\ x_{max} & \text{if } x > x_{max} \\ x & \text{else} \end{cases}, \tag{6.7}$$

where x stands for any of the quantities listed in Eq. 6.6. With the exception of the FOM, x_{min} would then be the desired result for which a score of 1 is assigned while everything above x_{max} is considered a failure and a score of 0 is assigned. For the FOM, the roles of x_{min} and x_{max} are reversed since a higher FOM indicates a superior performance.

In Eq. 6.6, the quantities T, $\|\epsilon\|$, FOM, and τ_ψ^w all depend on the mesh spacing h, which is defined as:

$$h = \max_{\vec{i}\in\mathcal{D}} \left(\Delta x_i, \Delta y_j, \Delta z_k\right). \tag{6.8}$$

It is, however, more convenient to write the dependence of the said quantities on the dimen-

sionless cell optical thickness t defined by:

$$t = \max_{\vec{i} \in \mathcal{D}} \left(\sigma_t^{\vec{i}} \Delta x_i, \sigma_t^{\vec{i}} \Delta y_j, \sigma_t^{\vec{i}} \Delta z_k \right). \tag{6.9}$$

The transport equations within each cell can be cast in a form that depends solely on the cell optical thickness and not on the physical thickness of the cell. Thus the transport process is much better characterized in terms of the cell optical thickness t than the physical thickness h.

Evaluating the score, Γ_p, for the t values for which data is actually available, i.e. the base points, is straight forward. However, within this work the decision metric score is not only desired at base point values of t but also at intermediate points. Therefore, an interpolation scheme has to be devised that interpolates T, $\|\epsilon\|$, FOM, and τ_ψ^w over t.

For the asymptotic dependence of the execution time, error and FOM on h and thus t it is known that:

$$\begin{aligned} T \propto h^{-3} &\rightarrow T \propto t^{-3} \\ \|\epsilon\| \propto h^{\lambda} &\rightarrow \|\epsilon\| \propto t^{\lambda} \\ \text{FOM} \propto h^{0} &\rightarrow \text{FOM} \propto t^{0}. \end{aligned} \tag{6.10}$$

For all these quantities a power interpolation of the form

$$x = Ct^p, \tag{6.11}$$

would be appropriate. For τ_ψ^w the same interpolation is used for two reasons: first τ_ψ^w changes by orders of magnitude when the mesh is refined by a factor of two, $t \leftarrow t/2$, which disqualifies linear interpolation and second there is no theoretical model of how τ_ψ^w should change when the mesh is refined. Therefore, there is no indication that any other interpolation method would perform superior to the power interpolation Eq. 6.11.

The interpolation procedure searches the data array obtained from the MMS test problem and Lathrop's test problem such that $t_l < t < t_{l+1}$. Then it computes C and p as follows:

$$\begin{aligned} p &= \frac{\log\left(\frac{x_{l+1}}{x_l}\right)}{\log\left(\frac{t_{l+1}}{t_l}\right)} \\ C &= \frac{x_l}{t_l^p}, \end{aligned} \tag{6.12}$$

and finally uses Eq. 6.11 to compute the interpolated $x(t)$. In case the value of t is outside the

range of the data array, an extrapolation is performed by selecting:

$$l = \begin{cases} 1 & \text{if } t < t_{min} \\ l_{max} - 1 & \text{if } t > t_{max} \end{cases} . \tag{6.13}$$

6.3 Validation of Decision Metric's Prediction

This section exercises the decision metric to predict which methods will perform well for individual cases adapted from among the NEA box-in-box test suite. To this end, a *predicted score* is computed for the list of contending methods described before. Then, actual solutions computed with the exercised methods are used to compute an *actual score* on a relative scale. The comparison of the *predicted score* and the *actual score* constitutes the validation exercise presented in this section. Specifically, the question that will be answered here is how accurately does the developed decision metric predict the performance of methods relative to one another, given a specific set of weights β_p that reflects the user's sense of relative importance of the included measure of solution quality for different combinations of the problem parameters L, γ, σ_t, and c.

Table 6.3 lists the various combinations of parameters for setting up the NEA benchmark suite utilized within the validation exercise. In addition, Table 6.3 lists the corresponding MMS and Lathrop test data sets that are used to predict the performance of considered methods applied to the NEA suite's solution. Note, not all choices of parameters for the NEA benchmark listed in Table 6.3 are canonical, i.e. belong to the original set of parameters specified by Azmy[67]; see also Table 6.1. New sets of parameters are introduced to carefully match domain optical thicknesses and scattering ratios used in the MMS and Lathrop benchmark problems.

Within this work, each set of parameters characterizing the NEA benchmark suite, referred to as NEA-I, II, III, IV as listed in Table 6.3, is matched with a specific MMS data set and a specific Lathrop data set. The parameters that are used for matching the NEA cases with the MMS and Lathrop cases are the domain optical thickness $t_{\mathcal{D}}$, the domain aspect ratio κ and the (for Lathrop only) the minimum scattering ratio $c_{\mathcal{D}}$. These quantities are defined as follows:

$$\begin{aligned} t_{\mathcal{D},k} &= \max_{\vec{i}^k} \left(\sum_{i_k=1}^{I_k} \sigma_t^{\vec{i}} \Delta_k \right) \\ t_{\mathcal{D}} &= \max_k t_{\mathcal{D},k} \\ \kappa &= \frac{\max_k t_{\mathcal{D},k}}{\min_k t_{\mathcal{D},k}} \\ c_{\mathcal{D}} &= \min_{\vec{i}} c^{\vec{i}}. \end{aligned} \tag{6.14}$$

In Eq. 6.14, the quantity $t_{\mathcal{D},k}$ requires some explanation. It is the longest optical distance within the domain along each of the three coordinate axes: for example, $t_{\mathcal{D},x}$ is the longest optical distance along the x-axis, i.e. along $\hat{e}_x$.

For a better match between the NEA benchmark and Lathrop's test problem, two new Lathrop cases have been added to the ones listed in Table 3.4, namely Lathrop-IV-2 and Lathrop-V-1 with parameters that can be inferred from Table 6.3.

The relevance of the chosen test cases I through IV is as follows:

- Case I: This is an optically thin test case with a unit aspect ratio. The domain optical thickness $t_{\mathcal{D}}$, domain aspect ratio κ and domain scattering ratio $c_{\mathcal{D}}$ match well between the data sets (MMS and Lathrop) and the NEA-I problem. However, Lathrop's test case L-IV-2 is a very "simple" test case because of small optical cell thicknesses and decently large scattering ratios. Moreover, NEA-I is a homogeneous problem such that only a single total cross section and scattering ratio exist, and therefore matching it to an MMS and Lathrop data set of the same optical thickness and/or scattering ratio may be more justifiable than for a heterogeneous test case.

- Case II: Comparing to NEA-I, this test case exhibits two major differences. First, NEA-II is optically much thicker with $t_{\mathcal{D}} = 8$, and second the material properties in regions I and II differ. Region II's total cross section is larger than region I's cross section by a factor of five. Parameter set NEA-II is matched decently well to the MMS-II and Lathrop-I-1 test cases. In particular, NEA-II tests whether the decision metric's prediction significantly loses accuracy if the data from homogeneous test problems is extrapolated to non-homogeneous problems.

- Case III: In this case the optical thickness of the NEA-III domain is reduced to $t_{\mathcal{D}} = 3$ but it is still matched with MMS-II and Lathrop-I-1. In this case, however, MMS-II and Lathrop-I-1 match $t_{\mathcal{D}}$ poorly. NEA-III is designed to investigate how important a good match of MMS/Lathrop's $t_{\mathcal{D}}$ and the corresponding NEA $t_{\mathcal{D}}$ is.

- Case IV: In contrast to NEA-I through NEA-III, this parameter set features a non-unity aspect ratio, $\kappa = 10$. The corresponding MMS and Lathrop test cases match the aspect ratio, the domain optical thickness and the scattering ratio. This NEA parameter set is designed to investigate if conclusions change from test cases I-III if the aspect ratio is changed. NEA-IV is homogeneous, i.e. regions I and II feature the same material properties.

Solutions for the NEA benchmark suite are computed with the same codes that are used for creation of the MMS and Lathrop data sets. As the quadrature is not of concern within this work the S_4 level-symmetric quadrature is utilized throughout the entire validation exercise.

Table 6.3: Parameter variations of NEA benchmark used in the validation exercise.

Case	NEA parameters		MMS match	Lathrop match
I	$L = 1$	$c_2 = 0.5$	Match: MMS-I	Match: L-IV-2
	$\gamma = 0.5$	$t_{\mathcal{D}} = 1$	$t_{\mathcal{D}} = 1$	$t_{\mathcal{D}} = 1$
	$\sigma_{t,1} = 1$	$t_{max} = 6.25 \times 10^{-2}$	$\kappa = 1$	$\kappa = 1$
	$c_1 = 0.5$	$t_{min} = 1.56 \times 10^{-2}$	$t_{max} = 2.5 \times 10^{-1}$	$t_{max} = 1/3$
	$\sigma_{t,2} = 1$	$\kappa = 1$	$t_{min} = 4.0 \times 10^{-3}$	$t_{min} = 1.0 \times 10^{-2}$
				$c = 0.5$
II	$L = 1$	$c_2 = 0.1$	Match: MMS-II	Match: L-I-1
	$\gamma = 0.1$	$t_{\mathcal{D}} = 8$	$t_{\mathcal{D}} = 8$	$t_{\mathcal{D}} = 12$
	$\sigma_{t,1} = 2.\bar{6}$	$t_{max} = 0.8\bar{3}$	$\kappa = 1$	$\kappa = 1$
	$c_1 = 0.1$	$t_{min} = 0.208\bar{3}$	$t_{max} = 2$	$t_{max} = 4$
	$\sigma_{t,2} = 13.\bar{3}$	$\kappa = 1$	$t_{min} = 3.125 \times 10^{-2}$	$t_{min} = 1.25 \times 10^{-1}$
				$c = 0.1$
III	$L = 1$	$c_2 = 0.1$	Match: MMS-II	Match: L-I-1
	$\gamma = 0.5$	$t_{\mathcal{D}} = 3$	$t_{\mathcal{D}} = 8$	$t_{\mathcal{D}} = 12$
	$\sigma_{t,1} = 1$	$t_{max} = 1.9 \times 10^{-1}$	$\kappa = 1$	$\kappa = 1$
	$c_1 = 0.5$	$t_{min} = 4.69 \times 10^{-2}$	$t_{max} = 2$	$t_{max} = 4$
	$\sigma_{t,2} = 5$	$\kappa = 1$	$t_{min} = 3.125 \times 10^{-2}$	$t_{min} = 1.25 \times 10^{-1}$
				$c = 0.1$
IV	$L = 10$	$c_2 = 0.1$	Match: MMS-VII	Match: L-V-1
	$\gamma = 0.5$	$t_{\mathcal{D}} = 2$	$t_{\mathcal{D}} = 2$	$t_{\mathcal{D}} = 2$
	$\sigma_{t,1} = 0.2$	$t_{max} = 0.125$	$\kappa = 10$	$\kappa = 10$
	$c_1 = 0.1$	$t_{min} = 3.13 \times 10^{-2}$	$t_{max} = 0.5$	$t_{max} = 0.\bar{6}$
	$\sigma_{t,2} = 0.2$	$\kappa = 10$	$t_{min} = 1.56 \times 10^{-2}$	$t_{min} = 0.0208\bar{3}$
				$c = 0.1$

$t_{\mathcal{D}}$: Domain optical thickness — κ: Domain Aspect Ratio

t_{max}: Coarsest mesh opt. cell thickness — t_{max}: Finest mesh opt. cell thickness

Further, the iterative stopping criterion is set to 1.0×10^{-10}. However, some of the target volume's exact averaged scalar flux is sufficiently small, i.e. $< 1.0 \times 10^{-3}$, such that on fine meshes the errors are $\approx 10^{-8}$, and the computed discretization error is contaminated with iterative stopping error. In order to ensure an accurate estimation of the discretization error only those target volumes in Table 6.2 are considered that feature averaged scalar fluxes $> 1.0 \times 10^{-2}$.

For validation of the performance metric three different choices of weights are considered. Let $\vec{\beta} = \left(\beta_{\|\epsilon\|}, \beta_{\tau_\psi^w}, \beta_T, \beta_{\text{FOM}}\right)$, then the three considered sets of weights choices are:

$$\begin{aligned} \vec{\beta}_1 &= (0,0,0,1) \\ \vec{\beta}_2 &= (1,0,1,0) \\ \vec{\beta}_3 &= (0,1,0,2)\,. \end{aligned} \tag{6.15}$$

The first two choices, $\vec{\beta}_1$ and $\vec{\beta}_2$, align in terms of the objectives of the code practitioner: The FOM is a measure of efficiency and is used as the sole quantity to compute the methods' score Γ, while for $\vec{\beta}_2$ both errors and execution times are combined using unit weight for both. Thus both $\vec{\beta}_1$ and $\vec{\beta}_2$ are scenarios in which the practitioner seeks an efficient discretization method. In contrast choice three combines resilience against negative angular face fluxes with the FOM with weighting factors one and two, respectively. This indicates a scenario, where the user is mostly interested in an efficient methods. A certain level of negative fluxes is acceptable within this scenario but the user is interested in a small measure of negative fluxes.

For a concise presentation of the validation exercise's results a penalty function is defined measuring how well the predictions match the actual best-to-worst performance rankings. Let Γ be the predicted scores and $\tilde{\Gamma}$ be the actual scores. Then, for each level of mesh optical thickness separately, a ranking ordered from best performer to worst performer can be inferred from the predicted scores:

$$\Gamma\left(\mathcal{M}\right) \rightarrow \vec{\mathcal{M}} = \left(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, ..., \mathcal{M}_{L-1}, \mathcal{M}_L\right), \tag{6.16}$$

where $\Gamma\left(\mathcal{M}\right)$ is the score associated with discretization scheme $\mathcal{M}$. The discretization scheme $\mathcal{M}$ comprises the specification of a method and a particular order, so $\mathcal{M}$ could for example stand for LL or AHOTN-3 or any of the other introduced methods of a specified order. From the scores a ranking of performance from best to worst is inferred that is saved in the vector $\vec{\mathcal{M}}$ with $\mathcal{M}_1$ being the best performing discretization scheme and $\mathcal{M}_L$ being the worst performer.

Further, the best performer according to **actual** scores, again separately for each level of mesh optical thickness, is denoted by $\mathcal{M}^*$, and it satisfies:

$$\tilde{\Gamma}\left(\mathcal{M}^*\right) \rightarrow \tilde{\Gamma}_{max}. \tag{6.17}$$

The l-th penalty function is defined as the ratio of the **actual** score of the $\mathcal{M}_l$ discretization scheme and $\tilde{\Gamma}_{max}$:

$$r_l = \frac{\tilde{\Gamma}\left(\mathcal{M}_l\right)}{\tilde{\Gamma}\left(\mathcal{M}^*\right)}. \tag{6.18}$$

The penalty function r_l is a measure of how well the prediction reproduces the best-to-worst ranking of the actual scores. A good prediction should feature $r_1 = 1$ and $r_1 > r_2 > r_3 > ... > r_{L-1} > r_L$ for all levels of cell optical thickness. In addition, from the discretization schemes that received the three highest scores at least one should be within a range of 5-10% of the best performer, i.e. $r_l > 0.9$ for $l = 1, 2,$ or 3. The penalty function is designed to quantify the penalty (if any) measured by the objective $\vec{\beta}$ that the user actually incurs for using the metric's prediction.

The penalty functions r_1, r_2, r_3, and r_L are plotted versus the cell optical thickness for validation cases NEA-I to NEA-IV in Figs. 6.2 to 6.5. If not stated otherwise target quantity 1.a is considered. An explicit listing of all validation results obtained in the course of this work can be found in appendix F. The computed penalty functions presented here are based on these findings and capture the essential information that can be inferred from them in support of our conclusions.

For the NEA-I validation case penalty functions are depicted in Fig. 6.2. The quality of the prediction is very good for $\vec{\beta} = (0, 0, 0, 1)$ and $\vec{\beta} = (1, 0, 1, 0)$ if target quantity 1.a is considered. For both cases the penalty function r_1 is very close to unity and for five out of the six depicted data points it is exactly unity corresponding to correctly identifying the best performer. For the intermediate cell optical thickness level and $\vec{\beta} = (0, 0, 0, 1)$, neither the first, second, nor third choice is the actual best performer. The prediction metric misses that AHOTN-2 performs best under the said circumstances. However, the possible detriment arising from this inaccurate prediction is small considering how close r_1 is to unity.

The second and third best performer are well predicted when using the $\vec{\beta} = (1, 0, 1, 0)$ weights but for $\vec{\beta} = (0, 0, 0, 1)$ their prediction is not as good. First, the ranking of these two discretization schemes, predicted to be LN and LL, is switched: LN is predicted to perform better than LL but in reality it is the other way around. Second, the decision metric misses that AHOTN-2 should be the third method of choice, and not LN.

When selecting 3.c as target quantity with weights $\vec{\beta} = (1, 0, 1, 0)$ the results resemble the ones obtained with the same weights but for target quantity 1.a with the single difference that for the coarsest optical thickness the predicted winner, the LD method, does not perform as well as projected. The actual best performer for the given optical thickness is the DGC-2 methods which is also not predicted as second or third best. However, the inaccuracy does not lead to a detrimental choice given that LD would still have about 90% of the score that DGC-2 has.

Finally, the accuracy of the prediction for $\vec{\beta} = (0, 1, 0, 2)$ is poor. The best performer is not

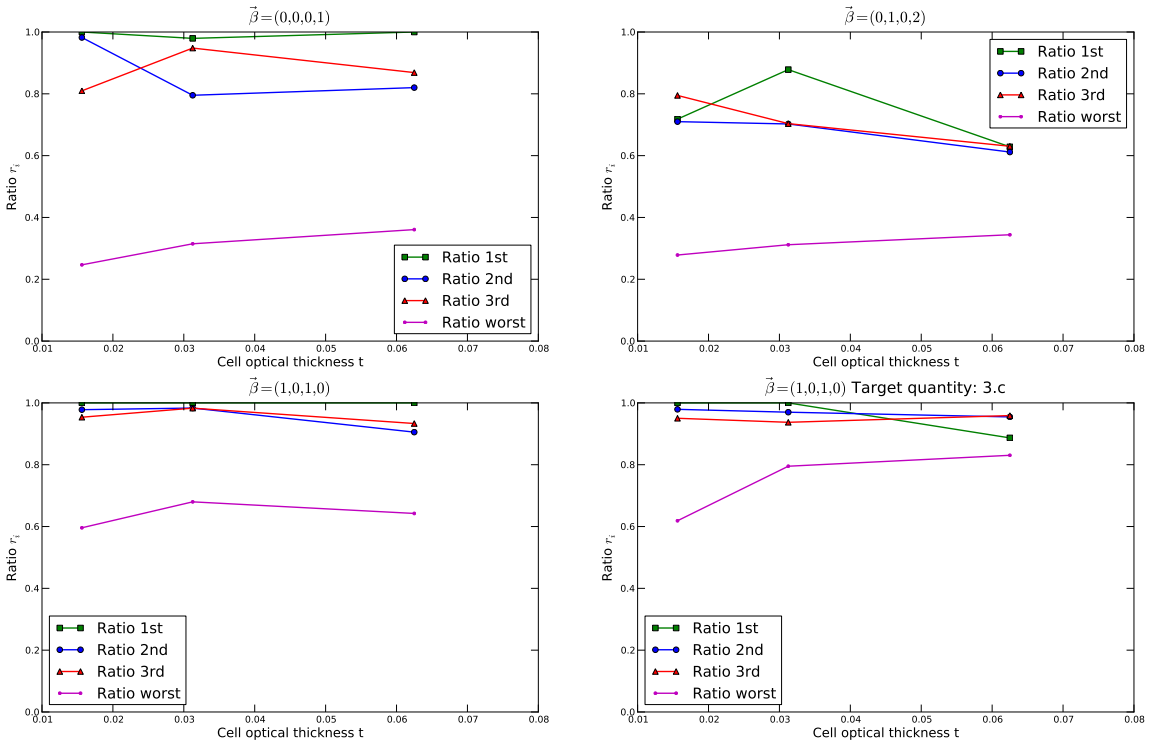


Figure 6.2: Penalty functions r_l for $l = 1, 2, 3, L$ and NEA-I test cases for all considered decision metric's weights $\vec{\beta}$.

found for any of the three mesh refinement levels and the incurred performance penalty may be up to 40 % (predicted best performer has 0.6 times the score of the actual best performer). Note that the performance score is a logarithmic scale so 40 % amounts to a significant performance disadvantage. The reason for the poor accuracy of the prediction is that the Lathrop-IV-2 test case is too simple to supply reliable information regarding methods' resilience against negative fluxes. Even for methods with little resilience against negative fluxes and on coarse meshes it is found that the solutions are entirely positive, which biases the performance metric's prediction. However, the benefit of using the suggested method as opposed to the actual worst performer is still large: 40% penalty versus a 70% penalty. Therefore, even under worst case conditions the decision metric does not suggest using detrimentally poor performers.

For the NEA-II validation case penalty functions are depicted in Fig. 6.3. Except for the last case, $\vec{\beta} = (1, 0, 1, 0)$ applied on target quantity 3.c, the best performer is, in general, not predicted correctly. For each of the weights $(0, 0, 0, 1)$, $(0, 1, 0, 2)$, and $(1, 0, 1, 0)$ (1.a) the best performer is only predicted correctly for a single level of optical thicknesses. However,

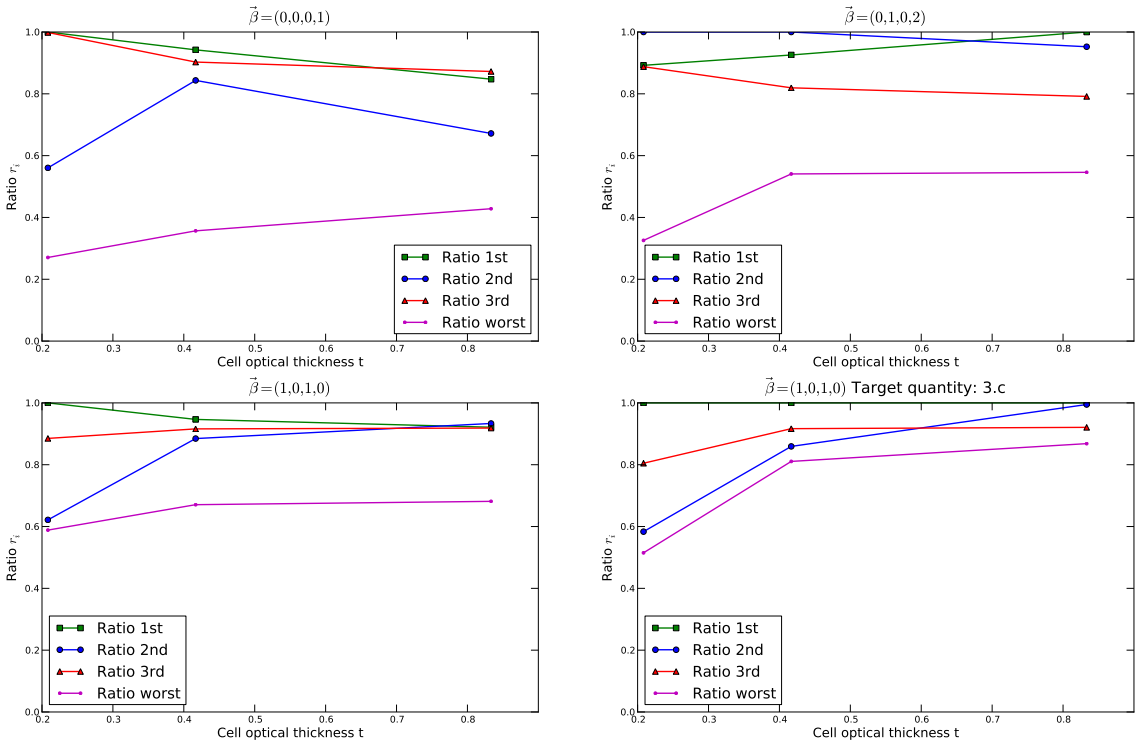


Figure 6.3: Penalty functions r_l for $l = 1, 2, 3, L$ and NEA-II test cases for all considered decision metric's weights $\vec{\beta}$.

the suggested discretization scheme never fails the best performance mark by much so it would actually be a good candidate for deployment on the NEA-II problem with performance penalties only about 10-15%. The predictions of the best performer for target quantity 3.c and $\vec{\beta} = (1, 0, 1, 0)$ are very accurate.

With the exception of the $(0, 1, 0, 2)$ weighting, the second best performer is not predicted correctly for any of the sets of weights. The reason for this inaccuracy is that LL is over predicted such that it becomes the runner-up discretization scheme, but in reality its scores drop significantly on the finest considered spatial mesh (lowest cell optical thickness). As a consequence, all r_2 curves exhibit a drop for the finest utilized mesh.

Observing that $\vec{\beta} = (0, 0, 0, 1)$ often features the worst predictions, it should be pointed out here that the FOM introduced here for spatial discretization methods of the S_N methods is only valid in the asymptotic regime, i.e. once the error decreases monotonically with mesh refinement at a constant rate λ. This condition may not be satisfied for the NEA benchmark results even though it is satisfied for the MMS test cases that the prediction is based on.

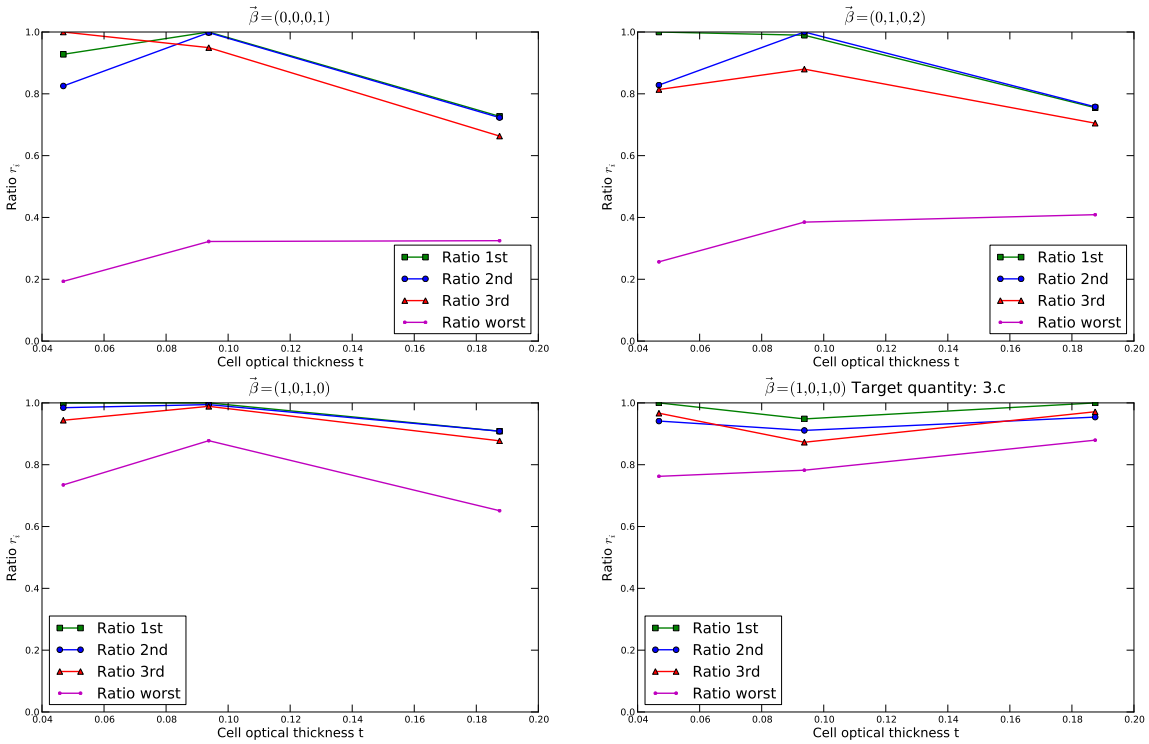


Figure 6.4: Penalty functions r_l for $l = 1, 2, 3, L$ and NEA-III test cases for all considered decision metric's weights $\vec{\beta}$.

For the $(0, 1, 0, 2)$ weighting the only shortcoming of the prediction is that the best-performer and the runner-up methods are switched with minor impact on the total score.

For the NEA-III benchmark case, penalty functions are depicted in Fig. 6.4. This benchmark case is intended to investigate if a mismatch of global parameters between the NEA benchmark case and the MMS/Lathrop test cases would lead to poorer predictions than with well-matched global parameters. From Fig. 6.4, we infer that the answer depends on the choice of the weights.

For both $\vec{\beta} = (1, 0, 1, 0)$ weighted examples, the predictions are accurate: The best performer is correctly identified for most of the cell optical thicknesses and for the two data points where that is not the case the incurred penalty is less than 10%. The ranking of the second and third best methods is reasonably accurate. For target quantity 1.a it strictly holds that $r_2 < r_1$, while for target quantity 3.c the said condition does not strictly hold. However, the actual scores $\tilde{\Gamma}(\mathcal{M}_2)$ and $\tilde{\Gamma}(\mathcal{M}_3)$ do not differ by much such that getting the exact order of the top three performers incorrect is inconsequential to the objective defined by $\vec{\beta} = (1, 0, 1, 0)$.

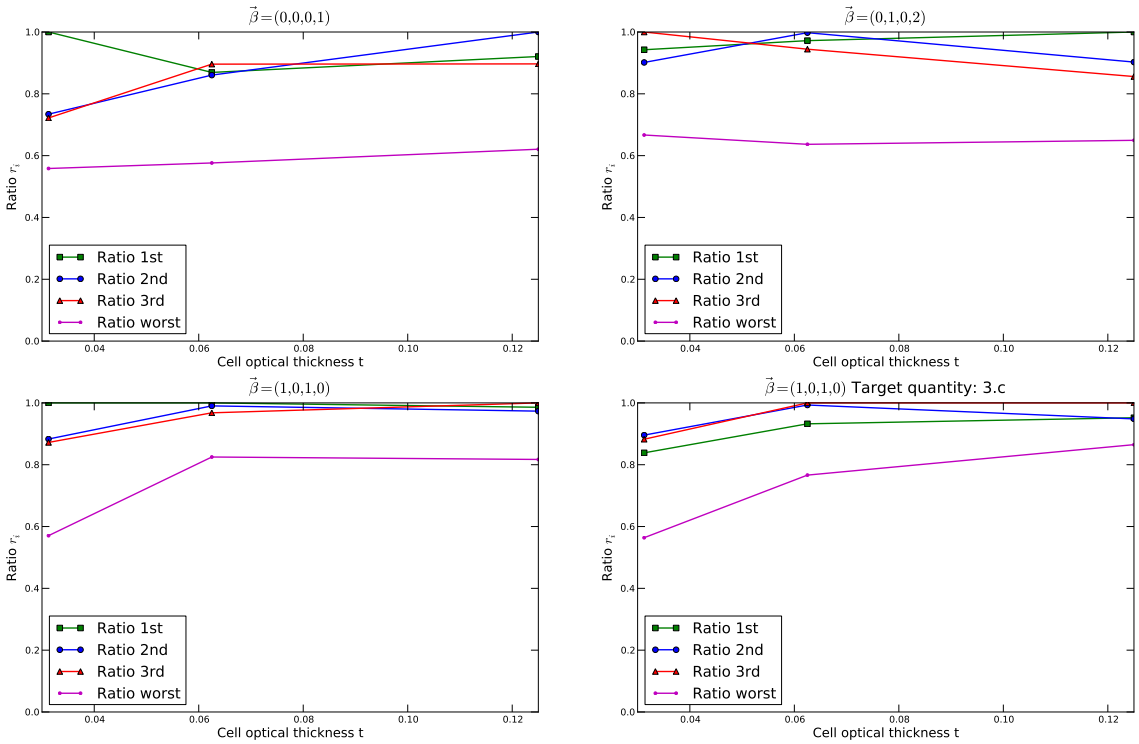


Figure 6.5: Penalty functions r_l for $l = 1, 2, 3, L$ and NEA-IV test cases for all considered decision metric's weights $\vec{\beta}$.

For the remaining set of weights $\vec{\beta} = (0, 0, 0, 1)$ and $(0, 1, 0, 2)$ the predictions are deficient, because for the coarsest cell optical thickness none of the first three methods is within 20% of the best performer. Therefore, a choice based on the decision metric would have detrimental effects on the quality of the transport computation performed on the coarsest mesh. However, for the two finer mesh refinement levels, the predictions are accurate enough to base a beneficial decision on them.

Figure 6.5 depicts the penalty functions for the NEA-IV benchmark case. In general, the predictions are accurate. With the exception of the $\vec{\beta} = (0, 0, 0, 1)$ and the $\vec{\beta} = (1, 0, 1, 0)$ (target 3.c) weighted scores, the best performer is either accurately predicted ($\vec{\beta} = (1, 0, 1, 0)$) or is so close to the actual best performer that the incurred performance penalty is of little consequence ($\vec{\beta} = (0, 1, 0, 2)$). Selection of the second or third best method would incur performance penalties of about 10%.

For the $\vec{\beta} = (0, 0, 0, 1)$ scenario predictions are not as accurate. For the second mesh refinement step for the $\vec{\beta} = (0, 0, 0, 1)$ case neither of the three top predicted methods comes

within 15% of the true best performer. For the coarsest and finest mesh refinement steps, the predicted runner-up and predicted best-performer turn out to be the actual best performer, respectively, such that in these cases at least one of the top-three methods is good candidate method for producing an efficient solution of the NEA-IV benchmark problem. However, the prediction metric fails to make a sensible suggestion for the intermediate mesh refinement step.

Finally, for the $\vec{\beta} = (1, 0, 1, 0)$ (target 3.c) scenario, predictions are reasonably accurate for the two coarser mesh refinement steps with the actual best performer predicted as the third best discretization scheme. In addition, the suggested best performer is within 5-10% of the actual best performer. However, on the finest mesh, the decision metric fails to find and suggest a discretization scheme that comes within 15% of the actual best performer. However, comparing to the worst performer which is about 40% worse than the best performer, shows that the suggested methods are far from detrimental choices.

The validation exercise shows that the performance metric can be valuable as it correctly predicts the best performer for a fairly large fraction of the validation test cases. In addition, it never suggests any of the worst performing methods to be the most suitable, so catastrophic predictions have not been observed in the attempted cases. For the NEA-I, $\vec{\beta}_3$ case a plausible explanation for the poor prediction is given: the far too simple Lathrop-IV-2 test case. For the NEA-III validation exercise the mismatch between the data sets and the NEA-III parameter is intended to throw the prediction off so it is not surprising that the predictions are actually worst for this example. Often, the best performer is predicted correctly, or at least the suggested discretization scheme is within 10% of the actual best performer. Under the worst circumstances, among the first three suggestions of the decision metric is at least one that comes within 15% of the best performer's score.

However, the decision metric is not always absolutely right. If the stricter criterion, namely that only the first suggestion from the decision metric is considered, is applied, performance penalties, between suggestion and actual best-performer, up to 20% penalties are possible[1]. However, these 20% are still much smaller than possible penalties that would arise from using one of the worst performing methods: these can be up to 70%. It should be stressed that these translate into orders of magnitude of larger solution errors or negativity measures because of the logarithmic scale of performance scores. This underscores that even at its worst the decision metric does not suggest using detrimentally bad performing discretization schemes.

Compared to a purely qualitative assessment of the performance, the decision metric offers some advantage. First, if two performance aspects are combined, say positivity and efficiency, a qualitative assessment cannot aggregate them as well a quantitative metric can. Basically, a qualitative assessment can only look at each of the two separately and a method could be

[1]Excluding the NEA-I case with $\vec{\beta} = (0, 1, 0, 2)$ and the NEA-III benchmark case, where for already discussed reason larger penalties would occur.

picked that does well for both aspects if this method exists. In a scenario where each method is good for one aspect but not for the other, the decision will be much harder. One could select the method solely based on the more important performance aspect. However, this approach neglects methods that rank say second or third for both aspects but the combined rank is better than any of the other methods.

In order to improve the decision metric's accuracy three approaches shall be mentioned here: First, more data may be collected both from the existing test cases and additional benchmarks. Second, a suitable interpolation should be devised to obtain predicted scores for problem parameters within the set of "base points". In addition, a way should be developed to aggregate data from different benchmarks: for example accuracy data from the MMS benchmark problem that is combined with several other available benchmark data. Third, the target quantities can be separated in better, more accurate categories. For example, instead of only separating target quantities by the norm they are measured in, one could classify a region averaged flux by its size compared to the domain size if that changes the accuracy of different discretization methods relative to each other. Naturally, more separation of target quantities would require to obtain more data from the test cases, hence more storage capacity.

In conclusion, the decision metric can be a useful utility over a purely qualitative assessment of the methods' performance. However, its predictions are not always accurate which may be partially attributed to lack of sufficient data to utilize and/or insufficient distinction among target quantities within this work.

Chapter 7

Summary and Conclusions

The goal of this work is to conduct a comprehensive, quantitative comparison of a family of spatial discretization methods referred to as nodal methods. These nodal methods are closely related to Discontinuous Finite Element Methods, and they share a set of common properties such as local definition of function spaces and weak cell coupling only through face fluxes (coupling in an integral sense). Building on work in Ref. [1], the FEM framework wrapping around most of the methods considered in this work is developed: The DGFEM method naturally is a nodal method, the AHOTN method is shown to be a nodal method in [1], and the HODD method is proved to be a nodal method, in particular a Discontinuous Petrov Galerkin FEM, within this work.

The traditional derivation of the methods considered in this work are also discussed in detail: higher-order diamond difference method (HODD), arbitrarily high-order transport method of the nodal type (AHOTN), DGFEM methods using the *Lagrange* and *complete* function spaces (DGLA and DGC, respectively), the simple corner balance method (SCB) and the linear nodal (LN) and linear-linear (LL) methods.

The comprehensive comparison of the contending methods is facilitated by exercising these methods based on three test problems each of which is designed to measure a certain aspect of the various qualities that radiation transport code users are interested in. Within this work the three test cases: a three-dimensional Method of Manufactured Solution test suite, Lathrop's test problem, and a simple cube thick diffusion test problem are used to investigate the contending methods' performance with respect to accuracy and execution time, resilience against negative fluxes, and possession of the thick diffusion limit, respectively.

The MMS test problem leverages the approach of Duo[1] for creating a test problem with escalating order of solution non-smoothness for two-dimensional Cartesian geometry for which the exact solution is known. Within this work, Duo's work is extended to three-dimensional Cartesian meshes. To this end tracking, tessellation, and analytical integration routines are

developed for computing cell Legendre moments of the exact angular flux, scalar flux, and distributed source, which serves as input to the radiation transport codes that are exercised based on the MMS test problem.

The tracking procedure determines intersection points of the Singular Characteristic Line (SC) and Singular Planes (SP) with the mesh cells. It puts a high premium on computational efficiency and accurate determination of the cells that are intersected by SC/SP. The tessellation routines are wrappers around several Geompack90[52] subroutines. Finally, the integration routines compute the exact cell Legendre moments of the angular/scalar flux and distributed source almost to double precision (2.24×10^{-16}). The MMS test suite is implemented into the code MMS3D which allows variation of the smoothness of the exact flux solution which is the most important property when it comes to the expected accuracy and performance of spatial discretization methods solving the MMS test problem. For quantifying the accuracy of the contending discretization methods error norms are defined and classified following the description of the MMS test problem. The two most important error norms used within this work are the $\mathcal{L}$ error norm and the integral error norm.

Lathrop's test problem is used to assess the resilience of spatial discretization methods against negative fluxes. We focus on the occurrence of negative cell-averaged scalar fluxes and negative face-averaged angular fluxes. Two new measures, τ_{ψ}^{w} and τ_{ϕ}^{w}, are introduced that essentially play the same role for measuring the positivity of the solution as the error norms do for quantifying the methods' accuracy for the MMS test problem.

The thick-diffusion limit test problem is adapted from Ref. [58] for three spatial dimensions. A set of problems is created via the small parameter ϵ, driving the total cross section to infinity and the scattering ratio to unity when $\epsilon \to 0$. The solution satisfies a diffusion equation to leading order for this test problem and the contending spatial discretization methods are required to limit to the diffusion solution in the case that the optical cell thickness increases as $1/\epsilon$. If a method approaches the diffusion solution from above, then it possesses the thick diffusion limit, but if the computed flux limits to zero, it does not possess the diffusion limit.

Numerical results of the contending methods, i.e. performance data, for the three described test problems are a key item within this work. They are used for two different purposes. First, a comprehensive comparison is undertaken based on the performance data to assess which discretization methods performs well with respect to which performance rubric and under which circumstances (solution smoothness, configuration parameters such as domain optical thickness, selected error norm etc.). Second, the data is used to develop a decision metric that is intended for predicting which spatial discretization method will perform best on a different test problem for a given weighted list of solution performance properties that the code user deems important. The decision metric thereby computes a fitness score from the data obtained for the MMS test problem and Lathrop's test problem and uses this score to find the best discretization method

among the set of considered methods in this work.

The set of spatial discretization methods is implemented as efficiently as possible to level the playing field for conducting a fair comparison of their performance as much as possible. Particular care was taken that none of the methods incurs additional overhead compared with the other methods. The grind times, i.e. the execution time of the kernel subroutine or the execution time necessary to solve a single mesh cell for a single discrete ordinate, were measured, and the ranking of the methods regarding the grind time is summarized in section 7.1.

A new method, the SCT-Step method, was implemented for improving on two shortcoming of standard methods when the exact solution is discontinuous. First, standard methods' solutions are not convergent in the $\mathcal{L}_\infty$ norm, i.e. cell-wise, and second they exhibit extremely small convergence rates in all other $\mathcal{L}_p$ norms. The SCT-Step method is an extension of Duo's SCT method[1] to three-dimensional Cartesian geometry. The SC and SPs are tracked using the tracking procedure implemented for the MMS test suite, and the cell segments that arise from the separation are solved separately using the *Step* approximation within each segment. The face fluxes are subsequently saved and propagated by cell segment and are not smeared over the outflow faces. Thus, cell-wise convergence and order one observed convergence rates are restored, even for test problems with a discontinuous solution. It is important to stress that splitting of the mesh into segments is a local operation: the segments only exist for the brief period that the cell is solved, and the volumetric flux is immediately collapsed to the full-cell scale at the conclusion of the cell solution instructions.

For increasing efficiency, reliability and accuracy of the AHOTN, LL, and LN methods, a new method for computing the spatial weights was developed. It is based on a table lookup in conjunction with a (1,1) Pade interpolation between base points. For each base point, the three constants defining the Pade approximation at the corresponding optical cell thickness are saved. The table lookup is very cheap and accurate, and no splitting of the weight computation into optically thin and thick cells is required. Numerical results corroborate the efficiency and accuracy of this approach.

Numerical results are obtained for a range of parameters for the MMS and Lathrop test problems. A comprehensive discussion and comparison among all contending methods was performed, carefully separating the different parameter regimes where methods' performance results change. The obtained results are archived for the purpose of computation of the decision metric score, which constitutes the quantitative decision metric developed within this work. A detailed listing of the findings can be found in section 7.1.

As a preliminary study for Lathrop's test problem, the cause of negative fluxes based on the solution of a single cell is investigated. This study is very much in the spirit of a local error analysis. The outflow face-averaged angular fluxes are connected to the inflow face-averaged fluxes and the cell-averaged source by coupling coefficients. If these coupling coefficients become

negative, negative fluxes can occur. Within the pertaining section it was shown that cell sources always increase the flux level and thus never cause negative face-averaged outflow fluxes. However, West to East, South to North and Bottom to Top coupling coefficients can become negative and cause negative outflow fluxes.

As the cell-averaged source always increases the flux level and thus may prevent negative face-averaged angular outflow fluxes, the following experiment was performed. Lathrop's test problem was solved with and without using a first collision source. The first collision source is computed by ray-tracing and is therefore positive. The hope is that the first collision source elevates the flux level in outlying source-free regions and prevents the occurrence of negative fluxes. Results for this experiment can be found in section 7.1.

Numerical results for the thick diffusion limit test problem are augmented by analysis as performed in Ref. [7] for the DD, HODD-1, AHOTN-0, AHOTN-1, LL, and LN methods in three-dimensional Cartesian geometry. To the knowledge of the author, the analysis performed for the named methods is new for three-dimensional Cartesian geometry. The numerical results corroborate the analysis wherever available, and produce numerical evidence pertaining to the possession of the thick diffusion limit by all other methods. A detailed list of which methods possess the thick diffusion limit can be found in section 7.1.

The numerical results obtained for the three test problems culminate into a qualitative ranking of the contending spatial discretization methods. This ranking allows code practitioners to quickly check various performance rubrics and compare methods among each other to find a suitable, if not the most suitable, discretization method for their application.

The decision metric is an attempt to create a quantitative measure that predicts how well a discretization method will perform for a certain purpose, i.e. given a checklist of aspects that the code practitioner is interested in. The decision metric score is a generalized geometric average of weighted single-aspect scores, each of which is obtained from the data collected from the MMS and Lathrop's test problems. A properly normalized logarithmic scale is used to obtain single-aspect scores varying between zero and one.

The predictions of the developed decision metric are validated against the NEA box-in-box benchmark problem. The validation exercise consists of predicting the performance of the contending discretization methods for different parameter sets of the NEA benchmark which concludes the prediction step. At the end of the prediction step the predicted scores ranking the contending methods from best to worst is obtained. The validation step comprises solving the NEA benchmark problem directly using the contending methods, determining accuracy, execution time and negativity measures and using these quantities for computing the real performance score. The validation exercise is completed by comparing the predicted and real scores via a penalty function.

7.1 Findings

Nodal FEM Framework: The HODD method is found to belong to the class of nodal finite element methods, but in contrast to the two DGFEM families, used within this work the HODD is a Discontinuous Petrov Galerkin FEM characterized by different test and trial spaces. The test space $\mathcal{V}$ of the HODD-Λ method is a *Lagrange* space of order Λ, i.e. $x^{m_x}y^{m_y}z^{m_z}$ for $m_x, m_y, m_z = 0, ...\Lambda$, while the trial space $\mathcal{T}$ is equal to the test space augmented by the following trial functions:

$$\mathcal{T} = \mathcal{V} \cup \{x^{\Lambda+1}y^{m_y}z^{m_z}\}_{m_y,m_z=0,...,\Lambda} \cup \{x^{m_x}y^{\Lambda+1}z^{m_z}\}_{m_x,m_z=0,...,\Lambda} \cup \{x^{m_x}y^{m_y}z^{\Lambda+1}\}_{m_x,m_y=0,...,\Lambda}.$$

For generating enough equations for uniquely determining all the expansion coefficients weak continuity between the inflow interior and exterior traces is imposed.

MMS3D Tracking Routine: The code MMS3D implements the preparation of all necessary input data for the execution of a generic S_N solver and computation of the cell-wise spatial discretization error afflicting this solution. Within the MMS3D code, tracking of the Singular Characteristic (SC) and Singular Planes (SPs) is facilitated using an algorithm that is based on following/tracking the SC, and subsequently the SPs, through the computational domain. It is shown that this algorithm scales with the number of linear subdivisions I_k with $k = x, y, z$, where I_k is the number of cells in the k dimension. If utilized in a *Singular Characteristic Tracking* type S_N solver implementation of this algorithm will consume much less execution time in the limit of many mesh cells than the actual mesh sweep.

Computation of Spatial Weights for AHOTN, LL, LN: A new approach for computing the spatial weights of TMB methods, i.e. AHOTN, LL, and LN, is introduced within this work. It is based on a table lookup with Pade interpolation for obtaining values in-between base points. The advantage of the utilized (1,1) Pade approximation is that it correctly reproduces both the fine and coarse mesh limits for even and odd Λ. The implementation is found to be very efficient and accurate, with relative errors of the computed weights being bounded below 1 %. Most errors however, are smaller than 0.01 %.

Methods' Grind Times: The grind time is the execution time required for completing all operations necessary for the solution of a single mesh cell for a single direction. Grind times are measured for all contending discretization methods, along with a break-down of the operations it consists of grouped into four categories: computation of spatial weights, setting up the local linear system, solution of the local linear system, and upstreaming. It is found that the LD and DD method have the shortest grind times ($0.1\mu s$), followed by the LN ($0.5\mu s$) and LL ($1\mu s$)

methods.

The arbitrary order methods start with grind times of 1.7, 4.3, 4.5, and 5.3 μs for DGC-1, DGLA-1, HODD-1 and AHOTN-1, respectively. Increasing Λ leads to significantly longer grind times because the linear solve step scales with the ninth power of Λ[1]. The modest increase in grind time is observed for the DGC method because it omits some of the higher-order mixed unknowns. Following the scaling argument, the linear solve takes the largest fraction of the grind time even for $\Lambda = 1$ method, but increases with Λ to be the sole dominating task during the cell solve. The ranking from fastest to slowest does not remain the same for all orders of Λ: for $\Lambda = 1$ the order is DGC-1, DGLA-1, HODD-1 and AHOTN-1, while for $\Lambda = 3$ the order is DGC-3, HODD-3, AHOTN-3 and finally DGLA-3. The DGLA method's grind time appears to increase faster than the other methods' grind times.

FOM for Comparison of Discretization Methods: The Figure of Merit plays a dominant role in comparing the efficiency of various Monte-Carlo methods with and without variance reduction. The higher the FOM, the more efficient the associated Monte-Carlo method. Within this work, a similar quantity was derived for spatial discretization methods that is based on the observation that the observed accuracy (rate of convergence of the discretization error) is limited by the smoothness of the exact solution. Therefore, it does not matter which discretization methods is utilized, the rate of convergence λ will be problem dependent but invariant for all methods. The FOM for comparison of spatial discretization methods suggested within this work is given by:

$$\text{FOM} = \frac{1}{\|\epsilon\| T^{\lambda/3}},$$

where $\|\epsilon\|$ is the discretization error and T the total execution time. This quantity becomes constant in the asymptotic regime and grows larger the more efficient the method is. Therefore, it plays essentially the same role for S_N discretization methods as the FOM for Monte-Carlo methods. The FOM is used to measure computational efficiency in the framework of the decision metric discussion.

Cancellation of Error: Cancellation of error is an artifact of error norms that "average before applying absolute values". Under these conditions positive and negative contributions of the pointwise error distribution can cancel and the error measured in a particular error norm appears to be abnormally small. It is found within this work that cancellation of error measured with the quantity Ca decreases with mesh refinement if discrete $\mathcal{L}_p$ norms are employed, but does not decrease with mesh refinement for integral error norms. Numerical evidence that cancellation of error is responsible for volatile dips and non-monotonicity in the error-vs-execution time curves is provided by comparing discrete $\mathcal{L}_2$ error norms (allows cancellation of error) with continuous

[1]LU decomposition scales like n^3, where n is the number of unknowns and here $n \propto (\Lambda + 1)^3$

$\mathcal{L}_2$ error norms (does not allow cancellation of error). For the latter case, the dips disappear and a straight line in a log-log plot of error versus execution time is obtained.

It is observed that cancellation of error is more severe for C_0 problems, which is attributed to the oscillation of numerical solutions around discontinuities (Gibbs' phenomenon). The oscillations translate into a pointwise error distribution that has a small mean value but large amplitudes which naturally encourages the occurrence of cancellation of error.

The SCT-Step Method: The SCT-Step method is created to resolve problems that standard methods encounter during the solution of C_0 problems, namely extremely small rates of convergence and lack of cell-wise convergence. The SCT-Step method utilizes the efficient tracking algorithm originally developed for the MMS3D code such that numerical overhead compared with standard discretization methods decreases as the mesh is refined. The Step approximation is utilized within cells that are intersected by the SC or SPs and AHOTN-0 is employed for all other cells.

In C_0 configurations, the SCT-Step method recovers cell-wise convergence and is more efficient than standard methods for sufficiently fine meshes. However, it is found that while some methods feature smaller errors than SCT-Step on coarse meshes, these results are largely facilitated by cancellation of error, which makes these results not trustworthy. In contrast, for C_1 problems and when the error is measured in the integral norm, standard methods are superior to SCT-Step. This is due to the poor quality of the Step approximation. Therefore, it is suggested to devise a higher-order version of SCT algorithm capitalizing on the possible higher rates of convergence.

HODD for Optically Thick Non-Diffusive Cells: In the non-diffusive thick limit, i.e. $\sigma_t \to \infty$, but $c << 1$, the exact angular flux solution behaves like $1/\sigma_t$. In order to yield reasonable results, a discretization method must satisfy the same behavior in the said limit for both cell flux moments and face flux moments. Using a discrete version of the asymptotic analysis, it is shown that weighted diamond difference (WDD) methods satisfy the $1/\sigma_t$ behavior only if the spatial weights limit to unity for $\sigma_t \to \infty$. This condition is satisfied for the AHOTN method but not for the HODD method. Therefore, the HODD method for all orders fails dramatically for optically coarse meshes.

General Results for MMS Test Problem:

- Results for C_1 problems:
  - The most efficient method depends on the norm that is used to quantify the error. A synopsis of the best performer depending on norm and test case specification is provided in Table 7.1.

- When $\mathcal{L}_p$ norms are utilized, higher-order methods perform better than lower-order methods. However, if integral quantities are sought, first order methods perform better than higher-order methods. The reason for this behavior is the larger permissible rate of convergence λ for integral error norms. This can be proven by looking at the ratio of the FOM for two spatial discretization methods, one high-order method and one low-order method. The ratio of their FOMs are:

$$d = d_\epsilon d_t^{\lambda/3},$$

where d_ϵ characterizes the ratio of the low-order method's error to the high-order method's error on the same mesh and d_t is the ratio of their grind times. For $d < 1$ the higher-order method is better, while $d > 1$ means that the low-order method is better. Typically, $d_\epsilon < 1$ and $d_t > 1$. However, with increasing λ more importance is given to a shorter execution time such that for some λ the cheaper method is always more efficient.

- For $\mathcal{L}_p$ error norms AHOTN-3 performs best except for highly skewed aspect ratios.
- For $\mathcal{L}_p$ error norms and highly skewed aspect ratios the DGC-3 method performs best. The reasons are its inherent cheapness compared to AHOTN-3, HODD-3, and DGLA-3, and the low impact of mixed cross moments on the solution for highly skewed meshes.
- For optically thick cells the TMB methods, such as AHOTN ($\mathcal{L}_p$ norms) and LN (integral norms) outperform the competitors by a large margin.
- Generally, for integral error norms the LN methods performs most efficiently.
- A boundary layer effect is observed for optically thick domains. It is caused by the combination of a norm that allows cancellation of error and a flux profile that is very steep close to the boundary but essentially flat away from the boundary. It is shown that smearing across the mesh cells that contain the boundary layer first lead to a very small error, then to a peak in the error versus mesh-spacing curve and finally to a decrease of error leading into the asymptotic regime. While the boundary layer effect is related to cancellation of errors, the distinct difference from generic error cancellation effects is the smoothness of the resulting curves.

- Results for C_0 problems:
  - If the error is measured in a $\mathcal{L}$ norm, SCT-Step is the best performing method.
  - If the error is measured in an integral norm, the same conclusion holds as for C_1 problems: LN is the best performer.

Table 7.1: Synopsis of the best performers for different parameters and norms for the MMS test suite .

	C_0		C_1	
	$\mathcal{L}_p$	Integral Norm	$\mathcal{L}_p$	Integral Norm
Standard Case	SCT-Step	LN	AHOTN-3	LN
Optically Thick	SCT-Step	LN	AHOTN-3	LN
Standard Case	SCT-Step	LN	DGC-3	LN

Positivity of Spatial Discretization Methods:

- Coupling Coefficients:
  - Coupling coefficients couple inflow and source averages to outflow face-averaged fluxes. Since inflow/source average ought to be positive, the cause for negative outflow averages are necessarily negative coupling coefficients.
  - Source-outflow coupling is found to be always greater than zero. Therefore, cell-averaged sources always contribute to a positive outflow flux.
  - Incorrect coupling is defined as coupling between inflow and outflow faces that, in the real world, should be uncoupled. Incorrectly coupled faces of the type $-k \rightarrow k$ with $k = x, y, z$ often feature negative coupling coefficients causing negative outflow face-averaged fluxes for intermediate and thick optical cells.
- Results from Lathrop's Test Problem:
  - Odd order methods: AHOTN-1,3, DGLA-1,3, and HODD-0,2 for example are more prone to developing negative fluxes.
  - HODD features larger τ_k^w with $k = \psi, \phi$ than AHOTN.
  - LL/LN perform very similarly to AHOTN-1.
  - SCB is the most resilient against negative fluxes with negative fluxes hardly occurring during the whole Lathrop test case study.
  - Following SCB, AHOTN-2, DGLA-2, and DGC-2 are least prone to negative fluxes.
- A first collision source is tested as a remedy for negative fluxes. The rationale for using a first collided flux is that a larger, positive source within some region will always increase the flux level and mitigate the occurrence of negative cell-averaged face fluxes. Because of the ray-tracing procedure utilized for computing the uncollided flux, the first collision

source is positive. It is found that using the first collision source reduces the negativity measure τ_ϕ^w by up to three orders of magnitude, but does not eliminate negative fluxes completely.

Thick Diffusion Limit: Adams' analysis applied to DGFEM methods of order one[7] was extended to WDD type methods of order up to one: DD, HODD-1, AHOTN-0, AHOTN-1, LL, and LN. The analysis shows that AHOTN-1 has a diffusion limit, while none of the other methods possess it. Numerical results using the thick diffusion test case are obtained for all contending methods up to order $\Lambda = 3$. The results are:

- DGLA methods possess the diffusion limit except for $\Lambda = 0$.
- DGC methods do not possess the diffusion limit for any Λ.
- AHOTN methods possess the diffusion limit except for $\Lambda = 0$.
- HODD methods do not possess the diffusion limit for any Λ.
- LL and LN do not possess the diffusion limit.

Wherever analysis is available, numerical results corroborate the analysis.

Construction of a Quantitative Decision Metric:

The decision metric proved to be a useful utility for selecting methods given a certain weighted list of performance aspects for many validation test cases. For some cases its predictions were not optimal in the sense that the true best performer was not identified correctly or even that the first three suggestions did not include the best performer. However, the decision metric never suggested using a detrimentally bad performer either, i.e. it never suggested using the actual worst or second worst discretization scheme. In addition, even under the worst circumstances, among the first three suggestions of the decision metric was at least one that came within 15% of the best performer's score.

For improving the accuracy of the decision metric three remedies are suggested:

- Increase the amount of data used to make decisions. Enable aggregation of data from different tests but pertaining to the same performance aspect, for example accuracy.
- Make a more accurate classification of target quantities. For example, region-averaged fluxes could be classified by size compared to domain size and their position with respect to the boundary. This approach might preclude the logical contradiction of the decision score mentioned earlier.

- Enable accurate interpolation based on all important macroscopic quantities between models to match suitable data sets to the problem that performances are to be predicted for.

7.2 Conclusion

The primary conclusion of this work is that the decision metric can be a useful utility compared to a purely qualitative assessment of the methods' performance. In most cases, the decision metric's suggestions are reasonable. Often the best performer is identified correctly, but even if it is not, typically the decision metric's top suggestion is within 10% of the actual best performer.

Improvements on the data basis, the aggregation of data from different sources, and a better classification of target quantities are expected to increase the reliability of the decision metric. Therefore, this work can only serve as a proof of principle that prediction of methods' performance for some problem of interest can be based on data from a set of different already solved problems. Clearly, the results are encouraging given the great agreement of prediction for some of the validation exercise cases. Especially for cases where multiple performance aspects are being optimized, the guidance could be very valuable. However, the potential user of the decision metric should keep in mind that its predictions are afflicted with an inherent uncertainty.

In addition to the development of the quantitative decision metric, an MMS test suite was developed that may serve as a test problem for new spatial discretization methods. It is implemented in the code MMS3D, which will be made available to the public soon. Its features are an efficient and accurate tracking procedure and accurate analytical integration routines that allow for an arbitrary expansion order of the discretization methods receiving the data. The test suite itself allows for creation of an exact solution with an arbitrary order of non-smoothness along with the associated distributed source and boundary conditions.

The most important conclusion from the numerical results is the inadequate accuracy and computational efficiency of standard discretization methods for C_0 problems. As a remedy, the SCT-Step method was implemented that resolves the lack of cell-wise convergence of standard methods for C_0 problems. However, the SCT-Step method suffers from the low quality of the Step approximation. If a higher-order methods is implemented in conjunction with the SCT algorithm, the resulting method will allow rates of convergence greater than or equal to two, overcoming one of the most limiting conditions imposed by the non-smoothness of the S_N equations' exact solution. A clear path to such a method is outlined within this work.

7.3 Future Work

There are some directions that appear worthy for further research. A list of these directions that is certainly not complete, is provided in the following:

- **Development of a high-order SCT method:** The underlying problem for deployment of high-order expansion methods for S_N problems is that they cannot play out their greatest benefit over higher-order methods: the increased rate of convergence. The SCT algorithm appears to provide the means to recover the theoretically predicted convergence rates for practical problems. A clear path to a high-order SCT has already been outlined before, so it shall not be repeated here. The potential benefit of a functional higher-order SCT methods is enormous because it might easily be the most efficient discretization method.

- **Increasing the data basis for decision metric:** The data basis used for computing the decision score employed within this work is not sufficient. First, the existing test problems need to be solved for a much greater variety of parameters. Second, additional test problems should be added to widen the scope of the data sets.

- **Interpolation between test problems:** In order to better match the data that the predicted scores are based on and the actual test case, interpolation of the data between values of macroscopic key parameters such as optical domain thickness, scattering ratios etc, is required. In addition, an exhaustive list of all relevant macroscopic key parameters needs to be compiled for each performance aspect.

- **Aggregation of data into the decision metric:** The single aspect scores are computed from only one test data set within this work. A way could be devised to combine data from various sets into a single aspect score in order to increase its reliability.

- **Adding more performance aspects to the decision metric:** So far only two aspects of performance were investigated: negativity and efficiency of the solution. Other aspects should could be added, for example the quality of the approximation in the thick diffusive limit. The thick diffusive limit was only discussed qualitatively within this work but a quantitative measure of a discretization method's performance in the thick diffusion limit could be devised.

REFERENCES

[1] J.I. Duo. *Error Estimate For Nodal and Short Characteristics Spatial Approximations of Two-Dimensional Discrete Ordinates Method.* PhD thesis, The Pennsylvania State University, State College, PA, USA, May 2008.

[2] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory.* Van Nostrand Reinhold Company, 1970.

[3] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis.* John Wiley and Sons, 1976.

[4] S. Chandrasekhar. *Radiative Transfer.* Oxford, 1950.

[5] M Krook. On the solutions of the equations of transfer. *Astrophys. J.*, 122:488–495, 1955.

[6] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Vol. I.* Wiley-Interscience, 1962.

[7] M. L. Adams. Discontinuous Finite Element transport solutions in thick diffusive problems. *Nuclear Science and Engineering*, 137:298–333, 2001.

[8] K. Thompson and M.L. Adams. A spatial discretization for solving the transport equation on unstructured grids of polyhedra. In *Proceedings of the International Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications.* American Nuclear Society, September 1999.

[9] K.D. Lathrop. Spatial differencing of the transport equation: Positivity vs. accuracy. *Journal of Computational Physics*, 4:475–498, 1969.

[10] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods.* Springer, 2008.

[11] Y. Wang and J.C. Ragusa. On the convergence of DGFEM applied to the discrete ordinates transport equation for structured and unstructured triangular meshes. *Nuclear Science and Engineering*, 163:56–72, 2009.

[12] Y. Wang and J.C. Ragusa. A high-order discontinuous galerkin method for the S_N transport equations on 2D unstructured triangular meshes. *Annals of Nuclear Energy*, 36:931–939, 2009.

[13] Y.Y Azmy. The weighted diamond difference form of nodal transport methods. *Nuclear Science and Engineering*, 98:29–40, 1988.

[14] Y.Y Azmy. Comparison of three approximations to the linear-linear nodal transport method in weighted diamond-difference form. *Nuclear Science and Engineering*, 100:190–200, 1988.

[15] A. Hebert. High-order diamond differencing schemes. *Annals of Nuclear Energy*, 33:1479–1488, 2006.

[16] N. Martin and A. Hebert. A three-dimensional S_N high-order diamond differencing discretization with a consistent acceleration scheme. *Annals of Nuclear Energy*, 36:1787–1796, 2009.

[17] J.I. Duo and Y.Y Azmy. Spatial convergence study of discrete ordinates methods via the singular characteristic tracking algorithm. *Nuclear Science and Engineering*, 162:41–55, 2009.

[18] E. W. Larsen. Spatial convergence properties of the diamond difference method in x,y geometry. *Nuclear Science and Engineering*, 80:710–713, 1982.

[19] Lingus. C. Analytical test cases for neutron and radiation transport codes. In *Proc. 2nd Conf. Transport Theory*, Los Alamos Scientific Laboratory, 1971.

[20] J Arkuszewski, T Kulikowska, and J Mika. Effect of singularities on approximation in S_N methods. *Nuclear Science and Engineering*, 49:20–26, 1972.

[21] Y.Y. Azmy. Error analysis of variations on Larsen's benchmark problem. In *ANS International Meeting on Mathematical Methods for Nuclear Applications*, Salt Lake City, UT, USA, September 2001. American Nuclear Society.

[22] J.I. Duo and Y.Y. Azmy. Error comparison of diamond difference, nodal and characteristic methods for solving multidimensional transport problems with the discrete ordinates approximation. *Nuclear Science and Engineering*, 156:139–153, 2007.

[23] E.E. Lewis and W.F. Miller Jr. *Computational Methods in Neutron Transport*. American Nuclear Society, Inc, 1993.

[24] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno. Denovo - a new three-dimensional parallel discrete ordinates code in SCALE. *Nuclear Technology*, 171(2):171–200, 2010.

[25] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[26] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-dependent Problems*. Siam, 2007.

[27] Randall J. LeVeque. *Finite Volume Methods for Hyberbolic Problems*. Cambridge University Press, 2002.

[28] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method : Its Basis and Fundamentals*. Oxford, 2004.

[29] S. Merton. A non-linear optimal discontinuous Petrov-Galerkin method for stabilising the solution of the transport equation. In *International Conference on Mathematics, Computational Methods and Reactor Physics.*, Saratoga Springs, USA, May 2009.

[30] Y.Y Azmy. Arbitrarily high order characteristic methods for solving the neutron transport equation. *Annals of Nuclear Energy*, 19:593–606, 1992.

[31] R.M. Ferrer. *An Arbitrarily High Order Transport Method of the Characteristic Type*. PhD thesis, The Pennsylvania State University, State College, PA, USA, May 2010.

[32] R.E. Alcouffe et al. Computational efficiency of numerical methods for the multigroup, discrete-ordinates neutron transport equations: The slab geometry case. *Nuclear Science and Engineering*, 71:111–127, 1979.

[33] E. W. Larsen and P. Nelson. Finite-difference approximations and superconvergence for the discrete-ordinate equations in slab geometry. *Siam J. Numer. Anal.*, 19:334–348, 1982.

[34] Jr. H.D. Victory and K. Ganguly. On finite-difference methods for solving discrete-ordinates transport equations. *Siam J. Numer. Anal.*, 23:78–108, 1986.

[35] N.K. Madsen. Convergence of singular difference approximations for the discrete ordinate equations in x-y geometry. *Mathematics of Computation*, 117:45–50, 1972.

[36] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[37] L. et al. Gastaldo. High-order discrete ordinate transport in non-conforming 2D cartesian meshes. In *International Conference on Mathematics and Computational Methods and Reactor Physics*, Saratoga Springs, NY, May 2009. American Nuclear Society.

[38] P. Lesaint and P.A. Raviart. On a finite-element method for solving the neutron transport equation. *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–122, 1974.

[39] G.R. Richter. An optimal-order error estimate for the discontinuous galerkin method. *Math. Comput.*, 50:181, 1988.

[40] M.L. Adams. Subcell balance methods for radiative transfer on arbitrary grids. *Transport Theory and Statistical Physics*, 26:385–431, 1997.

[41] O.M. Zamonsky, G.C. Buscaglia, and Y.Y. Azmy. Improving the accuracy of high-order nodal transport methods. In *Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 1999. American Nuclear Society.

[42] W.F. Walters. The relation between finite element methods and nodal methods in transport theory. In *International Seminar in Finite Element and Allied Method for Reactor Physics and Shielding Calculations*, September 1985.

[43] W.F. Walters and R.D. O'Dell. A comparison of linear nodal, linear discontinuous, and diamond schemes for solving the transport equation in (x,y) geometry. In *Proceedings of the American Nuclear Society, ANS Winter Meeting*, volume 35. American Nuclear Society, October 1981.

[44] J.I. Duo, Y.Y. Azmy, and L.T. Zikatanov. A posteriori error estimator and AMR for discrete ordinates nodal transport methods. *Annals of Nuclear Energy*, 36:268–273, 2009.

[45] K. Salari and P. Knupp. Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Laboratories, 2000.

[46] C.J. Roy. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, 205:131–156, 2005.

[47] E. W. Larsen and Warren F. Miller, Jr. Convergence rates of spatial difference equations for the discrete-ordinates neutron transport equations in slab geometry. *Nuclear Science and Engineering*, 73:76–83, 1980.

[48] S. Schunert and Y.Y. Azmy. A two-dimensional method of manufactured solutions benchmark suite based on variations of Larsen's benchmark with escalating order of smoothness of the exact solution. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, May 2011. American Nuclear Society.

[49] S.D. Pautz. Verification of transport codes by the method of manufactured solutions: The Attila experience. Technical Report LA-UR-0101487, Los Alamos National Laboratory, 2001.

[50] S. Pautz and C. Drumm. Manufactured solution verification of the Ceptre code. In *Mathematics and Computation, Reactor Physics and Nuclear and Biological Applications.*, Avignon, France, 2005.

[51] C. Drumm. Order-convergence anomalies in second-order finite element transport methods. In *International Conference on Mathematics, Computational Methods and Reactor Physics.*, Saratoga Springs, NY, USA, 2009.

[52] B. Joe. Geompack - a software package for the generation of meshes using geometric algorithms. *Adv. Eng. Software*, 13:325–331, 1991.

[53] S. Schunert, D. Fournier, R. Le Tellier, and Y.Y. Azmy. Comparison of the accuracy of various spatial discretization schemes of the discrete ordinates equations in 2D cartesian geometry. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, May 2011. American Nuclear Society.

[54] F. Malvagi and G.C. Pomraning. Initial and boundary conditions for diffusive linear transport problems. *J. Math. Physics*, 32:805, 1991.

[55] E.W. Larsen. Initial and boundary conditions for diffusive linear transport problems. *Ann. Nucl. Energy*, 7:249–255, 1980.

[56] E.W. Larsen, J.E. Morel, and W.F. Miller JR. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *J. Comp. Physics*, 69:283–324, 1987.

[57] E.W. Larsen and J.E. Morel. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii. *J. Comp. Physics*, 83:212–236, 1989.

[58] W.A. Wieselquist. *The Quasidiffusion Method for Transport Problems on Unstructured Meshes*. PhD thesis, North Carolina State University, Raleigh, NC, USA, May 2009.

[59] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[60] R.E. Alcouffe et al. A time-dependent, parallel neutral particle transport code system. Technical report, Los Alamos National Laboratory, 2005.

[61] G.A. Baker. *Pade Approximants*. Cambridge University Press, 1996.

[62] X-5 Team Monte Carlo. *MCNP - A General N-Particle Transport Code, Version 5, Volume I: Overview and Theory*. Los Alamos National Laboratory, 2003.

[63] Q. Zhang and H.S. Abdel-Khalik. Global variance reduction for monte carlo reactor physics calculations. In *ANS Winter Meeting*. American Nuclear Society, Nov. 2012.

[64] Y. Wang. *Adaptive Mesh Refinement Solution Techniques for the Multigroup S_N Transport Equation Using Higher-order Discontinuous Finite Element Method*. PhD thesis, Texas A&M University, College Station, TX, USA1, May 2009.

[65] W. A. Rhoades and D. B. Simpson. The TORT three-dimensional discrete ordinates neutron/photon transport code. Technical report, Oak Ridge National Laboratory, 1997.

[66] R. L. Childs. Grtuncl: First collision source program, ornl informal notes. Technical report, Oak Ridge National Laboratory, 1982.

[67] Y.Y. Azmy. Benchmarking the accuracy of solution of 3-dimensional transport codes and methods over a range in parameter space. Technical report, Nuclear Energy Agency, 2007.

[68] K.B. Bekar and Y.Y. Azmy. TORT solutions to the nea suite of benchmarks for 3D transport codes and methods over a range in parameter space. *Ann. Nucl. Energy*, 36:368–374, 2009.

[69] L.F. Richardson. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London*, 210, 1911.

[70] K.F. Riley, M.P. Hobson, and S.J. Bence. *Mathematical Methods for Physics and Engineering*. Cambridge University Press, 2006.

[71] J. Berrut and L.N. Trefethen. Barycentric lagrange interpolation. *Siam Review*, 46:501–517, 2004.

[72] H. Pollard. The convergence almost everywhere of Legendre series. In *Proceedings of the American Mathematical Society*, volume 35. American Mathematical Socienty, October 1972.

APPENDICES

Appendix A

Special Functions

A.1 Legendre Polynomials

The Legendre polynomials[70] $P_l(s)$ with $l = 0, 1, 2, ...$ are fully defined via the three-point recursion relation:

$$(l+1)P_{l+1}(s) = (2l+1)sP_l(s) - lP_{l-1}(s), \tag{A.1}$$

and the first two Legendre polynomials:

$$\begin{aligned} P_0(s) &= 1 \\ P_1(s) &= s. \end{aligned}$$

The Legendre Polynomials are orthogonal on the interval $s \in [-1, 1]$:

$$\int_{-1}^{1} ds P_l(s) P_k(s) = \frac{2}{2l+1}\delta_{lk}, \tag{A.2}$$

and complete, i.e. any function on $s \in [-1, 1]$ can be expanded into a series of Legendre polynomials:

$$f(s) = \sum_{l=0}^{\infty} (2l+1) a_l P_l(s) \tag{A.3}$$

where the modes a_l can be obtained by

$$a_l = \frac{1}{2}\int_{-1}^{1} ds P_l(s) f(s). \tag{A.4}$$

The Legendre polynomials scaled to cell i denoted by $p_l^i(s)$ are defined by:

$$p_l^i(s) = P_l\left(2\frac{s-\frac{s_i+s_{i-1}}{2}}{s_i-s_{i-1}}\right), \tag{A.5}$$

where s_i and s_{i-1} are the upper and lower cell boundary in s direction. For convenience the cell index i might be dropped whenever possible. In analogy to the original Legendre polynomials the scaled Legendre polynomials can be used to expand an arbitrary function on the interval $[s_{i-1}, s_i]$ via:

$$f(s) = \sum_{l=0}^{\infty}(2l+1)a_l p_l^i(s) \tag{A.6}$$

$$a_l = \frac{1}{s_i - s_{i-1}}\int_{s_{i-1}}^{s_i} ds\ p_l^i(s)f(s). \tag{A.7}$$

A.2 Interpolation via Lagrange Polynomials

Following [71] the interpolation of order L uses an L-th order polynomial

$$f(s) = \sum_{l=0}^{L} b_l d_l(s), \tag{A.8}$$

to match the value of the function $f(s)$ at L distinct points within the domain. If we denote the interpolation points by s_l , $l = 0, 1, ...$ and select the $d_l(s)$ to be the lth Lagrange polynomial of order L defined by:

$$d_l(s) = \frac{\prod_{k=0,k\neq l}^{L}(s-s_k)}{\prod_{k=0,k\neq l}^{L}(s_l-s_k)}. \tag{A.9}$$

which features the following convenient property:

$$d_l(x_k) = \delta_{lk}, \tag{A.10}$$

then the b_l can be identified as the values of the interpolated functions at the interpolation points:

$$b_l = f(s_l). \tag{A.11}$$

In the FEM method Lagrange basis functions are used within the nodal basis, where the unknowns are the nodal values of the unknown angular flux. If the interpolated function is a polynomial of degree L then the interpolation of the same order is equivalent to the expan-

sion into Legendre polynomials truncated after the L-th order term. The modes and nodes are moreover related by the generalized Vandermonde matrix[10]. However, if the interpolated function is not a polynomial then the interpolation and truncated Legendre series will generally not coincide.

A.3 Relation between Continuous and Discrete $\mathcal{L}_2$ Norm

For the $\mathcal{L}_2$ norm error the discrete error norm can shown to be an approximation of the continuous error norm. To show this we write the continuous error norm as:

$$\|\epsilon_n\|^2_{c,\psi,2} = \sum_{n=1}^{N} w_n \sum_{\vec{i}} \int_{Q_{\vec{i}}} dV \left(\psi_n - \psi_n^h\right)^2. \tag{A.12}$$

Then we expand the approximate and the true solution within each cell into an infinite series of cell normalized Legendre polynomials $p_l^i(s)$. Several properties and the first few Legendre polynomials are reviewed in Sec. A.1. Due to the completeness of the Legendre polynomial the exact and approximate angular flux can be expanded as follows

$$\begin{aligned}
\psi_n^{\vec{i}} &= \sum_{m_x=0}^{\infty}\sum_{m_y=0}^{\infty}\sum_{m_z=0}^{\infty} (2m_x+1)(2m_y+1)(2m_z+1)\,\psi_{n,\vec{m}}^{\vec{i}} p_{m_x}^i(x)\, p_{m_y}^j(y)\, p_{m_z}^k(z) \\
&= \sum_{\vec{m}=0}^{\infty} (2m_x+1)(2m_y+1)(2m_z+1)\,\psi_{n,\vec{m}}^{\vec{i}} p_{\vec{m}}^{\vec{i}}(\vec{r}) \\
\psi_n^{h,\vec{i}} &= \sum_{m_x=0}^{\infty}\sum_{m_y=0}^{\infty}\sum_{m_z=0}^{\infty} (2m_x+1)(2m_y+1)(2m_z+1)\,\psi_{n,\vec{m}}^{h,\vec{i}} p_{m_x}^i(x)\, p_{m_y}^j(y)\, p_{m_z}^k(z) \\
&= \sum_{\vec{m}=0}^{\infty} (2m_x+1)(2m_y+1)(2m_z+1)\,\psi_{n,\vec{m}}^{h,\vec{i}} p_{\vec{m}}^{\vec{i}}(\vec{r}),
\end{aligned} \tag{A.13}$$

where $\psi_{n,\vec{m}}^{\vec{i}}$ and $\psi_{n,\vec{m}}^{h,\vec{i}}$ are the cell Legendre moments of the exact and approximate angular flux that can be computed by:

$$\begin{aligned}
\psi_{n,\vec{m}}^{\vec{i}} &= M_{\vec{m}}^{\vec{i}}\left\{\psi_n(\vec{r})\right\} \\
M_{\vec{m}}^{\vec{i}}\{\bullet\} &= \frac{1}{V^{\vec{i}}}\int_V dV p_{\vec{m}}^{\vec{m}}(\vec{r})\,\bullet\,.
\end{aligned} \tag{A.14}$$

For convenience the triple sums and the triple products of Legendre polynomials are abbreviated by:

$$\begin{aligned}\sum_{\vec{m}=0}^{\Lambda} \bullet &= \sum_{m_x=0}^{\Lambda}\sum_{m_y=0}^{\Lambda}\sum_{m_z=0}^{\Lambda} \bullet \\ p_{\vec{m}}^{\vec{i}}(\vec{r}) &= p_{m_x}^{i}(x)\, p_{m_y}^{j}(y)\, p_{m_z}^{k}(z)\,. \end{aligned} \tag{A.15}$$

Note, that even though the exact solution might be discontinuous **within** a cell, the Legendre polynomial expansion converges to the original function almost everywhere as long as it is square integrable[72]. Substituting the Legendre Polynomial expansions Eqs. A.13 into the $\mathcal{L}_2$ norm expression Eqs. A.12 we obtain:

$$\begin{aligned}\|\epsilon_n\|_{c,\psi,2}^2 = \sum_{n=1}^{N} w_n \sum_{\vec{i}} \int_{\mathrm{Q}_{\vec{i}}} dV \\ \left(\left[\sum_{\vec{m}=0}^{\infty} \psi_{n,\vec{m}}^{\vec{i}} p_{\vec{m}}(\vec{r})\right]^2 + \left[\sum_{\vec{m}=0}^{\infty} \psi_{n,\vec{m}}^{h,\vec{i}} p_{\vec{m}}(\vec{r})\right]^2 - \left[\sum_{\vec{m}=0}^{\infty} \psi_{n,\vec{m}}^{\vec{i}} p_{\vec{m}}(\vec{r})\right]\left[\sum_{\vec{m}=0}^{\infty} \psi_{n,\vec{m}}^{h,\vec{i}} p_{\vec{m}}(\vec{r})\right] \right). \end{aligned} \tag{A.16}$$

Multiplying out the sums and using the orthogonality of the Legendre polynomials Eq. A.16 can be simplified to:

$$\|\epsilon_n\|_{c,\psi,2}^2 = \sum_{n=1}^{N} w_n \sum_{\vec{i}} \left[\sum_{\vec{m}=0}^{\infty} V^{\vec{i}} \frac{\left(\psi_{n,\vec{m}}^{\vec{i}} - \psi_{n,\vec{m}}^{h,\vec{i}}\right)^2}{(2m_x+1)(2m_y+1)(2m_z+1)} \right] \tag{A.17}$$

Now we separate out the $\vec{m}=0$ contribution from the summation and since $\vec{\epsilon}_n^{\vec{i}} = \psi_{n,0}^{\vec{i}} - \psi_{n,0}^{h,\vec{i}}$ we can write:

$$\|\epsilon_n\|_{c,\psi,2}^2 = \|\epsilon_n\|_{d,\psi,2}^2 + \sum_{n=1}^{N} w_n \sum_{\vec{i}} \left[\sum_{\vec{m}=0}^{\infty} V^{\vec{i}} \frac{\left(\psi_{n,\vec{m}}^{\vec{i}} - \psi_{n,\vec{m}}^{h,\vec{i}}\right)^2}{(2m_x+1)(2m_y+1)(2m_z+1)} (1-\delta_{\vec{m}}) \right], \tag{A.18}$$

where $\delta_{\vec{m}} = \delta_{m_x,m_y}\delta_{m_x,m_z}$ is given in terms of Kronecker deltas. The continuous $\mathcal{L}_2$ norm in Eq. A.18 can thus be separated into the the discrete $\mathcal{L}_2$ norm plus higher Legendre moment error norms. Therefore, the discrete $\mathcal{L}_2$ error norm can be seen as a truncated continuous $\mathcal{L}_2$ error norm. However, the same is not true in general for continuous and discrete $\mathcal{L}_p$ error norms.

For a C_1 test case with $\sigma_t = 1$ and $c = 0.2$ the error vs. mesh size h is depicted in Fig. A.1 for various truncation orders λ. λ is the maximum order up to which the second summation in

Eq. A.18 is performed.

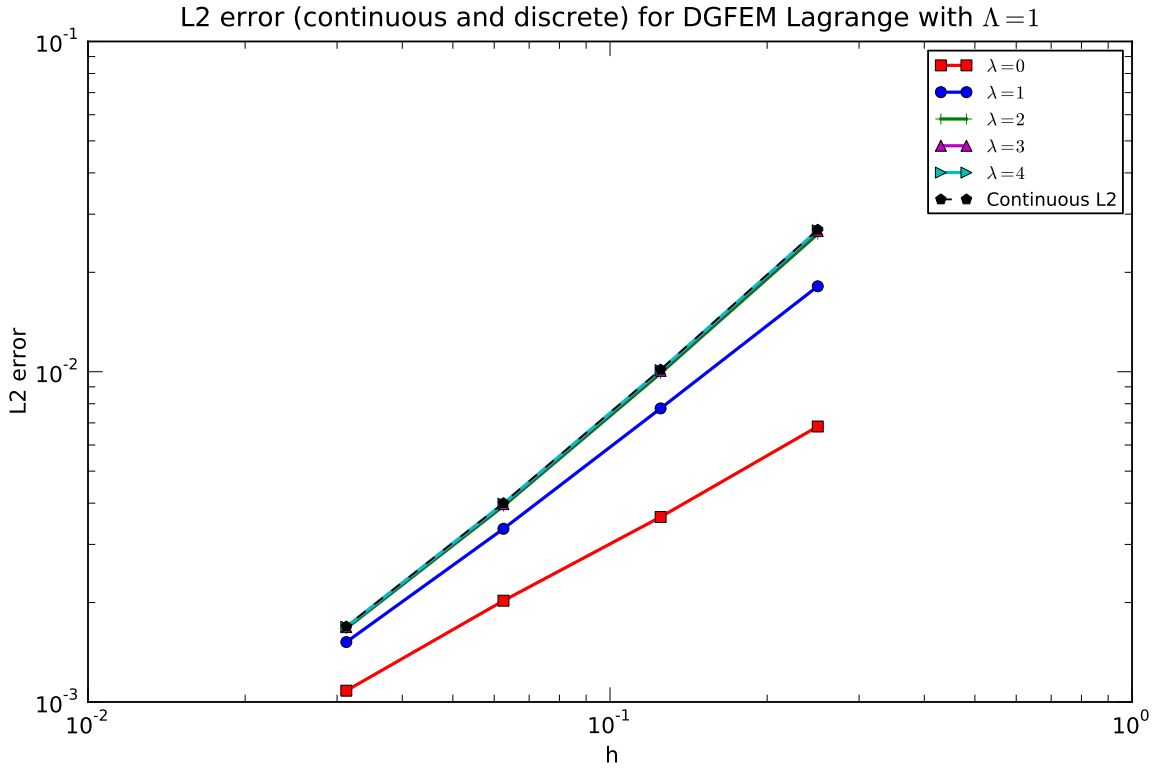


Figure A.1: "Approximation" of continuous L_2 norm by the sum in Eq. A.18 for different maximum orders λ up to which the second summation is performed.

Appendix B

Spatial Discretization Schemes

B.1 HODD Derivation via Interpolation

The HODD relations by Hebert can be derived by assuming flux shape Eq. 2.26 which is required to reproduce the cell face moments $\psi^h_{F,\vec{m}^t}$ $t = x, y, z$ and $\psi^h_{\vec{m}}$. For demonstration it is sufficient to show the derivation along a single spatial dimension x, y or z because the other two dimensions follow by analogy. We select to derive the HODD relations along the x dimension relating the east and west face Legendre moments. The following quantities need to be reproduced by Eq. 2.26:

$$\begin{aligned}
\psi^h_{W,\vec{m}^x} &= \frac{1}{\Delta y \Delta z} \int_{y_{j-1}}^{y_j} dy \int_{z_{k-1}}^{z_k} dz \; p_{m_y}(y) p_{m_z}(z) \psi^h(x_{i-1}, y, z), \; m_y, m_z = 0, ..., \Lambda \\
\psi^h_{E,\vec{m}^x} &= \frac{1}{\Delta y \Delta z} \int_{y_{j-1}}^{y_j} dy \int_{z_{k-1}}^{z_k} dz \; p_{m_y}(y) p_{m_z}(z) \psi^h(x_i, y, z), \; m_y, m_z = 0, ..., \Lambda \\
\psi^h_{\vec{m}} &= \frac{1}{\Delta x \Delta y \Delta z} \int_{x_{i-1}}^{x_i} dx \int_{y_{j-1}}^{y_j} dy \int_{z_{k-1}}^{z_k} dz \; p_{m_x}(x) p_{m_y}(y) p_{m_z}(z) \psi^h(x, y, z), \\
& \qquad m_x, m_y, m_z = 0, ..., \Lambda, \quad \text{(B.1)}
\end{aligned}$$

Substituting Eq. 2.26 into Eq. B.1 gives the following expressions for $\psi^h_{W,\vec{m}^x}$, $\psi^h_{E,\vec{m}^x}$ and $\psi^h_{\vec{m}}$:

$$\begin{aligned}
\psi^h_{W,\vec{m}^x} &= \frac{1}{(2m_y+1)(2m_z+1)} \sum_{m_x}^{\Lambda+1} (-1)^{m_x} \alpha_{\vec{m}}, \; m_y, m_z = 0, ..., \Lambda \\
\psi^h_{E,\vec{m}^x} &= \frac{1}{(2m_y+1)(2m_z+1)} \sum_{m_x}^{\Lambda+1} \alpha_{\vec{m}}, \; m_y, m_z = 0, ..., \Lambda \\
\psi^h_{\vec{m}} &= \frac{\alpha_{\vec{m}}}{(2m_x+1)(2m_y+1)(2m_z+1)}, \; m_x, m_y, m_z = 0, ..., \Lambda. \quad \text{(B.2)}
\end{aligned}$$

In order to relate the west and the east face moments we compute $\psi^h_{E,\vec{m}^x} - (-1)^{\Lambda+1}\psi^h_{W,\vec{m}^x}$ such that $\alpha_{\Lambda+1,m_y,m_z}$ is canceled out because it is the only $\alpha_{\vec{m}}$ that does not directly correspond to a cell Legendre moment. We obtain the following relationships:

$$\begin{aligned}
\Lambda \text{ even: } \psi^h_{E,\vec{m}^x} + \psi^h_{W,\vec{m}^x} &= 2 \sum_{m_x=0,even}^{\Lambda} \frac{\alpha_{\vec{m}}}{(2m_y+1)(2m_z+1)} \\
\Lambda \text{ odd: } \psi^h_{E,\vec{m}^x} - \psi^h_{W,\vec{m}^x} &= 2 \sum_{m_x=1,odd}^{\Lambda} \frac{\alpha_{\vec{m}}}{(2m_y+1)(2m_z+1)},
\end{aligned} \tag{B.3}$$

and by Eq. B.2 we obtain the HODD auxiliary relation along the x dimension:

$$\begin{aligned}
\Lambda \text{ even: } \psi^h_{E,\vec{m}^x} + \psi^h_{W,\vec{m}^x} &= 2 \sum_{m_x=0,even}^{\Lambda} (2m_x+1)\,\psi^h_{\vec{m}} \\
\Lambda \text{ odd: } \psi^h_{E,\vec{m}^x} - \psi^h_{W,\vec{m}^x} &= 2 \sum_{m_x=1,odd}^{\Lambda} (2m_x+1)\,\psi^h_{\vec{m}}.
\end{aligned} \tag{B.4}$$

B.2 Evaluated TLD Matrices for Lagrange Interpolation Polynomials

TLD equations generated with Lagrange interpolation functions.
Numbering of unknowns as in Fig. 3.6

Define the function space first

```
lagrange = {{(xu-x)/(xu-xl), (yu-y)/(yu-yl), (zu-z)/(zu-zl)},
            {(x-xl)/(xu-xl), (yu-y)/(yu-yl), (zu-z)/(zu-zl)},
            {(x-xl)/(xu-xl), (y-yl)/(yu-yl), (zu-z)/(zu-zl)},
            {(xu-x)/(xu-xl), (y-yl)/(yu-yl), (zu-z)/(zu-zl)},
            {(xu-x)/(xu-xl), (yu-y)/(yu-yl), (z-zl)/(zu-zl)},
            {(x-xl)/(xu-xl), (yu-y)/(yu-yl), (z-zl)/(zu-zl)},
            {(x-xl)/(xu-xl), (y-yl)/(yu-yl), (z-zl)/(zu-zl)},
            {(xu-x)/(xu-xl), (y-yl)/(yu-yl), (z-zl)/(zu-zl)}};
```

Mass Matrix normalized by 216 / V

```
mass = Simplify[216/((xu - xl) (yu - yl) (zu - zl)) Integrate[lagrange.Transpose[lagrange],
    {x, xl, xu}, {y, yl, yu}, {z, zl, zu}]]; MatrixForm[mass]
```

$$\begin{pmatrix} 8 & 4 & 2 & 4 & 4 & 2 & 1 & 2 \\ 4 & 8 & 4 & 2 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 1 & 2 & 4 & 2 \\ 4 & 2 & 4 & 8 & 2 & 1 & 2 & 4 \\ 4 & 2 & 1 & 2 & 8 & 4 & 2 & 4 \\ 2 & 4 & 2 & 1 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 2 & 4 & 8 & 4 \\ 2 & 1 & 2 & 4 & 4 & 2 & 4 & 8 \end{pmatrix}$$

Lumped Mass Matrix normalized by 216 / V

```
lmass = Simplify[216/((xu - xl) (yu - yl) (zu - zl))
   Integrate[Table[Flatten[lagrange] [[i]] KroneckerDelta[i, j], {i, 1, 8}, {j, 1, 8}],
    {x, xl, xu}, {y, yl, yu}, {z, zl, zu}]]; MatrixForm[lmass]
```

$$\begin{pmatrix} 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 \end{pmatrix}$$

Surface Matrix at East Surface normalized by face area and appropriate factor

```
lagx = lagrange /. {x → xu};

surfE =
 Simplify[36/((yu - yl) (zu - zl)) Integrate[lagx.Transpose[lagx], {y, yl, yu}, {z, zl, zu}]];
MatrixForm[surfE]
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 0 & 0 & 2 & 1 & 0 \\ 0 & 2 & 4 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 4 & 2 & 0 \\ 0 & 1 & 2 & 0 & 0 & 2 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Lumped Surface Matrix at East Surface normalized by face area and appropriate factor

```
lsurfE =
 Simplify[36/((yu - yl) (zu - zl)) Integrate[Table[Flatten[lagx] [[i]] KroneckerDelta[i, j],
     {i, 1, 8}, {j, 1, 8}], {y, yl, yu}, {z, zl, zu}]]; MatrixForm[lsurfE]
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Stiffness matrix x-derivative normalized by face area and appropriate factor

```
stiffx = Simplify[72/((yu - yl) (zu - zl)) Integrate[D[lagrange, x].Transpose[lagrange],
    {x, xl, xu}, {y, yl, yu}, {z, zl, zu}]]; MatrixForm[stiffx]
```

$$\begin{pmatrix} -4 & -4 & -2 & -2 & -2 & -2 & -1 & -1 \\ 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 4 & 4 & 1 & 1 & 2 & 2 \\ -2 & -2 & -4 & -4 & -1 & -1 & -2 & -2 \\ -2 & -2 & -1 & -1 & -4 & -4 & -2 & -2 \\ 2 & 2 & 1 & 1 & 4 & 4 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 \\ -1 & -1 & -2 & -2 & -2 & -2 & -4 & -4 \end{pmatrix}$$

Lumping of the stiffness matrix normalized by surface area and appropriate factor

```
dlagx = Flatten[D[lagrange, x]];

lag   = Flatten[lagrange];
```

```
stiffI = Table[ 0 , {row, 1, 8}, {col, 1, 8}];
```

$$r = \begin{pmatrix} xl & yl & zl \\ xu & yl & zl \\ xu & yu & zl \\ xl & yu & zl \\ xl & yl & zu \\ xu & yl & zu \\ xu & yu & zu \\ xl & yu & zu \end{pmatrix};$$

```
Do[ b[x_, y_, z_] = dlagx[[col]] ;
  xc = r[[row, 1]]; yc = r[[row, 2]]; zc = r[[row, 3]];
  stiffI[[row, col]] = b[xc, yc, zc] Integrate[lag[[row]],
     {x, xl, xu}, {y, yl, yu}, {z, zl, zu}] , {row, 1, 8}, {col, 1, 8}];

lagxu = lagrange /. {x → xu};

lagxl = lagrange /. {x → xl};

El = Simplify[Integrate[Table[Flatten[lagxl] [[i]] KroneckerDelta[i, j],
     {i, 1, 8}, {j, 1, 8}], {y, yl, yu}, {z, zl, zu}]];

Eu = Simplify[Integrate[Table[Flatten[lagxu] [[i]] KroneckerDelta[i, j],
     {i, 1, 8}, {j, 1, 8}], {y, yl, yu}, {z, zl, zu}]];

lstiffx = - stiffI - El + Eu;

Simplify[ 8 / ((yu - yl) (zu - zl)) lstiffx ] // MatrixForm
```

$$\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

B.3 Evaluation of Pade Coefficients for Spatial Weights

Definition of the Spatial Weights

▪ General Definition of the weight α

ϵ are the projections of the exponentials on Legendre Polynomials as defined in the AHOT - N WDD paper

$$\epsilon[t\_, i\_] := \frac{2i+1}{2}\int_{-1}^{1} \text{Exp}[t\ x]\ \text{LegendreP}[i, x]\ dx$$

Define Numerator as in the WDD paper

```
Num[t_, Λ_] := (Cosh[t] - Sum[ϵ[t, i], {i, 0, Λ, 2}])
```

Define denominator accordingly:

```
Den[t_, Λ_] := (Sinh[t] - Sum[ϵ[t, i], {i, 1, Λ, 2}])
```

α is the ratio of these two. Note that for small t I expand using a Pade approximant around t=0 of order (10,10) to supress noise.

```
α0[t_] =
  Piecewise[{{PadeApproximant[Simplify[Num[t, 0] / Den[t, 0]], {t, 0, 10}], t < 1 / 10},
    {Simplify[Num[t, 0] / Den[t, 0]], t ≥ 1 / 10}}];

α1[t_] =
  Piecewise[{{PadeApproximant[Simplify[Num[t, 1] / Den[t, 1]], {t, 0, 10}], t < 1 / 10},
    {Simplify[Num[t, 1] / Den[t, 1]], t ≥ 1 / 10}}];

α2[t_] =
  Piecewise[{{PadeApproximant[Simplify[Num[t, 2] / Den[t, 2]], {t, 0, 10}], t < 1 / 10},
    {Simplify[Num[t, 2] / Den[t, 2]], t ≥ 1 / 10}}];

α3[t_] =
  Piecewise[{{PadeApproximant[Simplify[Num[t, 3] / Den[t, 3]], {t, 0, 10}], t < 1 / 10},
    {Simplify[Num[t, 3] / Den[t, 3]], t ≥ 1 / 10}}];
```

Determine Local Expansion of AHOTN Weight - arbitrary grid spacing

▪ Get list of local expansions coefficient

Set the order here

```
order = 0;
```

coeff contains the expansion coefficients: I use a Pade approximation of order (1,1) and a grid with constant spacing of

Δ so I will have (tm/ Δ)+1 grid points between 0 and tm. Note that the expansion coefficients can be very easily extracted
from the list coeff by a simple round operation and a subsequent Min((tm/ Δ)+1,Round(t/Δ)+1). Then the weight
α can be computed with 4 fetches, 2 multiplications, 2 additions and a division. It should therefore be very cheap. Note that
computing the table coeff takes some time though, but for the final implementation in fortran it will be read from a file at the
beginning of the code's execution and stored in an array.

```
Δ = 1 / 100; tm = 200;

coeff = Table[0, {i, 0, tm / Δ}, {j, 1, 4}];

Block[{$MaxExtraPrecision = 1000}, Do[
  num = N[CoefficientList[Numerator[PadeApproximant[
      Simplify[Num[t, order] / Den[t, order]], {t, i Δ, {1, 1}}]], t], 25];
  den = N[CoefficientList[Denominator[PadeApproximant[
      Simplify[Num[t, order] / Den[t, order]], {t, i Δ, {1, 1}}]], t], 25];
  coeff[[i + 1, 1]] = num[[1]];
  coeff[[i + 1, 2]] = If[Length[num] > 1, num[[2]], 0];
  coeff[[i + 1, 3]] = den[[1]];
  coeff[[i + 1, 4]] = If[Length[den] > 1, den[[2]], 0];, {i, 0, tm / Δ}]]
```

```
$Aborted
```

Define approximated α as function αPade

```
αPade[t_] := (i = Min[{Round[t / Δ], 1000}]; a = coeff[[i + 1, 1]];
   b = coeff[[i + 1, 2]]; c = coeff[[i + 1, 3]]; d = coeff[[i + 1, 4]]; N[(a + b t)/(c + d t), 16]);

Export["/home/sebastian/PhDWork/ThreeDimensional/Discretization-Schemes/AHOTN/AHOTN/
   Approximation_Weights/w3raw.dat", coeff, "Data"]
```

B.4 Fine Mesh Limit of AHOTN

In this section it going to be demonstrated that the AHOTN WDD equations limit to the HODD equations for vanishing cell size. To this end first recall the asymptotic behavior of the AHOTN spatial weights Eq. 2.50, i.e. for $\Lambda \in$ even, $\alpha \to 0$ and $\Lambda \in$ odd, $\alpha \to \infty$. Given that the only difference of the AHOTN method and the HODD method is in the auxiliary relations we only need to show that Eq. 2.48 limits to Eq. 2.24 or 2.25 for Λ even or odd, respectively. For the even case it is readily seen that:

$$\begin{aligned} \lim_{\alpha_{n,x}\to 0} \quad & \left[\frac{1+\alpha_{n,x}}{2}\psi^h_{E,\vec{m}^x} + \frac{1-\alpha_{n,x}}{2}\psi^h_{W,\vec{m}^x} = \sum_{m_x=0,even}^{\Lambda} (2m_x+1)\psi^h_{\vec{m}} + \sum_{m_x=0,odd}^{\Lambda} (2m_x+1)\alpha_{n,x}\psi^h_{\vec{m}} \right] \\ \to \quad & \left[\frac{1}{2}\psi^h_{E,\vec{m}^x} + \frac{1}{2}\psi^h_{W,\vec{m}^x} = \sum_{m_x=0,even}^{\Lambda} (2m_x+1)\psi^h_{\vec{m}} \right], \end{aligned} \tag{B.5}$$

which is identical to the HODD auxiliary relations in x-directions Eq. 2.24 with equivalent limits applying for the y and z-direction. For the $\Lambda \in$ odd Eq. 2.48 is divided by $\alpha_{n,x}$ and the limit $\alpha_{n,x} \to \infty$ is applied:

$$\begin{aligned} \lim_{\alpha_{n,x}\to\infty} & \left[\left(\frac{1}{2\ \alpha_{n,x}} + \frac{1}{2} \right) \psi^h_{E,\vec{m}^x} + \left(\frac{1}{2\ \alpha_{n,x}} - \frac{1}{2} \right) \psi^h_{W,\vec{m}^x} \right. \\ & \left. = \sum_{m_x=0,even}^{\Lambda} \frac{(2m_x+1)}{\alpha_{n,x}}\psi^h_{\vec{m}} + \sum_{m_x=0,odd}^{\Lambda} (2m_x+1)\psi^h_{\vec{m}} \right] \\ & \to \left[\frac{1}{2}\psi^h_{E,\vec{m}^x} - \frac{1}{2}\psi^h_{W,\vec{m}^x} = \sum_{m_x=1,odd}^{\Lambda} (2m_x+1)\psi^h_{\vec{m}} \right], \end{aligned} \tag{B.6}$$

which is identical to the HODD auxiliary relations in x-directions Eq. 2.25 with equivalent limits applying for the y and z-direction. This completes the proof that AHOTN-Λ limits to the HODD-Λ method for vanishing mesh size.

B.5 Equivalence of DGFEM and WDD for Two-Dimensional Geometries

The BPD method in two-dimensional geometry approximates the flux within each cell by:

$$\psi^h(x,y) = \sum_{l_x=0}^{\Lambda} \sum_{l_y=0}^{\Lambda} (2l_x+1)\,(2l_y+1)\,\psi^h_{\vec{l}} p_{l_x}(x) p_{l_y}(y), \tag{B.7}$$

In addition let the numerical flux on the four edges W, E, S and N be formally expanded in terms of Legendre polynomials even though by the virtue of the numerical upstreaming flux we already prescribed that the pertaining expansion coefficients on the inflow edges are known from solution the previous cell while the coefficients on the outflow edges are coupled to the nodal expansion coefficients by requiring that the interior and exterior traces are equal:

$$\psi_F^h(s) = \sum_{l_s=0}^{\Lambda} (2l_s+1)\,\psi_{F,l_s}^h p_{l_s}(s),\ \ s \text{ either } x \text{ or } y. \tag{B.8}$$

For simplicity let us restrict $\mu_n > 0$ and $\eta_n > 0$ such that east and north are outflow faces and west and south are inflow faces. The trial functions Eq. B.7 are then substituted into the weak formulation Eq. 2.12 and tested with $p_{\vec{m}}(x,y) = p_{m_x}(x)p_{m_y}(y),\ \ m_x, m_y = 0, ..., \Lambda$:

$$\begin{aligned} &\mu_n \left\{ \left\langle \psi_E^h(y), p_{m_y}(y) \right\rangle_E - (-1)^{m_x} \left\langle \psi_W^h(y), p_{m_y}(y) \right\rangle_W \right\} \\ &\eta_n \left\{ \left\langle \psi_N^h(x), p_{m_x}(x) \right\rangle_N - (-1)^{m_y} \left\langle \psi_S^h(x), p_{m_x}(x) \right\rangle_S \right\} \\ -\mu_n \left(\frac{\partial p_{\vec{m}}}{\partial x}, \psi^h(x,y) \right) &- \eta_n \left(\frac{\partial p_{\vec{m}}}{\partial y}, \psi^h(x,y) \right) + \sigma_t \left(p_{\vec{m}}, \psi^h(x,y) \right) = \left(p_{\vec{m}}, S^h(x,y) \right) \end{aligned} \tag{B.9}$$

Evaluating the integrals and dividing through by the cell area $\Delta x_i \cdot \Delta y_j$ results in the Legendre moments of the transport equation Eq. 2.23. As discussed several times before these equations have more unknowns than equations such that closure relations need to be found. However, by choosing a numerical upstream flux we already have devised the closure: The expansion coefficients on the inflow edges are known from the respective upstream cells, i.e. $\psi_{W,l_y}^{h,\vec{i}} = \psi_{E,l_y}^{h,\vec{i}-\hat{e}_1}$ and $\psi_{S,l_x}^{h,\vec{i}} = \psi_{N,l_x}^{h,\vec{i}-\hat{e}_2}$ and the flux on the outflow faces satisfies:

$$\begin{aligned} \psi_N^h(x) &= \psi^h(x, y_j) \\ \psi_E^h(y) &= \psi^h(x_i, y). \end{aligned} \tag{B.10}$$

Applying the operator $\langle \bullet, p_{m_s}(s) \rangle_F$ to both sides of Eqs. B.10 gives the following relationship of the of the nodal and edge flux expansion coefficients:

$$\begin{aligned} \psi_{N,m_x}^h &= \sum_{l_y=0}^{\Lambda} (2l_y+1)\,\psi_{m_x,l_y}^h \\ \psi_{E,m_y}^h &= \sum_{l_x=0}^{\Lambda} (2l_x+1)\,\psi_{l_x,m_y}^h, \end{aligned} \tag{B.11}$$

which are the WDD closure relations Eq. 2.48 for $\alpha_x = \alpha_y = 1$. Thus, the per-cell set of equations for the two-dimensional BPD FEM comprises the Legendre moments augmented by the WDD closure relations with $\alpha_x = \alpha_y = 1$.

B.6 Flux Reconstruction in 2D Cartesian Geometry

Within this work frequently the approximated flux shape $\psi^h(\vec{r})$ is referenced. However, it was never stated how to exactly compute it from the unknowns that the described discretization methods actually compute, i.e. e.g. the nodal flux moments $\psi_{\vec{m}}^{h,\vec{i}}$. The flux shape within the domain is the direct sum of the cell flux shapes. This reduces the problem to explaining how to compute the flux shape $\psi^{h,\vec{i}}(\vec{r})$ within cell $\vec{i}$. The idea behind consistently computing the cell flux shapes is to reconstruct the flux within each cell using the trial space of the underlying finite element scheme and relating the unknown coefficients within the finite element trial space to the actually computed quantities.

B.6.1 HODD

For the 2D HODD and general μ_n, η_n the following trial space is utilized:

$$\psi^{h,\vec{i}}(\vec{r}) = \sum_{l_x,l_y=0}^{\Lambda+1} (2l_x+1)(2l_y+1)\, a_{l_x,l_y} P_{l_x}(\hat{x}) P_{l_y}(\hat{y}), \text{ with } a_{\Lambda+1,\Lambda+1} = 0, \tag{B.12}$$

where

$$\begin{aligned} \hat{x} &= 2\frac{\text{sign}\mu_n \; x - \frac{x_i - x_{i-1}}{2}}{\Delta x_i} \\ \hat{y} &= 2\frac{\text{sign}\eta_n \; y - \frac{y_j - y_{j-1}}{2}}{\Delta y_j}. \end{aligned} \tag{B.13}$$

The WDD algorithm solves for the nodal flux moments $\psi_{\vec{m}}^{h,\vec{i}}$ and the outflow face flux moments $\psi_{[E,W],m_y}^{h,\vec{i}}$ and $\psi_{[N,S],m_x}^{h,\vec{i}}$ defined by:

$$\begin{aligned} \psi_{m_x,m_y}^{h,\vec{i}} &= M_{\vec{m}} \left\{ \psi^{h,\vec{i}}(\vec{r}) \right\} \\ \psi_{F,m}^{h,\vec{i}} &= \frac{1}{\Delta x_F} \left\langle p_m(\vec{r}_F), \psi^{h,\vec{i}}(\vec{r}_F) \right\rangle_F, \; F \in \mathcal{E}^o. \end{aligned} \tag{B.14}$$

Substituting the trial space Eq. B.12 into Eqs. B.14 gives equations for computing the a_{l_x,l_y} from the nodal and outflow flux moments:

$$\begin{aligned} a_{m_x,m_y} &= (\text{sign}\mu_n)^{m_x} (\text{sign}\eta_n)^{m_y} \psi^{h,\vec{i}}_{m_x,m_y}, \text{ for } m_x, m_y = 0, ..., \Lambda \\ a_{\Lambda+1,m_y} &= \frac{\psi^h_{[E,W],m_y} (\text{sign}\eta_n)^{m_y} - \sum_{l_x}^{\Lambda} (2l_x+1)\, a_{l_x,m_y}}{2\Lambda+3} \\ a_{m_x,\Lambda+1} &= \frac{\psi^h_{[N,S],m_x} (\text{sign}\mu_n)^{m_x} - \sum_{l_y}^{\Lambda} (2l_y+1)\, a_{m_x,l_y}}{2\Lambda+3}. \end{aligned} \tag{B.15}$$

B.6.2 DGFEM (BPD)

For DGFEM in its bipolynomial discontinous modal representation with the trial space given by

$$\psi^{h,\vec{i}}(\vec{r}) = \sum_{l_x,l_y=0}^{\Lambda} (2l_x+1)(2l_y+1)\, a_{l_x,l_y} P_{l_x}(\hat{x}) P_{l_y}(\hat{y}), \tag{B.16}$$

the expansion coefficients are related to the nodal moments by:

$$a_{m_x,m_y} = (\text{sign}\mu_n)^{m_x} (\text{sign}\eta_n)^{m_y} \psi^{h,\vec{i}}_{m_x,m_y}, \text{ for } m_x, m_y = 0, ..., \Lambda. \tag{B.17}$$

Note, that one could suggest (similar to the HODD) to use the outflow face flux moments to achieve a "better" flux reconstruction. However, this idea does not work because the face flux moments in the DGFEM method are just linear combinations of the nodal moments (see e.g. B.5); once the nodal moments are determined the outflow face flux moments, independent of the inflow flux moments, are uniquely determined. Therefore, no additional information is contained in the outflow flux moments.

B.6.3 AHOTN

The reconstruction of the AHOTN flux shape requires the solution of a linear system of equations per mesh cell and is thus the most difficult. However, the idea for constructing the linear system of equations follows closely the approach taken for the HODD and DGFEM flux reconstruction. First, the trial space for two-dimensional Cartesian geometry and $\mu_n,\ \eta_n > 0$ (for

simplicity) is given by:

$$\begin{aligned}\psi^h(x,y) &= \sum_{l_x=-1}^{\Lambda}\sum_{l_y=0}^{\Lambda} a_{\vec{l}}\,\zeta_{l_x}(x)\,p_{l_y}(y) + \sum_{l_x=0}^{\Lambda}\sum_{l_y=-1}^{\Lambda} b_{\vec{l}}\,\zeta_{l_y}(y)\,p_{l_x}(x) \\ &+ \sum_{l_x=0}^{\Lambda}\sum_{l_y=0}^{\Lambda} c_{\vec{l}}\,p_{l_x}(x)\,p_{m_y}(y)\,, \end{aligned} \tag{B.18}$$

subject to the constraints:

$$\begin{aligned} M_{m_x,m_y}\left\{\sum_{l_x=-1}^{\Lambda}\sum_{l_y=0}^{\Lambda} a_{\vec{l}}\,\zeta_{l_x}(x)\,p_{l_y}(y)\right\} &= \frac{c_{m_x,m_y}}{(2m_x+1)(2m_y+1)} \\ M_{m_x,m_y}\left\{\sum_{l_x=0}^{\Lambda}\sum_{l_y=-1}^{\Lambda} b_{\vec{l}}\,\zeta_{l_y}(y)\,p_{l_x}(x)\right\} &= \frac{c_{m_x,m_y}}{(2m_x+1)(2m_y+1)}. \end{aligned} \tag{B.19}$$

Using the definitions of the nodal moments and the linearity of the $M_{\vec{m}}\{\bullet\}$ operator we can relate the $c_{\vec{l}}$ directly to the nodal moments:

$$\begin{aligned}\psi^h_{\vec{m}} &= M_{\vec{m}}\left\{\psi^h(x,y)\right\} = M_{\vec{m}}\left\{\sum_{l_x=-1}^{\Lambda}\sum_{l_y=0}^{\Lambda} a_{\vec{l}}\,\zeta_{l_x}(x)\,p_{l_y}(y)\right\} + M_{\vec{m}}\left\{\sum_{l_x=0}^{\Lambda}\sum_{l_y=-1}^{\Lambda} b_{\vec{l}}\,\zeta_{l_y}(y)\,p_{l_x}(x)\right\} \\ &+ M_{\vec{m}}\left\{\sum_{l_x=0}^{\Lambda}\sum_{l_y=0}^{\Lambda} c_{\vec{l}}\,p_{l_x}(x)\,p_{m_y}(y)\right\} = \frac{3c_{\vec{m}}}{(2m_x+1)(2m_y+1)}. \end{aligned} \tag{B.20}$$

Knowing $c_{\vec{l}}$, the constraints Eqs. B.19 can be used to obtain $2(\Lambda+1)^2$ equations for the unknown $a_{\vec{l}}$ and $b_{\vec{l}}$ expansion coefficients:

$$\begin{aligned} M_{\vec{m}}\left\{\sum_{l_x=-1}^{\Lambda}\sum_{l_y=0}^{\Lambda} a_{\vec{l}}\,\zeta_{l_x}(x)\,p_{l_y}(y)\right\} &= \sum_{l_x=-1}^{\Lambda} a_{l_x,m_y}\int_{x_{i-1}}^{x_i} dx\; p_{m_x}(x)\zeta_{l_x}(x) = \frac{\Delta x_i c_{\vec{m}}}{2m_x+1} \\ M_{\vec{m}}\left\{\sum_{l_y=-1}^{\Lambda}\sum_{l_x=0}^{\Lambda} b_{\vec{l}}\,\zeta_{l_y}(y)\,p_{l_x}(x)\right\} &= \sum_{l_y=-1}^{\Lambda} b_{m_x,l_y}\int_{y_{j-1}}^{y_j} dy\; p_{m_y}(y)\zeta_{l_y}(y) = \frac{\Delta y_j c_{\vec{m}}}{2m_y+1} \end{aligned} \tag{B.21}$$

The outstanding $2(\Lambda+1)$ equations can be obtained by matching the E and N outflow face flux moments in the following manner:

$$\begin{aligned}\psi^h_{E,m_y} &= \frac{1}{\Delta y_j}\left\langle p_{m_y}(y), \psi^h\left(x_E, y\right)\right\rangle_E \\ \psi^h_{N,m_x} &= \frac{1}{\Delta x_i}\left\langle p_{m_x}(x), \psi^h\left(x, y_N\right)\right\rangle_N .\end{aligned} \tag{B.22}$$

Evaluating the expressions Eq. B.22 yields the following $2(\Lambda+1)$ equations:

$$\begin{aligned}\frac{1}{2m_y+1}\sum_{l_x=-1}^{\Lambda} a_{l_x,m_y}\zeta_{l_x}(1) &+ \frac{1}{\Delta y_j}\sum_{l_x=0}^{\Lambda}\sum_{l_y=-1}^{\Lambda} b_{\vec{l}}\int_{y_{j-1}}^{y_j} dy\; p_{m_y}(y)\zeta_{l_y}(y) \\ &= \psi^h_{E,m_y} - \sum_{l_x=0}^{\Lambda}\frac{c_{l_x,m_y}}{2m_y+1} \\ \frac{1}{2m_x+1}\sum_{l_y=-1}^{\Lambda} b_{m_x,l_y}\zeta_{l_y}(1) &+ \frac{1}{\Delta x_i}\sum_{l_y=0}^{\Lambda}\sum_{l_x=-1}^{\Lambda} a_{\vec{l}}\int_{x_{i-1}}^{x_i} dy\; p_{m_x}(x)\zeta_{l_x}(x) \\ &= \psi^h_{N,m_x} - \sum_{l_y=0}^{\Lambda}\frac{c_{m_x,l_y}}{2m_x+1}.\end{aligned} \tag{B.23}$$

The Eqs. B.21 and B.23 constitute a system of $2(\Lambda+1)^2+2(\Lambda+1)$ equations for the unknown $a_{\vec{l}}$ and $b_{\vec{l}}$.

Within this work the flux reconstruction is used for prolonging the flux from coarse onto fine meshes for computing type II error norms. In general, flux reconstruction can e.g. be used to devise consistent interpolation formulae within large cells or to restrict/prolong S_N solution in-between different spatial meshes (e.g. energy group dependent meshes).

Appendix C

Algorithms for the MMS3D Benchmark Suite

C.1 Semi-Symbolic Algorithms for Manipulation of Polynomials

Within the described work is was frequently necessary to manipulate polynomial expressions into more convenient forms. This section introduces the algorithms utilized to

1. Obtain the Legendre coefficients, i.e. the $\hat{p}_l^m$ in $P_m(x) = \sum_{l=0}^{m} \hat{p}_l^m x^l$.
2. Expand expressions like $(ax + by + cz + d)^m$.
3. Multiply two polynomials.

Polynomials are represented by their coefficient array which is a three-tupel saving the $c_{l,k,g}$ for a polynomial $f(x, y, z)$ given by:

$$f(x, y, z) = \sum_{l=0}^{L} \sum_{k=0}^{K} \sum_{g=0}^{G} c_{l,k,g} x^l y^k z^g. \tag{C.1}$$

The utilized algorithm's only approximation is the finite precision arithmetic used for the computation of the elements comprised in the coefficient array.

1. Legendre polynomial coefficients

The Legendre polynomial coefficients can be computed using Bonnet's recursion:

$$(m+1)P_{m+1}(x) = (2m+1)xP_m(x) - mP_{m-1}(x). \tag{C.2}$$

The actually implemented algorithm computing $\hat{p}_l^m$ for all $m \leq \Lambda$ is listed in algorithm 3.

Algorithm 3 Bonnet's recursion

Matrix $\underline{P}$ consisting of the elements $p_{m,l}$ stores the Legendre coefficients p_l^m.
$\underline{P} = 0$
$p_{0,0} = 1$
if $m > 0$ **then**
  $p_{1,1} = 1$
end if
for $m = 2$ to Λ **do**
  for $l = 0$ to $m - 2$ **do**
    $p_{m,l} \leftarrow p_{m,l} + \frac{m-1}{m} p_{m-2,l}$
  end for
  for $l = 0$ to $m - 1$ **do**
    $p_{m,l+1} \leftarrow p_{m,l+1} + \frac{2m-1}{m} p_{m-1,l}$
  end for
end for

2. Expand via Binomial Theorem

By using the binomial theorem repeatedly the expression:

$$f(x, y, z) = (ax + by + cz + d)^m \tag{C.3}$$

can be expanded into the standard polynomial form:

$$(ax + by + cz + d)^m = \sum_{l=0}^{m} \sum_{k=0}^{m} \sum_{g=0}^{m} c_{l,k,g} x^l y^k z^g, \tag{C.4}$$

using the algorithm listed in 4.

Algorithm 4 Polynomials Expansion via Binomial Theorem

for $l = 0$ to m **do**
  for $k = 0$ to l **do**
    for $g = 0$ to k **do**
      $c_{m-l,l-k,k-g} = \begin{pmatrix} m \\ l \end{pmatrix} \begin{pmatrix} l \\ k \end{pmatrix} \begin{pmatrix} k \\ g \end{pmatrix} a^{m-l} b^{l-k} c^{k-g} d^g$
    end for
  end for
end for

3. Multiplication of Polynomials

Let polynomial f_1 and f_2 be defined as follows:

$$\begin{aligned} f_1 &= \sum_{l=0}^{L_1}\sum_{k=0}^{K_1}\sum_{g=0}^{G_1} a_{l,k,g}x^l y^k z^g \\ f_2 &= \sum_{l=0}^{L_2}\sum_{k=0}^{K_2}\sum_{g=0}^{G_2} b_{l,k,g}x^l y^k z^g. \end{aligned} \tag{C.5}$$

Then let f_3

$$f_3 = \sum_{l=0}^{L_1+L_2}\sum_{k=0}^{K_1+K_2}\sum_{g=0}^{G_1G_2} c_{l,k,g}x^l y^k z^g. \tag{C.6}$$

be the product of f_1 and f_2, $f_3 = f_1 \cdot f_2$. The algorithm 5 computes the appropriate coefficients $c_{l,k,g}$ and stores them in the three-tupel $\underline{C}$.

Algorithm 5 Multiplication of polynomials

```
1: C = 0
2: for l = 0 to L1 + L2 do
3:    for k = 0 to K1 + K2  do
4:       for g = 0 to G1 + G2  do
5:          for l1 = max(0, l − L2) to min(l, L1) do
6:             l2 = l − l1
7:             for k1 = max(0, k − K2) to min(k, K1) do
8:                k2 = k − k1
9:                for g1 = max(0, g − G2) to min(g, G1) do
10:                  g2 = g − g1
11:                  c_{l,k,g} ← c_{l,k,g} + a_{l1,k1,g1} b_{l2,k2,g2}.
12:               end for
13:            end for
14:         end for
15:      end for
16:   end for
17: end for
```

C.2 Integration of 1D Integrals

This section is concerned with evaluating the integral:

$$e^b \int_{\theta_{i_\theta-1}}^{\theta_{i_\theta}} d\theta p_{m_\theta}(\theta)\, \theta^{l_\theta} e^{a\theta} \tag{C.7}$$

To this end we make the substitution:

$$\theta = -\text{sign}(a)\,(\theta_{i_\theta} - \theta_{i_\theta-1})\,\hat{\theta} + \frac{1-\text{sign}(a)}{2}\theta_{i_\theta-1} + \frac{1+\text{sign}(a)}{2}\theta_{i_\theta}, \tag{C.8}$$

such that the integral transforms to:

$$\begin{aligned} & e^b \int_{\theta_{i_\theta-1}}^{\theta_{i_\theta}} d\theta p_{m_\theta}(\theta)\, \theta^{l_\theta} e^{a\theta} = (\theta_{i_\theta} - \theta_{i_\theta-1}) \exp\left[b + a\left(\frac{1-\text{sign}(a)}{2}\theta_{i_\theta-1} + \frac{1+\text{sign}(a)}{2}\theta_{i_\theta}\right)\right] \\ \times \quad & \int_0^1 d\hat{\theta}\left(-\text{sign}(a)\,(\theta_{i_\theta} - \theta_{i_\theta-1})\,\hat{\theta} + \frac{1-\text{sign}(a)}{2}\theta_{i_\theta-1} + \frac{1+\text{sign}(a)}{2}\theta_{i_\theta}\right)^{l_\theta} \\ \times \quad & P_{m_\theta}\left[-\text{sign}(a)\left(2\hat{\theta}-1\right)\right] e^{\left(-|a|(\theta_{i_\theta}-\theta_{i_\theta-1})\hat{\theta}\right)}. \end{aligned} \tag{C.9}$$

By first expanding the Legendre polynomials into the sum of monomials:

$$P_{m_\theta}(\theta) = \sum_{l}^{m_\theta} \hat{P}_l^{m_\theta} \theta^l. \tag{C.10}$$

and then using the algorithms outlined in section C.1 the integral Eq. C.9 can be transformed into:

$$e^b \int_{\theta_{i_\theta-1}}^{\theta_{i_\theta}} d\theta p_{m_\theta}(\theta)\, \theta^{l_\theta} e^{a\theta} = \sum_{k=0}^{l_\theta+m_\theta} \hat{c}_k e^{\hat{b}} I_k^{1D}(\hat{a}), \tag{C.11}$$

with the following definitions:

- $\hat{c}_k$: Polynomial coefficients from algorithms in section C.1.
- $\hat{b} = b + a\left(\frac{1-\text{sign}(a)}{2}\theta_{i_\theta-1} + \frac{1+\text{sign}(a)}{2}\theta_{i_\theta}\right)$
- $\hat{a} = |a|\,(\theta_{i_\theta} - \theta_{i_\theta-1})$
- $I_k^{1D}(\hat{a}) = \int_0^1 \hat{\theta}^k e^{-|\hat{a}|\hat{\theta}}$.

Thus, the problem of computing the integral in Eq. C.7 can be reduced to computing $I_k^{1D}(\hat{a})$ for $k = 0, ..., l_\theta + m_\theta$. A fast algorithm for computing the $I_k^{1D}(\hat{a})$ is to use forward/backward

recursion:

$$I_k(\hat{a}) = \begin{cases} \frac{e^{\hat{a}} - kI_{k-1}(\hat{a})}{a} & k \leq \lfloor |\hat{a}| \rfloor \\ \frac{\hat{a}}{k+1}\left(\frac{e^{\hat{a}}}{\hat{a}} - I_{k+1}(\hat{a})\right) & k > \lfloor |\hat{a}| \rfloor \end{cases} \tag{C.12}$$

which is stable for all values of k and $\hat{a}$. For the forward recursion $I_0(\hat{a})$ needs to be computed while for the backward recursion it is necessary to evaluate $I_{l_\theta + m_\theta}(\hat{a})$. From numerical experiment we found that for different combinations of k and $\hat{a}$ different approaches for the direct numerical evaluation of $I_k(\hat{a})$ yield the best accuracy:

$$I_k(\hat{a}) \leftarrow \begin{cases} \text{Taylor Series Expansion} & |\hat{a}| \leq 10 \\ \text{Direct analytical integration} & |\hat{a}| > 10,\ k \leq 40 \\ \text{Romberg integration} & |\hat{a}| > 10,\ k > 40 \end{cases}$$

Taylor series expansion

The exponential in $I_k(\hat{a})$ is expanded into a Taylor series and then the integration is performed:

$$I_k(\hat{a}) = \frac{1}{(k+1)} + \sum_{l=1}^{L} \underbrace{\frac{\hat{a}^l}{l!\,(l+k+1)}}_{e_l}, \tag{C.13}$$

with L such that $e_l < \epsilon$ and $|e_{l-1}| > |e_l|$. The latter condition is required since for $\hat{a} > 0$ e_l first grows with increasing l to some maximum value and then starts decaying.

Direct integration

$$I_k(\hat{a}) = \frac{e^{\hat{a}}}{\hat{a}}\left[\sum_{l=0}^{k} \frac{1}{|\hat{a}|^l} \frac{k!}{(k-l)!}\right] + \frac{1}{|\hat{a}|^{k+1}} k!. \tag{C.14}$$

Romberg integration

The algorithm is listed in algorithm 6. The Romberg integration uses a sequence of nested meshes starting with a coarse mesh and dividing the mesh width h in half at every refinement step. Then Richardson iteration is used to extrapolate to finer meshes thus increasing the order of accuracy at every subdivion step.

C.3 Integration of 2D Integrals

This section is concerned with the computation of integrals of the form given in Eq. 3.38:

$$e^b \iint_{A_s} dA p_{m_\omega}(\omega)\, p_{m_\theta}(\theta) \omega^{l_\omega} \theta^{l_\theta} \exp(a\theta). \tag{C.15}$$

Algorithm 6 One-dimensional Romberg integration

$r_{0,0} = \frac{1}{2}e^{\hat{a}}$
$h_l = 2^{-l}$
$g(x) = e^{\hat{a}x}x^k$
for $l = 1$ to l_{max} **do**
  $r_l = \frac{1}{2}r_{l-1,0} + h_l \sum_{q=1}^{2^{l-1}} g\left((2q-1)h_l\right)$
  for $q = 1$ to l **do**
    $r_{i,j} = \frac{1}{4^q-1}\left(4^q r_{l,q-1} - r_{l-1,q-1}\right).$
  end for
  if $|r_{l,l} - r_{l-1,l-1}| < \epsilon$ **then**
    $I_k\left(\hat{a}\right) = r_{l,l}$
    STOP
  end if
end for

For convenience we apply a change of variables $\left(\hat{\theta}, \hat{\omega}\right) \leftarrow (\theta, \omega)$ onto the unit triangle $0 \leq \hat{\omega} \leq 1$ and $0 \leq \hat{\theta} \leq \hat{\omega}$ such that the coordinates of the corner points are:

$$\hat{r}_1 = (0,0)^T,\ \hat{r}_2 = (1,0)^T, \hat{r}_3 = (1,1)^T. \tag{C.16}$$

Let the three corners of the triangle in the (θ, ω) coordinate system be denoted by $\vec{r}_i$ with $i = 1, 2, 3$, then the transformation from the $\left(\hat{\theta}, \hat{\omega}\right)$ coordinates to the (θ, ω) coordinates is given by:

$$\begin{aligned} \vec{r} &= \underline{J}\,\hat{r} + \vec{d} \\ \underline{J} &= \left[\vec{r}_2 - \vec{r}_1, \vec{r}_3 - \vec{r}_2\right],\ \vec{d} = \vec{r}_1, \end{aligned} \tag{C.17}$$

where $\underline{J}$ is the transformation Jacobian comprising the elements $j_{i,j}, i, j = 0, 1$ and $\vec{d} = (d_1, d_2)$ is the offset of the transformation. Applying the transformation Eq. C.17 to the integral Eq. C.15 the following integral can be obtained:

$$\begin{aligned} & e^b \iint_{A_s} dA p_{m_\omega}(\omega)\, p_{m_\theta}(\theta) \omega^{l_\omega} \theta^{l_\theta} \exp(a\theta) = |\underline{J}| \exp(b + ad_1) \\ \times\ & \int_0^1 d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta} \left[\left(j_{2,1}\hat{\theta} + j_{2,2}\hat{\omega} + d_2\right)^{l_\omega} \left(j_{1,1}\hat{\theta} + j_{1,2}\hat{\omega} + d_2\right)^{l_\theta} \right. \\ \times\ & P_{m_\omega}\left(\frac{2}{\Delta\omega_{i_\omega}}\left(j_{2,1}\hat{\theta} + j_{2,2}\hat{\omega} + d_2 - \omega_{i_\omega} + \omega_{i_\omega - 1}\right)\right) e^{-|a j_{1,2}|\hat{\omega}} \\ \times\ & \left. P_{m_\theta}\left(\frac{2}{\Delta\theta_{i_\theta}}\left(j_{1,1}\hat{\theta} + j_{1,2}\hat{\omega} + d_1 - \theta_{i_\theta} + \theta_{i_\theta - 1}\right)\right) e^{-|a j_{1,1}|\hat{\theta}} \right] \end{aligned} \tag{C.18}$$

Note that the argument within the exponentials is always negative which can be ensured by selecting the appropriate numbering of the corners of the original triangle in the (θ, ω) space: There is a total of three permutation for selecting which corner is associated with $\vec{r}_i,\ i = 1, 2, 3$; the first point is selected out of a choice of three and the other two are then numbered counter-clockwise. One of these permutations yields a transformation such that both $aj_{1,2} < 0,\ aj_{1,1} < 0$. Using the algorithms in section C.1 the integral Eq. C.18 can be recast in the simpler form:

$$e^b \iint_{A_s} dA p_{m_\omega}(\omega)\, p_{m_\theta}(\theta)\omega^{l_\omega}\theta^{l_\theta}\exp(a\theta) = |\underline{J}|\, e^{\hat{b}} \sum_{k_\omega=0}^{K}\sum_{k_\theta=0}^{K} \hat{c}_{k_\omega,k_\theta} I^{2D}_{k_\omega,k_\theta}\left(\hat{a}_{\hat{\omega}}, \hat{a}_{\hat{\theta}}\right) \tag{C.19}$$

where the following definitions are used:

- $K = l_\theta + l_\omega + m_\theta + m_\omega$
- $\hat{c}_{k_\omega,k_\theta}$: Polynomial coefficients from algorithms in section C.1.
- $\hat{b} = b + ad_1$
- $\hat{a}_{\hat{\theta}} = aj_{1,2}$
- $\hat{a}_{\hat{\omega}} = aj_{1,1}$
- $I^{2D}_{k_\omega,k_\theta}\left(\hat{a}_{\hat{\omega}}, \hat{a}_{\hat{\theta}}\right) = \int_0^1 d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta}\hat{\omega}^{k_\omega}\hat{\theta}^{k_\theta} e^{-|\hat{a}_{\hat{\omega}}|\hat{\omega}} e^{-|\hat{a}_{\hat{\theta}}|\hat{\theta}}$

Thus, the problem can be reduced to evaluating $I^{2D}_{l,q}(a,b)$ for $l, q = 0, ..., K$. For this purpose a two-dimensional forward-backward recursion is devised since simple forward substitution is only conditionally stable. The algorithm first computes $I^{2D}_{0,0}$, $I^{2D}_{K,0}$, $I^{2D}_{0,K}$ and $I^{2D}_{K,K}$ directly via a method described later, then a sweep in q direction is started for both $l = 0$ and $l = K$ using the following prescription:

$$I^{2D}_{l,q}(a,b) = \begin{cases} \frac{I^{1D}_{l+q}(a+b) - qI^{2D}_{l,q-1}}{b} & q \le \lfloor |b| \rfloor \\ \frac{I^{1D}_{l+q+1}(a+b) - bI^{2D}_{l,q+1}}{q+1} & q > \lfloor |b| \rfloor \end{cases} \text{for } l = 0, K. \tag{C.20}$$

Subsequently, a sweep is started along the l index:

$$I^{2D}_{l,q}(a,b) = \begin{cases} \frac{e^a I^{1D}_q(b) - lI^{2D}_{l-1,q} - I^{1D}_{l+q}(a+b)}{a} & l \le \lfloor |a| \rfloor \\ \frac{e^a I^{1D}_q(b) - aI^{2D}_{l+1,q} - I^{1D}_{l+q+1}(a+b)}{l+1} & l > \lfloor |a| \rfloor \end{cases} \text{for } q = 0, .., K. \tag{C.21}$$

Finally, we need to be able to directly compute $I^{2D}_{l,q}(a,b)$ for the first step of the algorithm. Similar to the one-dimensional integration different combinations of a, b, l and k require different

approaches to achieve the optimal accuracy:

$$I_{l,q}^{2D}(a,b) \leftarrow \begin{cases} a \leq 10,\ b \leq 10 & \text{Taylor series expansion} \\ a > 10,\ b \leq 10 & \text{Partial Taylor series expansion} \\ a > 10,\ b \leq 10 & \text{Reformulation} \\ & \text{then partial Taylor series expansion} \\ a > 10,\ b > 10,\ l \leq 50, q \leq 50 & \text{Direct analytical integration} \\ a > 10,\ b > 10,\ l \text{ or } q > 50 & \text{Numerical integration} \end{cases} \tag{C.22}$$

(i) Taylor Series Expansion

The exponential in both the $\hat{\omega}$ and the $\hat{\theta}$ variables is expanded into a Taylor series and the integration is then performed:

$$I_{l,q}^{2D}(a,b) = \sum_{k_{\hat{\theta}}=0}^{K_{\hat{\theta}}} \underbrace{\frac{b^{k_{\hat{\theta}}}}{k_{\hat{\theta}}!} \sum_{k_{\hat{\omega}}=0}^{K_{\hat{\omega}}} \underbrace{\left[\frac{a^{k_{\hat{\omega}}}}{k_{\hat{\omega}}!} \frac{1}{\left(q+k_{\hat{\theta}}+1\right)\left(q+k_{\hat{\theta}}+l+k_{\hat{\omega}}+1\right)}\right]}_{e_{k_{\hat{\theta}}}}}_{e_{k_{\hat{\theta}}}}, \tag{C.23}$$

where $K_{\hat{\omega}}$ is such that $e_{k_{\hat{\theta}}} < \epsilon$ and $\left|e_{k_{\hat{\theta}}}\right| < \left|e_{k_{\hat{\theta}}-1}\right|$, i.e. the inner sum is first converged before the outer index $k_{\hat{\theta}}$ is incremented. The outer sum is truncated once $e_{k_{\hat{\omega}}} < \epsilon$ and $|e_{k_{\hat{\omega}}}| < |e_{k_{\hat{\omega}}-1}|$.

(ii) Partial Taylor Series Expansion

The exponential in $\hat{\theta}$ is expanded into a Taylor series resulting in:

$$\begin{aligned} I_{l,q}^{2D}(a,b) &= \sum_{k=0}^{K} \frac{b^k}{k!(q+k+1)} \int_0^1 d\hat{\omega} e^{-|a|\hat{\omega}} \hat{\omega}^{l+q+k+1} \\ &= \sum_{k=0}^{K} \underbrace{\frac{b^k}{k!(q+k+1)} I_{l+q+k+1}^{1D}(a)}_{e_k}, \end{aligned} \tag{C.24}$$

where K is chosen such that $e_k < \epsilon$ and $|e_k| < |e_{k-1}|$.

(iii) Reformulation then Partial Taylor Series Expansion

The following identity holds:

$$\begin{aligned} I_{l,q}^{2D}(a,b) &= \int_0^1 d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} \\ &= \int_0^1 d\hat{\theta} \int_{\hat{\theta}}^1 d\hat{\omega}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}}. \end{aligned} \tag{C.25}$$

Further note that by the additivity of the integration operator the following holds:

$$\begin{aligned} & \left[\int_0^1 d\hat{\omega}\ \hat{\omega}^l e^{-|a|\hat{\omega}}\right]\left[\int_0^1 d\hat{\theta}\ \hat{\theta}^q e^{-|b|\hat{\theta}}\right] \\ = & \int_0^1 d\hat{\theta} \int_{\hat{\theta}}^1 d\hat{\omega}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} + \int_0^1 d\hat{\theta} \int_0^{\hat{\theta}} d\hat{\omega}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} \\ = & \int_0^1 d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} + \int_0^1 d\hat{\theta} \int_0^{\hat{\theta}} d\hat{\omega}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} \\ \Rightarrow & \int_0^1 d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} \\ = & \left[\int_0^1 d\hat{\omega}\ \hat{\omega}^l e^{-|a|\hat{\omega}}\right]\left[\int_0^1 d\hat{\theta}\ \hat{\theta}^q e^{-|b|\hat{\theta}}\right] - \int_0^1 d\hat{\theta} \int_0^{\hat{\theta}} d\hat{\omega}\ \hat{\omega}^l \hat{\theta}^q e^{-|a|\hat{\omega}} e^{-|b|\hat{\theta}} \end{aligned} \tag{C.26}$$

Using short-hand notation we can write:

$$I_{l,q}^{2D}(a,b) = I_l^{1D}(a) I_q^{1D}(b) - I_{q,l}^{2D}(b,a)\,. \tag{C.27}$$

Since $a \leq 10$ and $b > 10$ we can now use the method outlined in (ii).

(iv) Direct Analytical Integration

Direct analytical integration results in the following expression:

$$I_{l,q}^{2D}(a,b) = I_l^{1D}(a) q! \frac{1}{|b|^{q+1}} - \sum_{k=0}^{q} \frac{1}{|b|^{k+1}} \frac{q!}{(q-k)} I_{l+q-k}^{1D}(a+b) \tag{C.28}$$

(v) Romberg integration via Duffy transform

The Duffy transform is applied resulting in the following integral:

$$I_{l,q}^{2D}(a,b) = \int_0^1 d\hat{\omega} \int_0^1 d\hat{\theta}\ \hat{\omega}^{l+q+1} \hat{\theta}^q e^{a\hat{\omega}} e^{b\hat{\theta}\cdot\hat{\omega}}. \tag{C.29}$$

Then the two-dimensional equivalent of the Romberg integration algorithm 6 is applied to the integral Eq. C.29

C.4 Integration of 3D Integrals

This section is concerned with evaluating the integral:

$$e^b \iiint\limits_{V_s} dV p_{m_\nu}(\nu)\, p_{m_\omega}(\omega)\, p_{m_\theta}(\theta)\nu^{l_\nu}\omega^{l_\omega}\theta^{l_\theta}e^{a\theta}. \tag{C.30}$$

For convenience we apply a change of variables $\left(\hat{\theta},\hat{\omega},\hat{\nu}\right) \leftarrow (\theta,\omega,\nu)$ onto the unit tetrahedron characterized by the corner points:

$$\hat{r}_1 = (0,0,0)^T,\ \hat{r}_2 = (1,0,0)^T, \hat{r}_3 = (1,1,0)^T, \hat{r}_4 = (1,1,1)^T. \tag{C.31}$$

Let the four corners of the original tetrahedron in the (θ,ω,ν) coordinate system be denoted by $\vec{r}_i$ with $i = 1,2,3,4$, then the transformation from the $\left(\hat{\theta},\hat{\omega},\hat{\nu}\right)$ coordinates to the (θ,ω,ν) coordinates is given by:

$$\begin{aligned} \vec{r} &= \underline{J}\,\hat{r} + \vec{d} \\ \underline{J} &= [\vec{r}_2 - \vec{r}_1, \vec{r}_3 - \vec{r}_2, \vec{r}_4 - \vec{r}_3]\,,\ \vec{d} = \vec{r}_1, \end{aligned} \tag{C.32}$$

where $\underline{J}$ is the transformation Jacobian comprising the elements $j_{i,j}, i,j = 1,2,3$ and $\vec{d} = (d_1,d_2,d_3)$ is the offset of the transformation. Applying the transformation Eq. C.32 to the integral Eq. C.29 the following integral can be obtained:

$$\begin{aligned} & e^b \iiint\limits_{V_s} dV p_{m_\nu}(\nu)\, p_{m_\omega}(\omega)\, p_{m_\theta}(\theta)\nu^{l_\nu}\omega^{l_\omega}\theta^{l_\theta}e^{a\theta} \\ = \ & |\underline{J}|\, e^{b+ad_1} \int_0^1 d\hat{\nu} \int_0^{\hat{\nu}} d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta} \left[\left(j_{3,1}\hat{\theta} + j_{3,2}\hat{\omega} + j_{3,3}\hat{\nu} + d_3 \right)^{l_\nu} \right. \\ \times \ & \left(j_{2,1}\hat{\theta} + j_{2,2}\hat{\omega} + j_{2,3}\hat{\nu} + d_2 \right)^{l_\omega} \\ \times \ & \left(j_{1,1}\hat{\theta} + j_{1,2}\hat{\omega} + j_{1,3}\hat{\nu} + d_1 \right)^{l_\theta} P_{m_\nu}\left(\frac{2}{\Delta\nu_{i_\nu}} \left(j_{3,1}\hat{\theta} + j_{3,2}\hat{\omega} + j_{3,3}\hat{\nu} + d_3 - \nu_{i_\nu} + \nu_{i_\nu - 1} \right) \right) \\ \times \ & e^{-|aj_{1,3}|\hat{\nu}} P_{m_\omega}\left(\frac{2}{\Delta\omega_{i_\omega}} \left(j_{2,1}\hat{\theta} + j_{2,2}\hat{\omega} + j_{2,3}\hat{\nu} + d_2 - \omega_{i_\omega} + \omega_{i_\omega - 1} \right) \right) e^{-|aj_{1,2}|\hat{\omega}} \\ \times \ & \left. P_{m_\theta}\left(\frac{2}{\Delta\theta_{i_\theta}} \left(j_{1,1}\hat{\theta} + j_{1,2}\hat{\omega} + j_{1,3}\hat{\nu} + d_1 - \theta_{i_\theta} + \theta_{i_\theta - 1} \right) \right) e^{-|aj_{1,1}|\hat{\theta}} \right] \end{aligned} \tag{C.33}$$

Note that the arguments within the exponentials are non-negative. This can be achieved by finding the suitable permutation of associating a corner point of the original tetrahedron with the vectors $\vec{r}_i,\ i = 1,..,4$. It is guaranteed that out of the $4! = 24$ permutations one will yield a transformation such that $(aj_{1,i}) < 0,\ i = 1,..,4$. Using the algorithms in section C.1 the

integral C.33 can be reduced to a sum of elementary integrals:

$$\begin{aligned} & e^{\hat{b}} \iiint\limits_{V_s} dV p_{m_\nu}(\nu)\, p_{m_\omega}(\omega)\, p_{m_\theta}(\theta) \nu^{l_\nu} \omega^{l_\omega} \theta^{l_\theta} e^{a\theta} \\ = \quad & |\underline{J}|\, e^{\hat{b}} \sum_{k_\nu=0}^{K} \sum_{k_\omega=0}^{K} \sum_{k_\theta=0}^{K} \hat{c}_{k_\nu,k_\omega,k_\theta} I^{3D}_{k_\nu,k_\omega,k_\theta}\left(\hat{a}_{\hat{\nu}}, \hat{a}_{\hat{\omega}}, \hat{a}_{\hat{\theta}}\right), \end{aligned} \tag{C.34}$$

where the following definitions are used:

- $K = l_\theta + l_\omega + l_\nu + m_\theta + m_\omega + m_\nu$
- $\hat{c}_{k_\nu,k_\omega,k_\theta}$: Polynomial coefficients from algorithms in section C.1.
- $\hat{b} = b + ad_1$
- $\hat{a}_{\hat{\theta}} = aj_{1,3}$
- $\hat{a}_{\hat{\omega}} = aj_{1,2}$
- $\hat{a}_{\hat{\nu}} = aj_{1,1}$
- $I^{3D}_{k_\nu,k_\omega,k_\theta}\left(\hat{a}_{\hat{\nu}}, \hat{a}_{\hat{\omega}}, \hat{a}_{\hat{\theta}}\right) = \int_0^1 d\hat{\nu} \int_0^{\hat{\nu}} d\hat{\omega} \int_0^{\hat{\omega}} d\hat{\theta} \hat{\nu}^{k_\nu} \hat{\omega}^{k_\omega} \hat{\theta}^{k_\theta} e^{-|\hat{a}_{\hat{\nu}}|\hat{\nu}} e^{-|\hat{a}_{\hat{\omega}}|\hat{\omega}} e^{-|\hat{a}_{\hat{\theta}}|\hat{\theta}}$

Thus the problem can be reduced to evaluating the integral $I^{3D}_{l,q,t}(a,b,c)$ for $l,q,t = 0,...,K$. To this end an unconditionally stable forward/backward recursion is devised which utilizes the following steps:

- Evaluate $I^{3D}_{l,q,t}(a,b,c)$ for $(l,q,t) = (0/K, 0/K, 0/K)$ using an algorithm that is going to be outlined later.
- Perform a forward/backward sweep along t using the following prescription:

$$I^{3D}_{l,q,t} = \begin{cases} \frac{I^{2D}_{l,q+t}(a,b+c) - tI^{3D}_{l,q,t-1}}{c} & t \le \lfloor |c| \rfloor \\ \frac{I^{2D}_{l,q+t+1}(a,b+c) - cI^{3D}_{l,q,t+1}}{t+1} & t > \lfloor |c| \rfloor \end{cases} \text{for } l,q = 0,K. \tag{C.35}$$

- Perform a forward/backward sweep along q using the following prescription:

$$I^{3D}_{l,q,t} = \begin{cases} \frac{I^{2D}_{l+q,t}(a+b,c) - I^{2D}_{l,q+t}(a,b+c) - qI^{3D}_{l,q-1,t}}{b} & q \le \lfloor |b| \rfloor \\ \frac{I^{2D}_{l+q+1,t}(a+b,c) - I^{2D}_{l,q+t+1}(a,b+c) - bI^{3D}_{l,q-1,t}}{q+1} & q > \lfloor |b| \rfloor \end{cases} \text{for } l = 0,K\ , t = 0,...,K. \tag{C.36}$$

- Perform a forward/backward sweep along l using the following prescription:

$$I^{3D}_{l,q,t} = \begin{cases} \frac{e^a I^{2D}_{q,t}(b,c) - I^{2D}_{l+q,t}(a+b,c) - l I^{3D}_{l-1,q,t}}{a} & l \leq \lfloor |a| \rfloor \\ \frac{e^a I^{2D}_{q,t}(b,c) - I^{2D}_{l+q+1,t}(a+b,c) - a I^{3D}_{l+1,q,t}}{l+1} & l > \lfloor |a| \rfloor \end{cases} \text{ for } q,t = 0,...,K. \tag{C.37}$$

Finally for the first step in the forward/backward recursion algorithm the direct evaluation of $I^{3D}_{l,q,t}$ needs to be discussed. As for the one and two-dimensional integrals it turns out to be beneficial for the accuracy to utilize different approaches depending on the values of l, q, t, a, b and c:

$$I^{3D}_{l,q,t}(a,b,c) \leftarrow \begin{cases} a,b,c \leq 10 & \text{(i) Taylor series expansion} \\ a > 10,\ c,c \leq 10 & \text{(ii) Partial Taylor series expansion} \\ b > 10,\ c \leq 10 & \text{(iii) Partial Taylor series expansion} \\ c > 10 & \text{(iv) Analytical Evaluation} \end{cases} \tag{C.38}$$

(i) Taylor series expansion

The exponentials in $\hat{\nu}$, $\hat{\omega}$ and $\hat{\theta}$ are expanded into a Taylor series and then the integration is performed analytically:

$$I^{3D}_{l,q,t}(a,b,c) = \sum_{i_\theta=0}^{K_\theta} e_{k_\theta},\ K_\theta \text{ s.t. } e_{K_\theta} \leq \epsilon \text{ and } |e_{K_\theta}| - |e_{K_\theta - 1}| < 0$$

$$e_{k_\theta} = \underbrace{\left[\sum_{i_\omega=0}^{K_\omega} \underbrace{\left[\sum_{i_\nu=0}^{K_\nu} \underbrace{\frac{c^{k_\theta}}{k_\theta!} \frac{b^{k_\omega}}{k_\omega!} \frac{a^{k_\nu}}{k_\nu!} \frac{1}{(k_\theta + t + 1)(k_\theta + t + k_\omega + q + 2)(k_\theta + t + k_\omega + q + k_\nu + l)}}_{e_{k_\nu}} \right]}_{e_{k_\omega}} \right]}$$

$$K_\omega \text{ s.t. } e_{K_\omega} \leq \epsilon \text{ and } |e_{K_\omega}| - |e_{K_\omega - 1}| < 0$$

$$K_\nu \text{ s.t. } e_{K_\nu} \leq \epsilon \text{ and } |e_{K_\nu}| - |e_{K_\nu - 1}| < 0. \tag{C.39}$$

Note, that the inner loops are converged before the loop index of the outer loop is increased.

(ii) Partial Taylor series expansion

The exponentials in $\hat{\omega}$ and $\hat{\theta}$ are expanded into a Taylor series and then the integration is

performed analytically:

$$I^{3D}_{l,q,t}(a,b,c) = \sum_{k_\theta=0}^{K_\theta} \underbrace{\sum_{k_\omega=0}^{K_\omega} \underbrace{\frac{b^{k_\omega}}{k_\omega!}\frac{c^{k_\theta}}{k_\theta!}\frac{1}{(1+k_\theta+t)(2+k_\theta+k_\omega+q+t)} I^{1D}_{k_\omega+k_\theta+l+q+t+2}(a)}_{e_\omega}}_{e_\theta}$$
$$K_\omega \text{ s.t. } e_{K_\omega} \leq \epsilon \text{ and } |e_{K_\omega}| - |e_{K_\omega-1}| < 0$$
$$K_\theta \text{ s.t. } e_{K_\theta} \leq \epsilon \text{ and } |e_{K_\theta}| - |e_{K_\theta-1}| < 0. \quad \text{(C.40)}$$

(iii) Partial Taylor series expansion

The exponential in the $\hat{\theta}$ variable is expanded into a Taylor Series and the integration is then performed analytically:

$$I^{3D}_{l,q,t}(a,b,c) = \sum_{k_\theta=0}^{K_\theta} \underbrace{\frac{c^{k_\theta}}{k_\theta!}\frac{1}{(k_\theta+t+1)} I^{2D}_{l,q+t+k_\theta+1}(a,b)}_{e_{k_\theta}}$$
$$K_\theta \text{ s.t. } e_{K_\theta} \leq \epsilon \text{ and } |e_{K_\theta}| - |e_{K_\theta-1}| < 0. \quad \text{(C.41)}$$

(iv) Analytical evaluation

The inner integral over $\hat{\theta}$ is performed analytically and evaluated at the upper and lower integration limit:

$$I^{3D}_{l,q,t}(a,b,c) = \frac{1}{|c|^{t+1}} t! I^{2D}_{l,q}(a,b) - \sum_{r=0}^{t} \frac{1}{|c|^{r+1}} \frac{t!}{(t-r)!} I^{2D}_{l,q+t-r}(a,b+c). \quad \text{(C.42)}$$

Appendix D

Method Implementation Details

Several Mathematica notebooks of hard-coded low order spatial discretization methods.

D.1 Linear Discontinuous Method

LD equations

Set up the 4 equations for the LD method. These four equations contain face and volume unknowns. The face unknowns will be removed using the upstreaming relations.

```
In[1]:= Eq1 = Δx Δy Δz ( μ/Δx (ψE - ψW) + η/Δy (ψN - ψS) + ξ/Δz (ψT - ψB) + σ ψa ) == Δx Δy Δz ( Sa );

In[2]:= Eq2 = Δx Δy Δz ( μ/Δx (ψz - ψWz) + η/Δy (ψz - ψSz) + 3 ξ/Δz (ψT + ψB - 2 ψa) + σ ψz ) ==
          Δx Δy Δz ( Sz );

In[3]:= Eq3 = Δx Δy Δz ( μ/Δx (ψy - ψWy) + 3 η/Δy (ψN + ψS - 2 ψa) + ξ/Δz (ψy - ψBy ) + σ ψy ) ==
          Δx Δy Δz ( Sy );

In[4]:= Eq4 = Δx Δy Δz ( 3 μ/Δx (ψE + ψW - 2 ψa ) + η/Δy (ψx - ψSx ) + ξ/Δz (ψx - ψBx ) + σ ψx ) ==
          Δx Δy Δz (Sx) ;
```

Replace the face fluxes by volumetric flux moments using the upstream relations (backwards).

```
In[5]:= repl = {ψE → ψa + ψx, ψN → ψa + ψy, ψT → ψa + ψz};

In[6]:= Eq1 = Eq1 /. repl; Eq2 = Eq2 /. repl; Eq3 = Eq3 /. repl; Eq4 = Eq4 /. repl;
```

Assemble equations into matrix form by using the CoefficientArrays function.

```
In[7]:= arr = CoefficientArrays[{Eq1, Eq2, Eq3, Eq4}, {ψa, ψz, ψy, ψx}];
```

Display the resulting matrix in a matrix format. In order to fit on a single line the matrix is divided by the volume. Keep in mind that the matrix that is actually used is V*T.

```
In[8]:= T = Normal[Simplify[arr[[2]] / (Δx Δy Δz)]]; T // MatrixForm
```

Out[8]//MatrixForm=

$$\begin{pmatrix} \frac{\eta}{\Delta y}+\frac{\mu}{\Delta x}+\frac{\xi}{\Delta z}+\sigma & \frac{\xi}{\Delta z} & \frac{\eta}{\Delta y} & \frac{\mu}{\Delta x} \\ -\frac{3\,\xi}{\Delta z} & \frac{\eta}{\Delta y}+\frac{\mu}{\Delta x}+\frac{3\,\xi}{\Delta z}+\sigma & 0 & 0 \\ -\frac{3\,\eta}{\Delta y} & 0 & \frac{3\,\eta}{\Delta y}+\frac{\mu}{\Delta x}+\frac{\xi}{\Delta z}+\sigma & 0 \\ -\frac{3\,\mu}{\Delta x} & 0 & 0 & \frac{\eta}{\Delta y}+\frac{3\,\mu}{\Delta x}+\frac{\xi}{\Delta z}+\sigma \end{pmatrix}$$

Display the resulting right hand side in a matrix format. Same here regarding division by the volume.

In[11]:=
```
b = -Normal[Simplify[arr[[1]] / (Δx Δy Δz)]]; b // MatrixForm
```

Out[11]//MatrixForm=

$$\begin{pmatrix} Sa + \frac{\xi\ \psi B}{\Delta z} + \frac{\eta\ \psi S}{\Delta y} + \frac{\mu\ \psi W}{\Delta x} \\ Sz - \frac{3\ \xi\ \psi B}{\Delta z} + \frac{\eta\ \psi Sz}{\Delta y} + \frac{\mu\ \psi Wz}{\Delta x} \\ Sy + \frac{\xi\ \psi By}{\Delta z} - \frac{3\ \eta\ \psi S}{\Delta y} + \frac{\mu\ \psi Wy}{\Delta x} \\ Sx + \frac{\xi\ \psi Bx}{\Delta z} + \frac{\eta\ \psi Sx}{\Delta y} - \frac{3\ \mu\ \psi W}{\Delta x} \end{pmatrix}$$

Direct Inversion of linear system of equations.

Instead of using the explicit matrix derived in before, we assign a placeholder for each non-zero matrix entry and solve the 4x4 linear system
of equations in terms of the placeholders. If we used the expressions derived before the resulting solution would be too complicated.

In[19]:=
```
T = {{a11, a12, a13, a14}, {-3 a12, a22, 0, 0}, {-3 a13, 0, a33, 0}, {-3 a14, 0, 0, a44}}; b = {b1, b2, b3, b4};
```

Show the result in matrix form.

In[20]:=
```
LinearSolve[T, b] // MatrixForm
```

Out[20]//MatrixForm=

$$\begin{pmatrix} \frac{a22\ a33\ a44\ b1 - a12\ a33\ a44\ b2 - a13\ a22\ a44\ b3 - a14\ a22\ a33\ b4}{3\ a14^2\ a22\ a33 + 3\ a13^2\ a22\ a44 + 3\ a12^2\ a33\ a44 + a11\ a22\ a33\ a44} \\ \frac{3\ a12\ a33\ a44\ b1 + 3\ a14^2\ a33\ b2 + 3\ a13^2\ a44\ b2 + a11\ a33\ a44\ b2 - 3\ a12\ a13\ a44\ b3 - 3\ a12\ a14\ a33\ b4}{3\ a14^2\ a22\ a33 + 3\ a13^2\ a22\ a44 + 3\ a12^2\ a33\ a44 + a11\ a22\ a33\ a44} \\ \frac{3\ a13\ a22\ a44\ b1 - 3\ a12\ a13\ a44\ b2 + 3\ a14^2\ a22\ b3 + 3\ a12^2\ a44\ b3 + a11\ a22\ a44\ b3 - 3\ a13\ a14\ a22\ b4}{3\ a14^2\ a22\ a33 + 3\ a13^2\ a22\ a44 + 3\ a12^2\ a33\ a44 + a11\ a22\ a33\ a44} \\ \frac{3\ a14\ a22\ a33\ b1 - 3\ a12\ a14\ a33\ b2 - 3\ a13\ a14\ a22\ b3 + 3\ a13^2\ a22\ b4 + 3\ a12^2\ a33\ b4 + a11\ a22\ a33\ b4}{3\ a14^2\ a22\ a33 + 3\ a13^2\ a22\ a44 + 3\ a12^2\ a33\ a44 + a11\ a22\ a33\ a44} \end{pmatrix}$$

D.2 Linear Nodal Method

Formulating and Solving the linear system of equations for the LN equations

Setting up the nodal unknown vector

$$\psi v = \{\psi_{0,0,0}, \psi_{0,0,1}, \psi_{0,1,0}, \psi_{1,0,0}\};$$

Balance Relations

These balance relations are divided by σ and written in dimenionless form by using the optical thicknesses.

$$b1 = \frac{\psi ox_{0,0} - \psi ix_{0,0}}{tx} + \frac{\psi oy_{0,0} - \psi iy_{0,0}}{ty} + \frac{\psi oz_{0,0} - \psi iz_{0,0}}{tz} + \psi_{0,0,0} == \frac{S_{0,0,0}}{\sigma};$$

$$b2 = \frac{\psi ox_{0,1} - \psi ix_{0,1}}{tx} + \frac{\psi oy_{0,1} - \psi iy_{0,1}}{ty} + sz\ \frac{\psi oz_{0,0} + \psi iz_{0,0} - 2\,\psi_{0,0,0}}{tz} + \psi_{0,0,1} == \frac{S_{0,0,1}}{\sigma};$$

$$b3 = \frac{\psi ox_{1,0} - \psi ix_{1,0}}{tx} + sy\ \frac{\psi oy_{0,0} + \psi iy_{0,0} - 2\,\psi_{0,0,0}}{ty} + \frac{\psi oz_{0,1} - \psi iz_{0,1}}{tz} + \psi_{0,1,0} == \frac{S_{0,1,0}}{\sigma};$$

$$b4 = sx\ \frac{\psi ox_{0,0} + \psi ix_{0,0} - 2\,\psi_{0,0,0}}{tx} + \frac{\psi oy_{1,0} - \psi iy_{1,0}}{ty} + \frac{\psi oz_{1,0} - \psi iz_{1,0}}{tz} + \psi_{1,0,0} == \frac{S_{1,0,0}}{\sigma};$$

```
balance = {b1, b2, b3, b4};
```

WDDx

Auxiliary equations/WDD equations in x direction. sx is the sign of the direction cosine w.r.t. x direction.

$$wddx1 = \frac{1+\alpha_0}{2}\ \psi ox_{0,0} + \frac{1-\alpha_0}{2}\ \psi ix_{0,0} == \psi_{0,0,0} + 3\,sx\ \alpha_0\ \psi_{1,0,0};$$

$$wddx2 = \frac{1+\alpha_1}{2}\ \psi ox_{1,0} + \frac{1-\alpha_1}{2}\ \psi ix_{1,0} == \psi_{0,1,0};$$

$$wddx3 = \frac{1+\alpha_1}{2}\ \psi ox_{0,1} + \frac{1-\alpha_1}{2}\ \psi ix_{0,1} == \psi_{0,0,1};$$

```
wddx = {wddx1, wddx2, wddx3};
```

Solve the 3 WDD relations for the three outflow unkowns on the +x face.

```
replx = First[Solve[wddx, {ψox0,0, ψox1,0, ψox0,1}]];
```

WDDy

Auxiliary equations/WDD equations in y direction. sy is the sign of the direction cosine w.r.t. y direction.

```
wddy1 = (1 + β0)/2 ψoy0,0 + (1 - β0)/2 ψiy0,0 == ψ0,0,0 + 3 sy β0 ψ0,1,0;
wddy2 = (1 + β1)/2 ψoy1,0 + (1 - β1)/2 ψiy1,0 == ψ1,0,0;
wddy3 = (1 + β1)/2 ψoy0,1 + (1 - β1)/2 ψiy0,1 == ψ0,0,1;
wddy = {wddy1, wddy2, wddy3};
```

Solve the 3 WDD relations for the three outflow unkowns on the +y face.

```
reply = First[Solve[wddy, {ψoy0,0, ψoy1,0, ψoy0,1}]];
```

WDDz

Auxiliary equations/WDD equations in z direction. sz is the sign of the direction cosine w.r.t. z direction.

```
wddz1 = (1 + γ0)/2 ψoz0,0 + (1 - γ0)/2 ψiz0,0 == ψ0,0,0 + 3 sz γ0 ψ0,0,1;
wddz2 = (1 + γ1)/2 ψoz1,0 + (1 - γ1)/2 ψiz1,0 == ψ1,0,0;
wddz3 = (1 + γ1)/2 ψoz0,1 + (1 - γ1)/2 ψiz0,1 == ψ0,1,0;
wddz = {wddz1, wddz2, wddz3};
```

Solve the 3 WDD relations for the three outflow unkowns on the +z face.

```
replz = First[Solve[wddz, {ψoz0,0, ψoz1,0, ψoz0,1}]];
```

Replace outflow moments from balance relations

Put the expressions for the outflow flux moments into one vector.

```
repl = Join[replx, reply, replz];
```

Substute this vector into the balance equations.

```
balance = balance /. repl;
```

Construct the local linear system of equations and show it!

```
mat = CoefficientArrays[balance, ψv];
```

```
Normal[mat[[2]]] // MatrixForm
```

$$\begin{pmatrix} 1+\frac{2}{tx\,(1+\alpha_0)}+\frac{2}{ty\,(1+\beta_0)}+\frac{2}{tz\,(1+\gamma_0)} & \frac{6\,sz\,\gamma_0}{tz\,(1+\gamma_0)} & \frac{6\,sy\,\beta_0}{ty\,(1+\beta_0)} \\ -\frac{2\,sz}{tz}+\frac{2\,sz}{tz\,(1+\gamma_0)} & 1+\frac{2}{tx\,(1+\alpha_1)}+\frac{2}{ty\,(1+\beta_1)}+\frac{6\,sz^2\,\gamma_0}{tz\,(1+\gamma_0)} & 0 \\ -\frac{2\,sy}{ty}+\frac{2\,sy}{ty\,(1+\beta_0)} & 0 & 1+\frac{2}{tx\,(1+\alpha_1)}+\frac{6\,sy^2\,\beta_0}{ty\,(1+\beta_0)}+\frac{2}{tz\,(1+\gamma} \\ -\frac{2\,sx}{tx}+\frac{2\,sx}{tx\,(1+\alpha_0)} & 0 & 0 \end{pmatrix}$$

```
rhs = -Normal[mat[[1]]]; rhs // MatrixForm
```

$$\begin{pmatrix} \frac{\psi ix_{0,0}}{tx}+\frac{\psi ix_{0,0}}{tx\,(1+\alpha_0)}-\frac{\alpha_0\,\psi ix_{0,0}}{tx\,(1+\alpha_0)}+\frac{\psi iy_{0,0}}{ty}+\frac{\psi iy_{0,0}}{ty\,(1+\beta_0)}-\frac{\beta_0\,\psi iy_{0,0}}{ty\,(1+\beta_0)}+\frac{\psi iz_{0,0}}{tz}+\frac{\psi iz_{0,0}}{tz\,(1+\gamma_0)}-\frac{\gamma_0\,\psi iz_{0,0}}{tz\,(1+\gamma_0)}+\frac{S_{0,0,0}}{\sigma} \\ \frac{\psi ix_{0,1}}{tx}+\frac{\psi ix_{0,1}}{tx\,(1+\alpha_1)}-\frac{\alpha_1\,\psi ix_{0,1}}{tx\,(1+\alpha_1)}+\frac{\psi iy_{0,1}}{ty}+\frac{\psi iy_{0,1}}{ty\,(1+\beta_1)}-\frac{\beta_1\,\psi iy_{0,1}}{ty\,(1+\beta_1)}-\frac{sz\,\psi iz_{0,0}}{tz}+\frac{sz\,\psi iz_{0,0}}{tz\,(1+\gamma_0)}-\frac{sz\,\gamma_0\,\psi iz_{0,0}}{tz\,(1+\gamma_0)}+\frac{S_{0,0,1}}{\sigma} \\ \frac{\psi ix_{1,0}}{tx}+\frac{\psi ix_{1,0}}{tx\,(1+\alpha_1)}-\frac{\alpha_1\,\psi ix_{1,0}}{tx\,(1+\alpha_1)}-\frac{sy\,\psi iy_{0,0}}{ty}+\frac{sy\,\psi iy_{0,0}}{ty\,(1+\beta_0)}-\frac{sy\,\beta_0\,\psi iy_{0,0}}{ty\,(1+\beta_0)}+\frac{\psi iz_{0,1}}{tz}+\frac{\psi iz_{0,1}}{tz\,(1+\gamma_1)}-\frac{\gamma_1\,\psi iz_{0,1}}{tz\,(1+\gamma_1)}+\frac{S_{0,1,0}}{\sigma} \\ -\frac{sx\,\psi ix_{0,0}}{tx}+\frac{sx\,\psi ix_{0,0}}{tx\,(1+\alpha_0)}-\frac{sx\,\alpha_0\,\psi ix_{0,0}}{tx\,(1+\alpha_0)}+\frac{\psi iy_{1,0}}{ty}+\frac{\psi iy_{1,0}}{ty\,(1+\beta_1)}-\frac{\beta_1\,\psi iy_{1,0}}{ty\,(1+\beta_1)}+\frac{\psi iz_{1,0}}{tz}+\frac{\psi iz_{1,0}}{tz\,(1+\gamma_1)}-\frac{\gamma_1\,\psi iz_{1,0}}{tz\,(1+\gamma_1)}+\frac{S_{1,0,0}}{\sigma} \end{pmatrix}$$

Solve 4 x4 system of equations

```
A = {{a11, a12, a13, a14}, {a21, a22, 0, 0}, {a31, 0, a33, 0}, {a41, 0, 0, a44}}; rv = {rhs1, rhs2, rhs3, rhs4};
```

```
xv = LinearSolve[A, rv];
```

```
Simplify[xv[[1]]]
```

```
(a12 a33 a44 rhs2 + a22 (-a33 a44 rhs1 + a13 a44 rhs3 + a14 a33 rhs4)) /
  (a14 a22 a33 a41 + (a13 a22 a31 + a12 a21 a33 - a11 a22 a33) a44)
```

```
Simplify[xv[[2]]]
```

```
(a21 a33 a44 rhs1 + a14 a33 a41 rhs2 +
    a13 a31 a44 rhs2 - a11 a33 a44 rhs2 - a13 a21 a44 rhs3 - a14 a21 a33 rhs4) /
  (a14 a22 a33 a41 + a13 a22 a31 a44 + a12 a21 a33 a44 - a11 a22 a33 a44)
```

```
Simplify[xv[[3]]]
```

```
(a12 a44 (-a31 rhs2 + a21 rhs3) +
    a22 (a31 a44 rhs1 + a14 a41 rhs3 - a11 a44 rhs3 - a14 a31 rhs4)) /
  (a14 a22 a33 a41 + (a13 a22 a31 + a12 a21 a33 - a11 a22 a33) a44)
```

```
Simplify[xv[[4]]]
```

```
(a12 a33 (-a41 rhs2 + a21 rhs4) +
    a22 (a33 a41 rhs1 - a13 a41 rhs3 + a13 a31 rhs4 - a11 a33 rhs4)) /
  (a14 a22 a33 a41 + (a13 a22 a31 + a12 a21 a33 - a11 a22 a33) a44)
```

Compute outflow face fluxes

In this last step the upstreaming expressions are derived. It basically uses the expressions obtained when the WDD equations are
solved for the outflow face moments:

$$\psi ox_{0,0} = \psi ox_{0,0} \text{ /. replx}$$

$$\frac{-\psi ix_{0,0} + \alpha_0\ \psi ix_{0,0} + 2\ \psi_{0,0,0} + 6\ sx\ \alpha_0\ \psi_{1,0,0}}{1 + \alpha_0}$$

$$\psi ox_{0,1} = \psi ox_{0,1} \text{ /. replx}$$

$$\frac{-\psi ix_{0,1} + \alpha_1\ \psi ix_{0,1} + 2\ \psi_{0,0,1}}{1 + \alpha_1}$$

$$\psi ox_{1,0} = \psi ox_{1,0} \text{ /. replx}$$

$$\frac{-\psi ix_{1,0} + \alpha_1\ \psi ix_{1,0} + 2\ \psi_{0,1,0}}{1 + \alpha_1}$$

$$\psi oy_{0,0} = \psi oy_{0,0} \text{ /. reply}$$

$$\frac{-\psi iy_{0,0} + \beta_0\ \psi iy_{0,0} + 2\ \psi_{0,0,0} + 6\ sy\ \beta_0\ \psi_{0,1,0}}{1 + \beta_0}$$

$$\psi oy_{0,1} = \psi oy_{0,1} \text{ /. reply}$$

$$\frac{-\psi iy_{0,1} + \beta_1\ \psi iy_{0,1} + 2\ \psi_{0,0,1}}{1 + \beta_1}$$

$$\psi oy_{1,0} = \psi oy_{1,0} \text{ /. reply}$$

$$\frac{-\psi iy_{1,0} + \beta_1\ \psi iy_{1,0} + 2\ \psi_{1,0,0}}{1 + \beta_1}$$

$$\psi oz_{0,0} = \psi oz_{0,0} \text{ /. replz}$$

$$\frac{-\psi iz_{0,0} + \gamma_0\ \psi iz_{0,0} + 2\ \psi_{0,0,0} + 6\ sz\ \gamma_0\ \psi_{0,0,1}}{1 + \gamma_0}$$

$$\psi oz_{0,1} = \psi oz_{0,1} \text{ /. replz}$$

$$\frac{-\psi iz_{0,1} + \gamma_1\ \psi iz_{0,1} + 2\ \psi_{0,1,0}}{1 + \gamma_1}$$

$$\psi oz_{1,0} = \psi oz_{1,0} \text{ /. replz}$$

$$\frac{-\psi iz_{1,0} + \gamma_1\ \psi iz_{1,0} + 2\ \psi_{1,0,0}}{1 + \gamma_1}$$

D.3 Linear-Linear Method

Formulating and Solving the linear system of equations for the LL equations

Setting up the nodal unknown vector

In[1]:=
$$\psi v = \{\psi_{0,0,0},\ \psi_{0,0,1},\ \psi_{0,1,0},\ \psi_{1,0,0}\};$$

Balance Relations

These balance relations are divided by σ and written in dimenionless form by using the optical thicknesses.

In[2]:=
$$b1 = \frac{\psi ox_{0,0} - \psi ix_{0,0}}{tx} + \frac{\psi oy_{0,0} - \psi iy_{0,0}}{ty} + \frac{\psi oz_{0,0} - \psi iz_{0,0}}{tz} + \psi_{0,0,0} == \frac{S_{0,0,0}}{\sigma};$$

In[3]:=
$$b2 = \frac{\psi ox_{0,1} - \psi ix_{0,1}}{tx} + \frac{\psi oy_{0,1} - \psi iy_{0,1}}{ty} + sz\ \frac{\psi oz_{0,0} + \psi iz_{0,0} - 2\,\psi_{0,0,0}}{tz} + \psi_{0,0,1} == \frac{S_{0,0,1}}{\sigma};$$

In[4]:=
$$b3 = \frac{\psi ox_{1,0} - \psi ix_{1,0}}{tx} + sy\ \frac{\psi oy_{0,0} + \psi iy_{0,0} - 2\,\psi_{0,0,0}}{ty} + \frac{\psi oz_{0,1} - \psi iz_{0,1}}{tz} + \psi_{0,1,0} == \frac{S_{0,1,0}}{\sigma};$$

In[5]:=
$$b4 = sx\ \frac{\psi ox_{0,0} + \psi ix_{0,0} - 2\,\psi_{0,0,0}}{tx} + \frac{\psi oy_{1,0} - \psi iy_{1,0}}{ty} + \frac{\psi oz_{1,0} - \psi iz_{1,0}}{tz} + \psi_{1,0,0} == \frac{S_{1,0,0}}{\sigma};$$

In[6]:=
```
balance = {b1, b2, b3, b4};
```

WDDx

Auxiliary equations/WDD equations in x direction. sx is the sign of the direction cosine w.r.t. x direction.

In[7]:=
$$wddx1 = \frac{1+\alpha_0}{2}\ \psi ox_{0,0} + \frac{1-\alpha_0}{2}\ \psi ix_{0,0} == \psi_{0,0,0} + 3\,sx\ \alpha_0\ \psi_{1,0,0};$$

In[8]:=
$$wddx2 = \frac{1+\alpha_1}{2}\ \psi ox_{1,0} + \frac{1-\alpha_1}{2}\ \psi ix_{1,0} == \psi_{0,1,0} - \frac{3\,sx\ \alpha_1}{sy\ ty}\left(\psi oy_{1,0} + \psi iy_{1,0} - 2\,\psi_{1,0,0}\right);$$

In[9]:=
$$wddx3 = \frac{1+\alpha_1}{2}\ \psi ox_{0,1} + \frac{1-\alpha_1}{2}\ \psi ix_{0,1} == \psi_{0,0,1} - \frac{3\,sx\ \alpha_1}{sz\ tz}\left(\psi oz_{1,0} + \psi iz_{1,0} - 2\,\psi_{1,0,0}\right);$$

In[10]:=
```
wddx = {wddx1, wddx2, wddx3};
```

WDDy

Auxiliary equations/WDD equations in z direction. sy is the sign of the direction cosine

w.r.t. y direction.

```
In[11]:= wddy1 = (1 + β0)/2 ψoy0,0 + (1 - β0)/2 ψiy0,0 == ψ0,0,0 + 3 sy β0 ψ0,1,0;

In[12]:= wddy2 = (1 + β1)/2 ψoy1,0 + (1 - β1)/2 ψiy1,0 == ψ1,0,0 - (3 sy β1)/(sx tx) (ψox1,0 + ψix1,0 - 2 ψ0,1,0);

In[13]:= wddy3 = (1 + β1)/2 ψoy0,1 + (1 - β1)/2 ψiy0,1 == ψ0,0,1 - (3 sy β1)/(sz tz) (ψoz0,1 + ψiz0,1 - 2 ψ0,1,0);

In[14]:= wddy = {wddy1, wddy2, wddy3};
```

WDDz

Auxiliary equations/WDD equations in z direction. sz is the sign of the direction cosine w.r.t. z direction.

```
In[15]:= wddz1 = (1 + γ0)/2 ψoz0,0 + (1 - γ0)/2 ψiz0,0 == ψ0,0,0 + 3 sz γ0 ψ0,0,1;

In[16]:= wddz2 = (1 + γ1)/2 ψoz1,0 + (1 - γ1)/2 ψiz1,0 == ψ1,0,0 - (3 sz γ1)/(sx tx) (ψox0,1 + ψix0,1 - 2 ψ0,0,1);

In[17]:= wddz3 = (1 + γ1)/2 ψoz0,1 + (1 - γ1)/2 ψiz0,1 == ψ0,1,0 - (3 sz γ1)/(sy ty) (ψoy0,1 + ψiy0,1 - 2 ψ0,0,1);

In[18]:= wddz = {wddz1, wddz2, wddz3};
```

Solve WDD relations for outflow moments

Collect all the 0-0 wDD equations into wdd1. The 0 - 0 equations can be solved independently.

```
In[19]:= wdd1 = {wddx1, wddy1, wddz1};

In[20]:= repl1 = First[Solve[wdd1, {ψox0,0, ψoy0,0, ψoz0,0}]];
```

Solve the six remaining WDD equations.

```
In[21]:= wdd2 = {wddx2, wddx3, wddy2, wddy3, wddz2, wddz3};

In[22]:= repl2 = First[Solve[wdd2, {ψox0,1, ψox1,0, ψoy0,1, ψoy1,0, ψoz0,1, ψoz1,0}]];
```

Replace outflow moments from balance relations

```
In[23]:= repl = Join[repl1, repl2];

In[24]:= balance = balance /. repl;
```

In[25]:=
```
balance // MatrixForm;
```

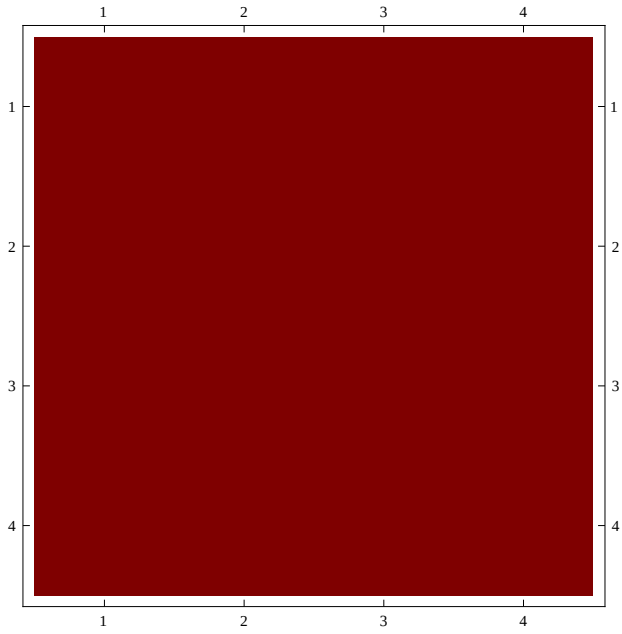
In[26]:=
```
mat = Simplify[CoefficientArrays[balance, ψv]];
```

In[27]:=
```
TT = Normal[mat[[2]]]; (*TT//MatrixForm*)
```

Show pattern in Matrix Plot:

In[28]:=
```
MatrixPlot[TT, ColorFunction → Monochrome]
```

Out[28]=



The LL matrix is full.

In[29]:=
```
rhs = -Normal[mat[[1]]]; (*rhs//MatrixForm *)
```

Solve 4 x4 system of equations

For LL the local matrix is full!

In[30]:=
$$A = \begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{pmatrix}; \quad rv = \{rhs1, rhs2, rhs3, rhs4\};$$

In[31]:=
```
xv = LinearSolve[A, rv];
```

In[32]:= **Simplify[xv[[1]]]**

Out[32]=
```
(-a22 a34 a43 rhs1 + a22 a33 a44 rhs1 + a14 a33 a42 rhs2 - a13 a34 a42 rhs2 - a14 a32 a43 rhs2 +
    a12 a34 a43 rhs2 + a13 a32 a44 rhs2 - a12 a33 a44 rhs2 + a14 a22 a43 rhs3 -
    a13 a22 a44 rhs3 - a14 a22 a33 rhs4 + a13 a22 a34 rhs4 + a24 (-a33 a42 rhs1 +
        a32 a43 rhs1 + a13 a42 rhs3 - a12 a43 rhs3 - a13 a32 rhs4 + a12 a33 rhs4) + a23
     (a34 a42 rhs1 - a32 a44 rhs1 - a14 a42 rhs3 + a12 a44 rhs3 + a14 a32 rhs4 - a12 a34 rhs4)) /
  (a12 a24 a33 a41 - a12 a23 a34 a41 - a11 a24 a33 a42 + a11 a23 a34 a42 -
    a12 a24 a31 a43 + a11 a24 a32 a43 + a12 a21 a34 a43 - a11 a22 a34 a43 +
    a14 (a23 a32 a41 - a22 a33 a41 - a23 a31 a42 + a21 a33 a42 + a22 a31 a43 - a21 a32 a43) +
    a12 a23 a31 a44 - a11 a23 a32 a44 - a12 a21 a33 a44 + a11 a22 a33 a44 +
    a13 (-a24 a32 a41 + a22 a34 a41 + a24 a31 a42 - a21 a34 a42 - a22 a31 a44 + a21 a32 a44))
```

In[33]:= **Simplify[xv[[2]]]**

Out[33]=
```
(a21 a34 a43 rhs1 - a21 a33 a44 rhs1 - a14 a33 a41 rhs2 + a13 a34 a41 rhs2 +
    a14 a31 a43 rhs2 - a11 a34 a43 rhs2 - a13 a31 a44 rhs2 + a11 a33 a44 rhs2 -
    a14 a21 a43 rhs3 + a13 a21 a44 rhs3 + a14 a21 a33 rhs4 - a13 a21 a34 rhs4 + a24
     (a33 a41 rhs1 - a31 a43 rhs1 - a13 a41 rhs3 + a11 a43 rhs3 + a13 a31 rhs4 - a11 a33 rhs4) +
    a23 (-a34 a41 rhs1 + a31 a44 rhs1 + a14 a41 rhs3 -
        a11 a44 rhs3 - a14 a31 rhs4 + a11 a34 rhs4)) /
  (a12 a24 a33 a41 - a12 a23 a34 a41 - a11 a24 a33 a42 + a11 a23 a34 a42 -
    a12 a24 a31 a43 + a11 a24 a32 a43 + a12 a21 a34 a43 - a11 a22 a34 a43 +
    a14 (a23 a32 a41 - a22 a33 a41 - a23 a31 a42 + a21 a33 a42 + a22 a31 a43 - a21 a32 a43) +
    a12 a23 a31 a44 - a11 a23 a32 a44 - a12 a21 a33 a44 + a11 a22 a33 a44 +
    a13 (-a24 a32 a41 + a22 a34 a41 + a24 a31 a42 - a21 a34 a42 - a22 a31 a44 + a21 a32 a44))
```

In[34]:= **Simplify[xv[[3]]]**

Out[34]=
```
(-a21 a34 a42 rhs1 + a21 a32 a44 rhs1 + a14 a32 a41 rhs2 - a12 a34 a41 rhs2 - a14 a31 a42 rhs2 +
    a11 a34 a42 rhs2 + a12 a31 a44 rhs2 - a11 a32 a44 rhs2 + a14 a21 a42 rhs3 -
    a12 a21 a44 rhs3 - a14 a21 a32 rhs4 + a12 a21 a34 rhs4 + a24 (-a32 a41 rhs1 +
        a31 a42 rhs1 + a12 a41 rhs3 - a11 a42 rhs3 - a12 a31 rhs4 + a11 a32 rhs4) + a22
     (a34 a41 rhs1 - a31 a44 rhs1 - a14 a41 rhs3 + a11 a44 rhs3 + a14 a31 rhs4 - a11 a34 rhs4)) /
  (a12 a24 a33 a41 - a12 a23 a34 a41 - a11 a24 a33 a42 + a11 a23 a34 a42 -
    a12 a24 a31 a43 + a11 a24 a32 a43 + a12 a21 a34 a43 - a11 a22 a34 a43 +
    a14 (a23 a32 a41 - a22 a33 a41 - a23 a31 a42 + a21 a33 a42 + a22 a31 a43 - a21 a32 a43) +
    a12 a23 a31 a44 - a11 a23 a32 a44 - a12 a21 a33 a44 + a11 a22 a33 a44 +
    a13 (-a24 a32 a41 + a22 a34 a41 + a24 a31 a42 - a21 a34 a42 - a22 a31 a44 + a21 a32 a44))
```

In[35]:=
```
Simplify[xv[[4]]]
```

Out[35]=
```
(a21 a33 a42 rhs1 - a21 a32 a43 rhs1 - a13 a32 a41 rhs2 + a12 a33 a41 rhs2 +
   a13 a31 a42 rhs2 - a11 a33 a42 rhs2 - a12 a31 a43 rhs2 + a11 a32 a43 rhs2 -
   a13 a21 a42 rhs3 + a12 a21 a43 rhs3 + a13 a21 a32 rhs4 - a12 a21 a33 rhs4 + a23
    (a32 a41 rhs1 - a31 a42 rhs1 - a12 a41 rhs3 + a11 a42 rhs3 + a12 a31 rhs4 - a11 a32 rhs4) +
   a22 (-a33 a41 rhs1 + a31 a43 rhs1 + a13 a41 rhs3 -
      a11 a43 rhs3 - a13 a31 rhs4 + a11 a33 rhs4)) /
 (a12 a24 a33 a41 - a12 a23 a34 a41 - a11 a24 a33 a42 + a11 a23 a34 a42 -
   a12 a24 a31 a43 + a11 a24 a32 a43 + a12 a21 a34 a43 - a11 a22 a34 a43 +
   a14 (a23 a32 a41 - a22 a33 a41 - a23 a31 a42 + a21 a33 a42 + a22 a31 a43 - a21 a32 a43) +
   a12 a23 a31 a44 - a11 a23 a32 a44 - a12 a21 a33 a44 + a11 a22 a33 a44 +
   a13 (-a24 a32 a41 + a22 a34 a41 + a24 a31 a42 - a21 a34 a42 - a22 a31 a44 + a21 a32 a44))
```

Compute outflow

The last step is to compute the outflow. The repl vector is used which stores the outflow angular flux moments in terms of volume moments and inflow face moments.

In[36]:=
```
ψox0,0 = ψox0,0 /. repl; FullSimplify[ψox0,0]
```

Out[36]=
$$\frac{(-1+\alpha_0)\,\psi ix_{0,0}+2\,\psi_{0,0,0}+6\,sx\,\alpha_0\,\psi_{1,0,0}}{1+\alpha_0}$$

In[37]:=
```
ψox0,1 = ψox0,1 /. repl; Collect[ψox0,1, {ψix0,1, ψiz1,0, ψ0,0,1, ψ1,0,0}, Simplify]
```

Out[37]=
$$\frac{(-tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(36+tx\,tz)\,\gamma_1))\,\psi ix_{0,1}}{tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1)}-\frac{12\,sx\,tx\,\alpha_1\,\gamma_1\,\psi iz_{1,0}}{sz\,(tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1))}+\frac{2\,(tx\,tz+(tx\,tz-36\,\alpha_1)\,\gamma_1)\,\psi_{0,0,1}}{tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1)}+\frac{12\,sx\,tx\,\alpha_1\,\gamma_1\,\psi_{1,0,0}}{sz\,(tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1))}$$

In[38]:=
```
ψox1,0 = ψox1,0 /. repl; Collect[ψox1,0, {ψix1,0, ψiy1,0, ψ0,0,1, ψ0,1,0}, Simplify]
```

Out[38]=
$$\frac{(-tx\,ty\,(1+\beta_1)+\alpha_1\,(tx\,ty+(36+tx\,ty)\,\beta_1))\,\psi ix_{1,0}}{tx\,ty\,(1+\beta_1)+\alpha_1\,(tx\,ty+(-36+tx\,ty)\,\beta_1)}-\frac{12\,sx\,tx\,\alpha_1\,\beta_1\,\psi iy_{1,0}}{sy\,(tx\,ty\,(1+\beta_1)+\alpha_1\,(tx\,ty+(-36+tx\,ty)\,\beta_1))}+\frac{2\,(tx\,ty+(tx\,ty-36\,\alpha_1)\,\beta_1)\,\psi_{0,1,0}}{tx\,ty\,(1+\beta_1)+\alpha_1\,(tx\,ty+(-36+tx\,ty)\,\beta_1)}+\frac{12\,sx\,tx\,\alpha_1\,\beta_1\,\psi_{1,0,0}}{sy\,(tx\,ty\,(1+\beta_1)+\alpha_1\,(tx\,ty+(-36+tx\,ty)\,\beta_1))}$$

In[39]:= $\psi oy_{0,0} = \psi oy_{0,0}$ /. repl; Simplify$\left[\psi oy_{0,0}\right]$

Out[39]= $$\frac{(-1+\beta_0)\ \psi iy_{0,0} + 2\ (\psi_{0,0,0} + 3\ sy\ \beta_0\ \psi_{0,1,0})}{1+\beta_0}$$

In[40]:= $\psi oy_{0,1} = \psi oy_{0,1}$ /. repl; Collect$\left[\psi oy_{0,1}, \{\psi iy_{0,1}, \psi iy_{0,1}, \psi_{0,1,0}, \psi_{0,0,1}\}, \text{Simplify}\right]$

Out[40]= $$\frac{(-ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(36+ty\ tz)\ \gamma_1))\ \psi iy_{0,1}}{ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1)} -$$
$$\frac{12\ sy\ ty\ \beta_1\ \gamma_1\ \psi iz_{0,1}}{sz\ (ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1))} +$$
$$\frac{2\ (ty\ tz+(ty\ tz-36\ \beta_1)\ \gamma_1)\ \psi_{0,0,1}}{ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1)} +$$
$$\frac{12\ sy\ ty\ \beta_1\ \gamma_1\ \psi_{0,1,0}}{sz\ (ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1))}$$

In[41]:= $\psi oy_{1,0} = \psi oy_{1,0}$ /. repl; Collect$\left[\psi oy_{1,0}, \{\psi iy_{1,0}, \psi ix_{1,0}, \psi_{0,1,0}, \psi_{1,0,0}\}, \text{FullSimplify}\right]$

Out[41]= $$-\frac{12\ sy\ ty\ \alpha_1\ \beta_1\ \psi ix_{1,0}}{sx\ tx\ ty\ (1+\beta_1)+sx\ \alpha_1\ (tx\ ty+(-36+tx\ ty)\ \beta_1)} +$$
$$\frac{(-tx\ ty\ (1+\alpha_1)+(tx\ ty+(36+tx\ ty)\ \alpha_1)\ \beta_1)\ \psi iy_{1,0}}{tx\ ty\ (1+\beta_1)+\alpha_1\ (tx\ ty+(-36+tx\ ty)\ \beta_1)} +$$
$$\frac{12\ sy\ ty\ \alpha_1\ \beta_1\ \psi_{0,1,0}}{sx\ tx\ ty\ (1+\beta_1)+sx\ \alpha_1\ (tx\ ty+(-36+tx\ ty)\ \beta_1)} +$$
$$\frac{2\ (tx\ ty+\alpha_1\ (tx\ ty-36\ \beta_1))\ \psi_{1,0,0}}{tx\ ty\ (1+\beta_1)+\alpha_1\ (tx\ ty+(-36+tx\ ty)\ \beta_1)}$$

In[42]:= $\psi oz_{0,0} = \psi oz_{0,0}$ /. repl; FullSimplify$\left[\psi oz_{0,0}\right]$

Out[42]= $$\frac{(-1+\gamma_0)\ \psi iz_{0,0} + 2\ \psi_{0,0,0} + 6\ sz\ \gamma_0\ \psi_{0,0,1}}{1+\gamma_0}$$

In[43]:= $\psi oz_{0,1} = \psi oz_{0,1}$ /. repl; Collect$\left[\psi oz_{0,1}, \{\psi iz_{0,1}, \psi iy_{0,1}, \psi_{0,1,0}, \psi_{0,0,1}\}, \text{Simplify}\right]$

Out[43]= $$-\frac{12\ sz\ tz\ \beta_1\ \gamma_1\ \psi iy_{0,1}}{sy\ (ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1))} +$$
$$\frac{(ty\ tz\ (-1+\gamma_1)+\beta_1\ (-ty\ tz+(36+ty\ tz)\ \gamma_1))\ \psi iz_{0,1}}{ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1)} +$$
$$\frac{12\ sz\ tz\ \beta_1\ \gamma_1\ \psi_{0,0,1}}{sy\ (ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1))} +$$
$$\frac{2\ (ty\ tz+\beta_1\ (ty\ tz-36\ \gamma_1))\ \psi_{0,1,0}}{ty\ tz\ (1+\gamma_1)+\beta_1\ (ty\ tz+(-36+ty\ tz)\ \gamma_1)}$$

In[44]:=
```
ψoz1,0 = ψoz1,0 /. repl; Collect[ψoz1,0, {ψiz1,0, ψix0,1, ψ1,0,0, ψ0,0,1}, Simplify]
```

Out[44]=

$$-\frac{12\,sz\,tz\,\alpha_1\,\gamma_1\,\psi ix_{0,1}}{sx\,(tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1))}+\frac{(tx\,tz\,(-1+\gamma_1)+\alpha_1\,(-tx\,tz+(36+tx\,tz)\,\gamma_1))\,\psi iz_{1,0}}{tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1)}+\frac{12\,sz\,tz\,\alpha_1\,\gamma_1\,\psi_{0,0,1}}{sx\,(tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1))}+\frac{2\,(tx\,tz+\alpha_1\,(tx\,tz-36\,\gamma_1))\,\psi_{1,0,0}}{tx\,tz\,(1+\gamma_1)+\alpha_1\,(tx\,tz+(-36+tx\,tz)\,\gamma_1)}$$

D.4 AHOTN-1* Method

AHOTN equations

The AHOTN equations consist of two separate sets of equations, the balance equations and the auxiliary relations. Also, there are two separate sets of variables: the volume moments and the outflow face flux moments. The goal of this notebook is to derive an expression for the AHOTN local matrix T obtained by substituting expressions for the outflow face fluxes from the auxiliary equations into the balance equations. This system of equations is solved by dgesv for the volumetric flux moments.

Set the expansion order:

In[1]:=
```
Λ = 1;
```

Vectors of unknowns

In[2]:=
```
ψ = Flatten[Table[ϕ_{jx,jy,jz}, {jx, 0, Λ}, {jy, 0, Λ}, {jz, 0, Λ}]];
(* Volume Flux Moments*)
```

In[3]:=
```
ψx = Flatten[Table[ϕx_{jy,jz}, {jy, 0, Λ}, {jz, 0, Λ}]];
(* Outflow Flux Moments at +x face*)
```

In[4]:=
```
ψy = Flatten[Table[ϕy_{jx,jz}, {jx, 0, Λ}, {jz, 0, Λ}]];
```

In[5]:=
```
ψz = Flatten[Table[ϕz_{jx,jy}, {jx, 0, Λ}, {jy, 0, Λ}]];
```

In[6]:=
```
ψxm = Flatten[Table[ϕxm_{jy,jz}, {jy, 0, Λ}, {jz, 0, Λ}]];
(* Inflow Flux Moments at -x face*)
```

In[7]:=
```
ψym = Flatten[Table[ϕym_{jx,jz}, {jx, 0, Λ}, {jz, 0, Λ}]];
```

In[8]:=
```
ψzm = Flatten[Table[ϕzm_{jx,jy}, {jx, 0, Λ}, {jy, 0, Λ}]];
```

Formulate balance equations of order Λ

In[9]:=
```
balance = Flatten[Table[(Sign[μ])^jx/Abs[tx] ϕx_{jy,jz} + (Sign[η])^jy/Abs[ty] ϕy_{jx,jz} + (Sign[ξ])^jz/Abs[tz] ϕz_{jx,jy} +
  -2/tx Sum_{l=0}^{Floor[(jx-1)/2]} (2 jx - 4 l - 1) ϕ_{jx-(2 l+1),jy,jz}
  -2/ty Sum_{l=0}^{Floor[(jy-1)/2]} (2 jy - 4 l - 1) ϕ_{jx,jy-(2 l+1),jz}
  -2/tz Sum_{l=0}^{Floor[(jz-1)/2]} (2 jz - 4 l - 1) ϕ_{jx,jy,jz-(2 l+1)}
  + ϕ_{jx,jy,jz} == S_{jx,jy,jz} + (-1)^jx (Sign[μ])^jx/Abs[tx] ϕxm_{jy,jz} + (-1)^jy (Sign[η])^jy/Abs[ty] ϕym_{jx,jz} +
  (-1)^jz (Sign[ξ])^jz/Abs[tz] ϕzm_{jx,jy}, {jx, 0, Λ}, {jy, 0, Λ}, {jz, 0, Λ}];
```

Formulate WDD equations

■ In x - direction

In[10]:=
```
WDDx = Flatten[Table[(1 + Abs[αx])/2 ϕx_{jy,jz} -
  Sum[(2 jx + 1) ϕ_{jx,jy,jz}, {jx, 0, Λ, 2}] -
  Sum[(2 jx + 1) Sign[μ] Abs[αx] ϕ_{jx,jy,jz}, {jx, 1, Λ, 2}]
  == -(1 - Abs[αx])/2 ϕxm_{jy,jz}, {jy, 0, Λ}, {jz, 0, Λ}]];
```

■ In y - direction

In[11]:=
```
WDDy = Flatten[Table[(1 + Abs[αy])/2 ϕy_{jx,jz} -
  Sum[(2 jy + 1) ϕ_{jx,jy,jz}, {jy, 0, Λ, 2}] -
  Sum[(2 jy + 1) Sign[η] Abs[αy] ϕ_{jx,jy,jz}, {jy, 1, Λ, 2}]
  == -(1 - Abs[αy])/2 ϕym_{jx,jz}, {jx, 0, Λ}, {jz, 0, Λ}]];
```

■ In z - direction

```
In[12]:= WDDz = Flatten[Table[(1 + Abs[αz])/2 φz_{jx,jy} -
              Sum[(2 jz + 1) φ_{jx,jy,jz}, {jz, 0, Λ, 2}] -
              Sum[(2 jz + 1) Sign[ξ] Abs[αz] φ_{jx,jy,jz}, {jz, 1, Λ, 2}]
            == -(1 - Abs[αz])/2 φzm_{jx,jy}, {jx, 0, Λ}, {jy, 0, Λ}]];
```

Solve WDD for the outflow moments and replace in balance equations.

```
In[13]:= replx = First[Solve[WDDx, ψx]];

In[14]:= reply = First[Solve[WDDy, ψy]];

In[15]:= replz = First[Solve[WDDz, ψz]];
```

Replace outflow unknowns in Balance and collect matrix T

```
In[16]:= newbalance = balance /. Join[replx, reply, replz];

In[17]:= T = Normal[CoefficientArrays[newbalance, ψ][[2]]];
```

The matrix T (rotated 90deg)

```
In[18]:= (*Rotate[Simplify[T]//MatrixForm,π/2]*)
```

The corresponding rhs:

```
In[19]:= rhs = Normal[CoefficientArrays[newbalance, ψ][[1]]];

In[20]:= (*Rotate[Simplify[rhs//MatrixForm],π/2]*)
```

Relations to obtain the Outflow given the inflow

In[21]:= **MatrixForm[Join[replx, reply, replz]]**

Out[21]//MatrixForm=

$$\begin{pmatrix}
\phi x_{0,0} \to \frac{-\phi xm_{0,0}+\text{Abs}[\alpha x]\ \phi xm_{0,0}+2\ \phi_{0,0,0}+6\ \text{Abs}[\alpha x]\ \text{Sign}[\mu]\ \phi_{1,0,0}}{1+\text{Abs}[\alpha x]} \\
\phi x_{0,1} \to \frac{-\phi xm_{0,1}+\text{Abs}[\alpha x]\ \phi xm_{0,1}+2\ \phi_{0,0,1}+6\ \text{Abs}[\alpha x]\ \text{Sign}[\mu]\ \phi_{1,0,1}}{1+\text{Abs}[\alpha x]} \\
\phi x_{1,0} \to \frac{-\phi xm_{1,0}+\text{Abs}[\alpha x]\ \phi xm_{1,0}+2\ \phi_{0,1,0}+6\ \text{Abs}[\alpha x]\ \text{Sign}[\mu]\ \phi_{1,1,0}}{1+\text{Abs}[\alpha x]} \\
\phi x_{1,1} \to \frac{-\phi xm_{1,1}+\text{Abs}[\alpha x]\ \phi xm_{1,1}+2\ \phi_{0,1,1}+6\ \text{Abs}[\alpha x]\ \text{Sign}[\mu]\ \phi_{1,1,1}}{1+\text{Abs}[\alpha x]} \\
\phi y_{0,0} \to \frac{-\phi ym_{0,0}+\text{Abs}[\alpha y]\ \phi ym_{0,0}+2\ \phi_{0,0,0}+6\ \text{Abs}[\alpha y]\ \text{Sign}[\eta]\ \phi_{0,1,0}}{1+\text{Abs}[\alpha y]} \\
\phi y_{0,1} \to \frac{-\phi ym_{0,1}+\text{Abs}[\alpha y]\ \phi ym_{0,1}+2\ \phi_{0,0,1}+6\ \text{Abs}[\alpha y]\ \text{Sign}[\eta]\ \phi_{0,1,1}}{1+\text{Abs}[\alpha y]} \\
\phi y_{1,0} \to \frac{-\phi ym_{1,0}+\text{Abs}[\alpha y]\ \phi ym_{1,0}+2\ \phi_{1,0,0}+6\ \text{Abs}[\alpha y]\ \text{Sign}[\eta]\ \phi_{1,1,0}}{1+\text{Abs}[\alpha y]} \\
\phi y_{1,1} \to \frac{-\phi ym_{1,1}+\text{Abs}[\alpha y]\ \phi ym_{1,1}+2\ \phi_{1,0,1}+6\ \text{Abs}[\alpha y]\ \text{Sign}[\eta]\ \phi_{1,1,1}}{1+\text{Abs}[\alpha y]} \\
\phi z_{0,0} \to \frac{-\phi zm_{0,0}+\text{Abs}[\alpha z]\ \phi zm_{0,0}+2\ \phi_{0,0,0}+6\ \text{Abs}[\alpha z]\ \text{Sign}[\xi]\ \phi_{0,0,1}}{1+\text{Abs}[\alpha z]} \\
\phi z_{0,1} \to \frac{-\phi zm_{0,1}+\text{Abs}[\alpha z]\ \phi zm_{0,1}+2\ \phi_{0,1,0}+6\ \text{Abs}[\alpha z]\ \text{Sign}[\xi]\ \phi_{0,1,1}}{1+\text{Abs}[\alpha z]} \\
\phi z_{1,0} \to \frac{-\phi zm_{1,0}+\text{Abs}[\alpha z]\ \phi zm_{1,0}+2\ \phi_{1,0,0}+6\ \text{Abs}[\alpha z]\ \text{Sign}[\xi]\ \phi_{1,0,1}}{1+\text{Abs}[\alpha z]} \\
\phi z_{1,1} \to \frac{-\phi zm_{1,1}+\text{Abs}[\alpha z]\ \phi zm_{1,1}+2\ \phi_{1,1,0}+6\ \text{Abs}[\alpha z]\ \text{Sign}[\xi]\ \phi_{1,1,1}}{1+\text{Abs}[\alpha z]}
\end{pmatrix}$$

Appendix E

Mathematica Scripts Pertaining to Method's Diffusion Limit

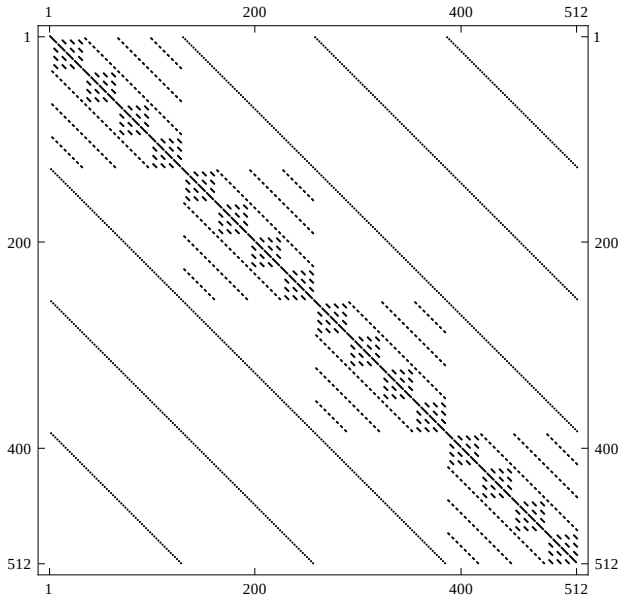
Several Mathematica notebooks for checking if the HODD-1, DD, AHOTN-1, LL and LN methods possess the diffusion limit.

E.1 HODD-1 Method

Construction of B matrix

```
ncells = 4; dofpc = 8;

size = ncells^3 * dofpc;

column[ix_, iy_, iz_, mx_, my_, mz_] =
  (ix + ncells * (iy - 1) + ncells^2 * (iz - 1) - 1) * dofpc + mz + 1 + my * 2 + mx * 4 ;

mat = Table[0, {row, 1, size}, {col, 1, size}];

Do[ row =
   (ix + ncells * (iy - 1) + ncells^2 * (iz - 1) - 1) * dofpc + mz + 1 + my * 2 + mx * 4 ;
  (* **** sx contributions **** *)
  If[mx == 0 ,
   (*mx==0*) mat[[row, row]] = mat[[row, row]] + 1;,
   (*mx≠0*) Do[mat[[row, column[ix - l, iy, iz, mx, my, mz]]] =
     mat[[row, column[ix - l, iy, iz, mx, my, mz]]] + 1, {l, 1, ix - 1}] ;
           Do[mat[[row, column[ix + l, iy, iz, mx, my, mz]]] =
     mat[[row, column[ix + l, iy, iz, mx, my, mz]]] - 1, {l, 1, ncells - ix}]; ];
  (* **** sy contributions **** *)
  If[my == 0 ,
   (*my==0*) mat[[row, row]] = mat[[row, row]] + 1;,
   (*my≠0*) Do[mat[[row, column[ix, iy - l, iz, mx, my, mz]]] =
     mat[[row, column[ix, iy - l, iz, mx, my, mz]]] + 1, {l, 1, iy - 1}] ;
           Do[mat[[row, column[ix, iy + l, iz, mx, my, mz]]] =
     mat[[row, column[ix, iy + l, iz, mx, my, mz]]] - 1, {l, 1, ncells - iy}]; ];
  (* **** sz contributions **** *)
  If[mz == 0 ,
   (*my==0*) mat[[row, row]] = mat[[row, row]] + 1;,
   (*my≠0*) Do[mat[[row, column[ix, iy, iz - l, mx, my, mz]]] =
     mat[[row, column[ix, iy, iz - l, mx, my, mz]]] + 1, {l, 1, iz - 1}] ;
           Do[mat[[row, column[ix, iy, iz + l, mx, my, mz]]] =
     mat[[row, column[ix, iy, iz + l, mx, my, mz]]] - 1, {l, 1, ncells - iz}]; ];
  , {iz, 1, ncells}, {iy, 1, ncells}, {ix, 1, ncells}, {mx, 0, 1}, {my, 0, 1}, {mz, 0, 1}];
```

```
MatrixPlot[mat, ColorFunction → "Monochrome"]
```




```
MatrixRank[mat]
```

512

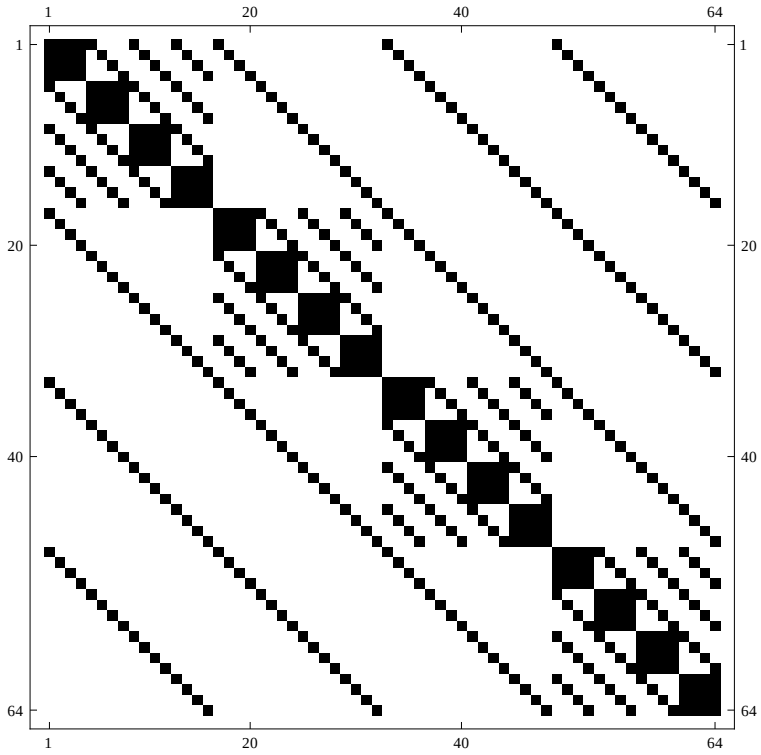
E.2 Diamond Difference Method

Three Spatial Dimensions

```
n = 4;

M = Table[0, {i, 1, n^3}, {j, 1, n^3}];

Do [
 row = ix + (iy - 1) * n + (iz - 1) n^2;
 M[[row, row]] = 1;
 Do[col = (ix - p) + (iy - 1) * n + (iz - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, ix - 1}];
 Do[col = (ix + p) + (iy - 1) * n + (iz - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, n - ix}];
 Do[col = (ix) + ((iy - p) - 1) * n + (iz - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, iy - 1}];
 Do[col = (ix) + ((iy + p) - 1) * n + (iz - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, n - iy}];
 Do[col = (ix) + ((iy) - 1) * n + ((iz - p) - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, iz - 1}];
 Do[col = (ix) + ((iy) - 1) * n + ((iz + p) - 1) n^2; M[[row, col]] = (-1)^p, {p, 1, n - iz}];
 , {iz, 1, n}, {iy, 1, n}, {ix, 1, n}]

MatrixRank[M]
```

64

```
MatrixPlot[M, ColorFunction → "Monochrome"]
```




```
M // MatrixForm;
```

E.3 AHOTN-1 Method

Construction of B matrix

Note that boundary cells are considered in this implementation by checking if the neighboring cell exist. If not then the corresponding
contributions are not added which is consistent with the derived equations.

In[1]:=
```
ncells = 4; dofpc = 8;
```

In[2]:=
```
size = ncells^3 * dofpc;
```

In[3]:=
```
column[ix_, iy_, iz_, mx_, my_, mz_] =
  (ix + ncells * (iy - 1) + ncells^2 * (iz - 1) - 1) * dofpc + mz + 1 + my * 2 + mx * 4 ;
```

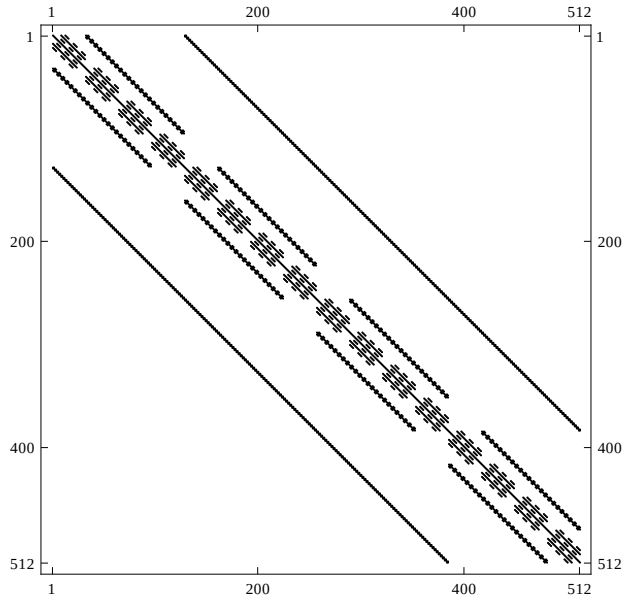
In[4]:=
```
mat = Table[0, {row, 1, size}, {col, 1, size}];
```

In[5]:=
```
Do[ row =
   (ix + ncells * (iy - 1) + ncells^2 * (iz - 1) - 1) * dofpc + mz + 1 + my * 2 + mx * 4 ;
  (*Now adding sx+ contributions*)
  mat[[row, column[ix, iy, iz, 0, my, mz]]]   =
   mat[[row, column[ix, iy, iz, 0, my, mz]]] + 1;
  If[ix > 1, mat[[row, column[ix - 1, iy, iz, 0, my, mz]]] =
    mat[[row, column[ix - 1, iy, iz, 0, my, mz]]] - (-1)^mx];
  mat[[row, column[ix, iy, iz, 1, my, mz]]]   =
   mat[[row, column[ix, iy, iz, 1, my, mz]]] + 3;
  If[ix > 1, mat[[row, column[ix - 1, iy, iz, 1, my, mz]]] =
    mat[[row, column[ix - 1, iy, iz, 1, my, mz]]] - 3 (-1)^mx];
  (*Now adding sx- contributions*)
  mat[[row, column[ix, iy, iz, 0, my, mz]]]   =
   mat[[row, column[ix, iy, iz, 0, my, mz]]] + (-1)^mx;
  If[ix < ncells, mat[[row, column[ix + 1, iy, iz, 0, my, mz]]] =
    mat[[row, column[ix + 1, iy, iz, 0, my, mz]]] - 1];
  mat[[row, column[ix, iy, iz, 1, my, mz]]]   =
   mat[[row, column[ix, iy, iz, 1, my, mz]]] - 3 (-1)^mx;
  If[ix < ncells, mat[[row, column[ix + 1, iy, iz, 1, my, mz]]] =
    mat[[row, column[ix + 1, iy, iz, 1, my, mz]]] + 3];
  (*Now adding sy+ contributions*)
  mat[[row, column[ix, iy, iz, mx, 0, mz]]]   =
   mat[[row, column[ix, iy, iz, mx, 0, mz]]] + 1;
  If[iy > 1, mat[[row, column[ix, iy - 1, iz, mx, 0, mz]]] =
    mat[[row, column[ix, iy - 1, iz, mx, 0, mz]]] - (-1)^my];
  mat[[row, column[ix, iy, iz, mx, 1, mz]]]   =
   mat[[row, column[ix, iy, iz, mx, 1, mz]]] + 3;
  If[iy > 1, mat[[row, column[ix, iy - 1, iz, mx, 1, mz]]] =
    mat[[row, column[ix, iy - 1, iz, mx, 1, mz]]] - 3 (-1)^my];
  (*Now adding sy- contributions*)
  mat[[row, column[ix, iy, iz, mx, 0, mz]]]   =
```

```
 mat[[row, column[ix, iy, iz, mx, 0, mz]]] + (-1)^my;
If[iy < ncells, mat[[row, column[ix, iy + 1, iz, mx, 0, mz]]] =
  mat[[row, column[ix, iy + 1, iz, mx, 0, mz]]] - 1];
mat[[row, column[ix, iy, iz, mx, 1, mz]]]   =
 mat[[row, column[ix, iy, iz, mx, 1, mz]]] - 3 (-1)^my;
If[iy < ncells, mat[[row, column[ix, iy + 1, iz, mx, 1, mz]]] =
  mat[[row, column[ix, iy + 1, iz, mx, 1, mz]]] + 3];
(*Now adding sz+ contributions*)
mat[[row, column[ix, iy, iz, mx, my, 0]]]   =
 mat[[row, column[ix, iy, iz, mx, my, 0]]] + 1;
If[iz > 1, mat[[row, column[ix, iy, iz - 1, mx, my, 0]]] =
  mat[[row, column[ix, iy, iz - 1, mx, my, 0]]] - (-1)^mz];
mat[[row, column[ix, iy, iz, mx, my, 1]]]   =
 mat[[row, column[ix, iy, iz, mx, my, 1]]] + 3;
If[iz > 1, mat[[row, column[ix, iy, iz - 1, mx, my, 1]]] =
  mat[[row, column[ix, iy, iz - 1, mx, my, 1]]] - 3 (-1)^mz];
(*Now adding sz- contributions*)
mat[[row, column[ix, iy, iz, mx, my, 0]]]   =
 mat[[row, column[ix, iy, iz, mx, my, 0]]] + (-1)^mz;
If[iz < ncells, mat[[row, column[ix, iy, iz + 1, mx, my, 0]]] =
  mat[[row, column[ix, iy, iz + 1, mx, my, 0]]] - 1];
mat[[row, column[ix, iy, iz, mx, my, 1]]]   =
 mat[[row, column[ix, iy, iz, mx, my, 1]]] - 3 (-1)^mz;
If[iz < ncells, mat[[row, column[ix, iy, iz + 1, mx, my, 1]]] =
  mat[[row, column[ix, iy, iz + 1, mx, my, 1]]] + 3];
 , {iz, 1, ncells}, {iy, 1, ncells}, {ix, 1, ncells}, {mx, 0, 1}, {my, 0, 1}, {mz, 0, 1}];
```

In[6]:=
```
MatrixPlot[mat, ColorFunction → "Monochrome"]
```

Out[6]=



In[7]:= **MatrixRank[mat]**

Out[7]= 485

E.4 LL and LN Methods

Construction of B matrix

Note that boundary cells are considered in this implementation by checking if the neighboring cell exist. If not then the corresponding
contributions are not added which is consistent with the derived equations.

In[1]:=
```
ncells = 4; dofpc = 4;
```

In[2]:=
```
size = ncells^3 * dofpc;
```

In[3]:=
```
subpos[mx_, my_, mz_] =
  Switch[{mx, my, mz}, {0, 0, 0}, 1, {0, 0, 1}, 2, {0, 1, 0}, 3, {1, 0, 0}, 4];
```

In[4]:=
```
pos[ix_, iy_, iz_, mx_, my_, mz_] =
  (ix + ncells * (iy - 1) + ncells^2 * (iz - 1) - 1) * dofpc + subpos[mx, my, mz] ;
```

In[5]:=
```
mat = Table[0, {row, 1, size}, {col, 1, size}];
```

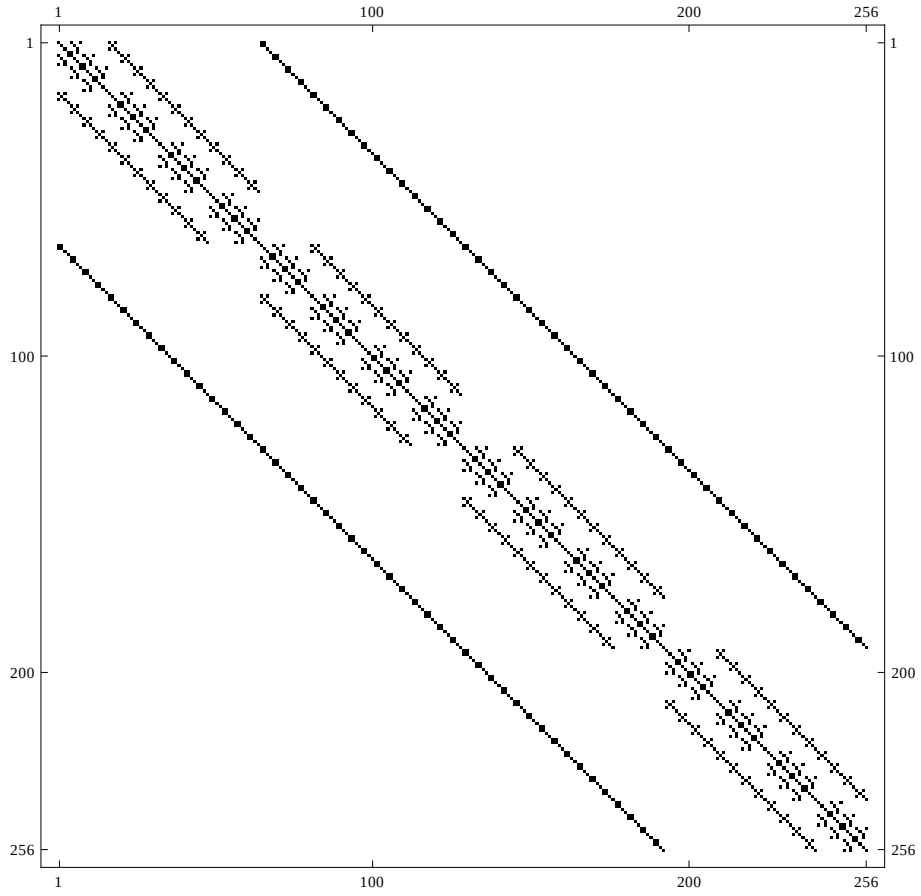
In[6]:=
```
Do[ row = pos[ix, iy, iz, mx, my, mz] ;
  (****Case (0,0,0)****)
  If[subpos[mx, my, mz] == 1,
    (* k=x *)
   mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] = mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] + 2;
   If[ix > 1,     mat[[row, pos[ix - 1, iy, iz, 0, 0, 0] ]] =
     mat[[row, pos[ix - 1, iy, iz, 0, 0, 0] ]] - 1];
   If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 0, 0, 0] ]] =
     mat[[row, pos[ix + 1, iy, iz, 0, 0, 0] ]] - 1];
   If[ix > 1,     mat[[row, pos[ix - 1, iy, iz, 1, 0, 0] ]] =
     mat[[row, pos[ix - 1, iy, iz, 1, 0, 0] ]] - 3];
   If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 1, 0, 0] ]] =
     mat[[row, pos[ix + 1, iy, iz, 1, 0, 0] ]] + 3];
   (* k=y *)
   mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] = mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] + 2;
   If[iy > 1,     mat[[row, pos[ix, iy - 1, iz, 0, 0, 0] ]] =
     mat[[row, pos[ix, iy - 1, iz, 0, 0, 0] ]] - 1];
   If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 0, 0, 0] ]] =
     mat[[row, pos[ix, iy + 1, iz, 0, 0, 0] ]] - 1];
   If[iy > 1,     mat[[row, pos[ix, iy - 1, iz, 0, 1, 0] ]] =
     mat[[row, pos[ix, iy - 1, iz, 0, 1, 0] ]] - 3];
   If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 0, 1, 0] ]] =
     mat[[row, pos[ix, iy + 1, iz, 0, 1, 0] ]] + 3];
   (* k=z *)
   mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] = mat[[row, pos[ix, iy, iz, 0, 0, 0] ]] + 2;
   If[iz > 1,     mat[[row, pos[ix, iy, iz - 1, 0, 0, 0] ]] =
     mat[[row, pos[ix, iy, iz - 1, 0, 0, 0] ]] - 1];
   If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 0, 0, 0] ]] =
     mat[[row, pos[ix, iy, iz + 1, 0, 0, 0] ]] - 1];
```

```
 If[iz > 1,     mat[[row, pos[ix, iy, iz - 1, 0, 0, 1]]] =
   mat[[row, pos[ix, iy, iz - 1, 0, 0, 1]]] - 3];
 If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 0, 0, 1]]] =
   mat[[row, pos[ix, iy, iz + 1, 0, 0, 1]]] + 3];
];
(****Case (0,0,1)****)
If[subpos[mx, my, mz] == 2,
  (* k=x *)
 mat[[row, pos[ix, iy, iz, 0, 0, 1]]] = mat[[row, pos[ix, iy, iz, 0, 0, 1]]] + 2;
 If[ix > 1,     mat[[row, pos[ix - 1, iy, iz, 0, 0, 1]]] =
   mat[[row, pos[ix - 1, iy, iz, 0, 0, 1]]] - 1];
 If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 0, 0, 1]]] =
   mat[[row, pos[ix + 1, iy, iz, 0, 0, 1]]] - 1];
  (* k=y *)
 mat[[row, pos[ix, iy, iz, 0, 0, 1]]] = mat[[row, pos[ix, iy, iz, 0, 0, 1]]] + 2;
 If[iy > 1,     mat[[row, pos[ix, iy - 1, iz, 0, 0, 1]]] =
   mat[[row, pos[ix, iy - 1, iz, 0, 0, 1]]] - 1];
 If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 0, 0, 1]]] =
   mat[[row, pos[ix, iy + 1, iz, 0, 0, 1]]] - 1];
  (* k=z *)
 If[iz > 1,     mat[[row, pos[ix, iy, iz - 1, 0, 0, 0]]] =
   mat[[row, pos[ix, iy, iz - 1, 0, 0, 0]]] + 1];
 If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 0, 0, 0]]] =
   mat[[row, pos[ix, iy, iz + 1, 0, 0, 0]]] - 1];
 mat[[row, pos[ix, iy, iz, 0, 0, 1]]] = mat[[row, pos[ix, iy, iz, 0, 0, 1]]] + 6;
 If[iz > 1, mat[[row, pos[ix, iy, iz - 1, 0, 0, 1]]] =
   mat[[row, pos[ix, iy, iz - 1, 0, 0, 1]]] + 3];
 If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 0, 0, 1]]] =
   mat[[row, pos[ix, iy, iz + 1, 0, 0, 1]]] + 3];]
(****Case (0,1,0)****)
If[subpos[mx, my, mz] == 3,
  (* k=x *)
 mat[[row, pos[ix, iy, iz, 0, 1, 0]]] = mat[[row, pos[ix, iy, iz, 0, 1, 0]]] + 2;
 If[ix > 1,     mat[[row, pos[ix - 1, iy, iz, 0, 1, 0]]] =
   mat[[row, pos[ix - 1, iy, iz, 0, 1, 0]]] - 1];
 If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 0, 1, 0]]] =
   mat[[row, pos[ix + 1, iy, iz, 0, 1, 0]]] - 1];
  (* k=y *)
 If[iy > 1,     mat[[row, pos[ix, iy - 1, iz, 0, 0, 0]]] =
   mat[[row, pos[ix, iy - 1, iz, 0, 0, 0]]] + 1];
 If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 0, 0, 0]]] =
   mat[[row, pos[ix, iy + 1, iz, 0, 0, 0]]] - 1];
 mat[[row, pos[ix, iy, iz, 0, 1, 0]]] = mat[[row, pos[ix, iy, iz, 0, 1, 0]]] + 6;
 If[iy > 1, mat[[row, pos[ix, iy - 1, iz, 0, 1, 0]]] =
   mat[[row, pos[ix, iy - 1, iz, 0, 1, 0]]] + 3];
 If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 0, 1, 0]]] =
   mat[[row, pos[ix, iy + 1, iz, 0, 1, 0]]] + 3];
  (* k=z *)
 mat[[row, pos[ix, iy, iz, 0, 1, 0]]] = mat[[row, pos[ix, iy, iz, 0, 1, 0]]] + 2;
```

```
  If[iz > 1,     mat[[row, pos[ix, iy, iz - 1, 0, 1, 0]]] =
    mat[[row, pos[ix, iy, iz - 1, 0, 1, 0]]] - 1];
  If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 0, 1, 0]]] =
    mat[[row, pos[ix, iy, iz + 1, 0, 1, 0]]] - 1];
 ];
(****Case (1,0,0)****)
If[subpos[mx, my, mz] == 4,
  (* k=x *)
 If[ix > 1,     mat[[row, pos[ix - 1, iy, iz, 0, 0, 0]]] =
   mat[[row, pos[ix - 1, iy, iz, 0, 0, 0]]] + 1];
 If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 0, 0, 0]]] =
   mat[[row, pos[ix + 1, iy, iz, 0, 0, 0]]] - 1];
 mat[[row, pos[ix, iy, iz, 1, 0, 0]]] = mat[[row, pos[ix, iy, iz, 1, 0, 0]]] + 6;
 If[ix > 1, mat[[row, pos[ix - 1, iy, iz, 0, 1, 0]]] =
   mat[[row, pos[ix - 1, iy, iz, 0, 1, 0]]] + 3];
 If[ix < ncells, mat[[row, pos[ix + 1, iy, iz, 0, 1, 0]]] =
   mat[[row, pos[ix + 1, iy, iz, 0, 1, 0]]] + 3];
  (* k=y *)
 mat[[row, pos[ix, iy, iz, 1, 0, 0]]] = mat[[row, pos[ix, iy, iz, 1, 0, 0]]] + 2;
 If[iy > 1,     mat[[row, pos[ix, iy - 1, iz, 1, 0, 0]]] =
   mat[[row, pos[ix, iy - 1, iz, 1, 0, 0]]] - 1];
 If[iy < ncells, mat[[row, pos[ix, iy + 1, iz, 1, 0, 0]]] =
   mat[[row, pos[ix, iy + 1, iz, 1, 0, 0]]] - 1];
 (* k=z *)
 mat[[row, pos[ix, iy, iz, 1, 0, 0]]] = mat[[row, pos[ix, iy, iz, 1, 0, 0]]] + 2;
 If[iz > 1,     mat[[row, pos[ix, iy, iz - 1, 1, 0, 0]]] =
   mat[[row, pos[ix, iy, iz - 1, 1, 0, 0]]] - 1];
 If[iz < ncells, mat[[row, pos[ix, iy, iz + 1, 1, 0, 0]]] =
   mat[[row, pos[ix, iy, iz + 1, 1, 0, 0]]] - 1];
];
(*** Iterator ***)
 , {iz, 1, ncells}, {iy, 1, ncells}, {ix, 1, ncells}, {mx, 0, 1}, {my, 0, 1}, {mz, 0, 1}];
```

In[7]:=
```
MatrixPlot[mat, ColorFunction → "Monochrome"]
```

Out[7]=



In[8]:=
```
MatrixRank[mat]
```

Out[8]= 256

Appendix F

Validation Exercise Complete Set of Results

The simple score Γ as a number is not meaningful unless two consistently computed Γ values for two discretization methods are compared to each other. Therefore, the obtained predicted and actual scores are rescaled as follows. First, among all the predicted scores, i.e. for all participating methods and orders Λ, the maximum and minimum score $\Gamma_{\text{pred,max}}$ and $\Gamma_{\text{pred,min}}$ are determined. Then each score is scaled as (l runs over all methods and orders Λ):

$$\Gamma_{\text{pred},l} \leftarrow \frac{\Gamma_{\text{pred},l} - \Gamma_{\text{pred,min}}}{\Gamma_{\text{pred,max}} - \Gamma_{\text{pred,min}}}. \tag{F.1}$$

The same procedure is applied to the scores computed directly from results obtained for the NEA benchmark problem:

$$\Gamma_{\text{NEA},l} \leftarrow \frac{\Gamma_{\text{NEA},l} - \Gamma_{\text{NEA,min}}}{\Gamma_{\text{NEA,max}} - \Gamma_{\text{NEA,min}}}. \tag{F.2}$$

Both $\Gamma_{\text{pred},l}$ and $\Gamma_{\text{NEA},l}$ now range from zero to one with these values actually being assumed by one of the methods data/prediction points.

Most of the presented verification results utilize quantity 1.a as target quantity. It shall be implicitly assumed that the target quantity that the real world accuracies are computed for is 1.a unless otherwise noted.

In each of the following figures the upper two plots are *predicted* scores plotted versus the cell optical thickness and the lower two plots are *actual* scores plotted versus the cell optical thickness.

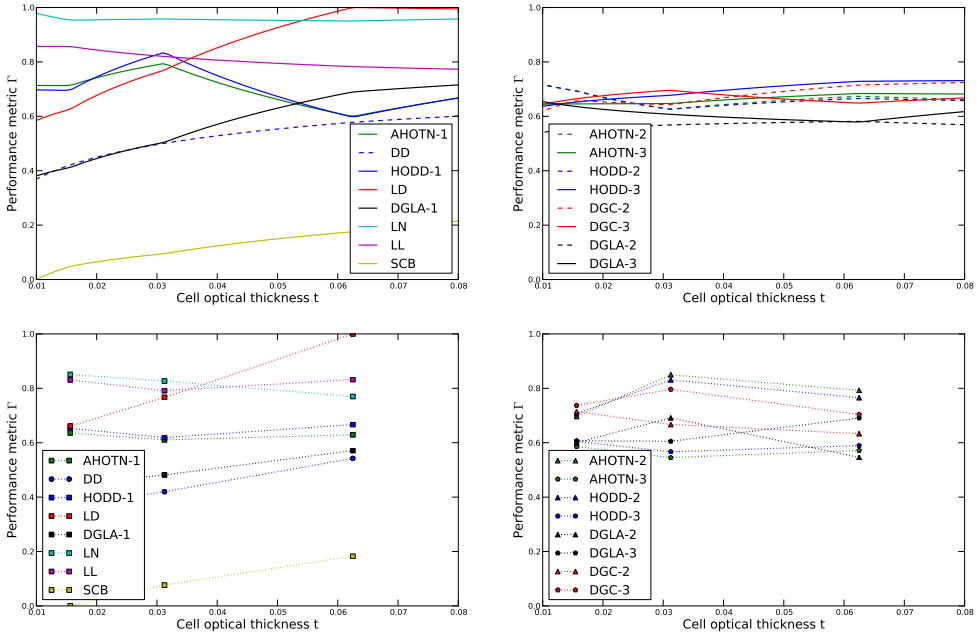


Figure F.1: Results of the validation exercise for NEA-I with $\vec{\beta} = (0, 0, 0, 1)$.

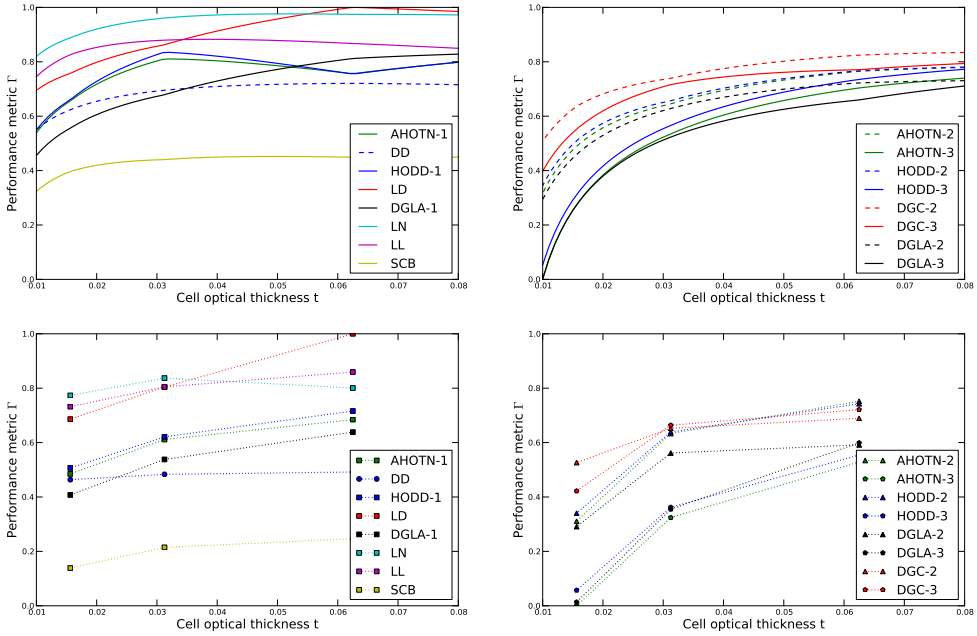


Figure F.2: Results of the validation exercise for NEA-I with $\vec{\beta} = (1, 0, 1, 0)$.

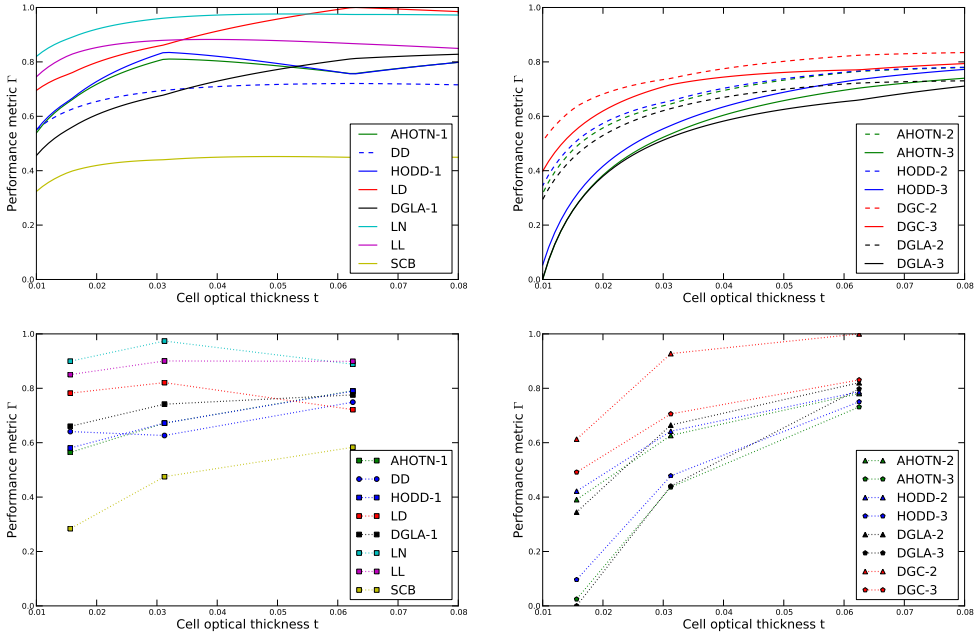


Figure F.3: Results of the validation exercise for NEA-I with $\vec{\beta} = (1, 0, 1, 0)$. Quantity 3.c is selected as target quantity.

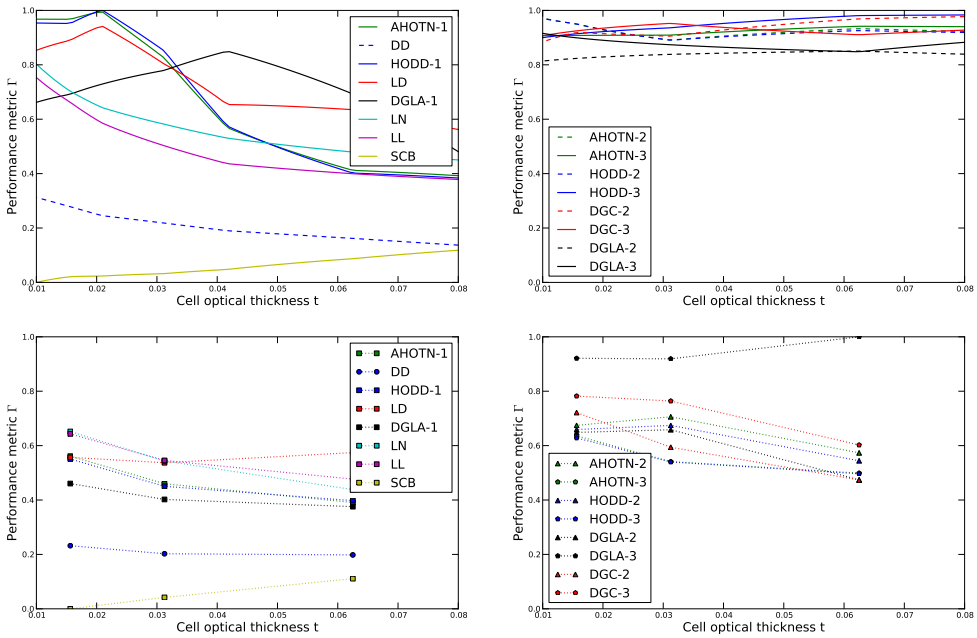


Figure F.4: Results of the validation exercise for NEA-I with $\vec{\beta} = (0, 1, 0, 2)$.

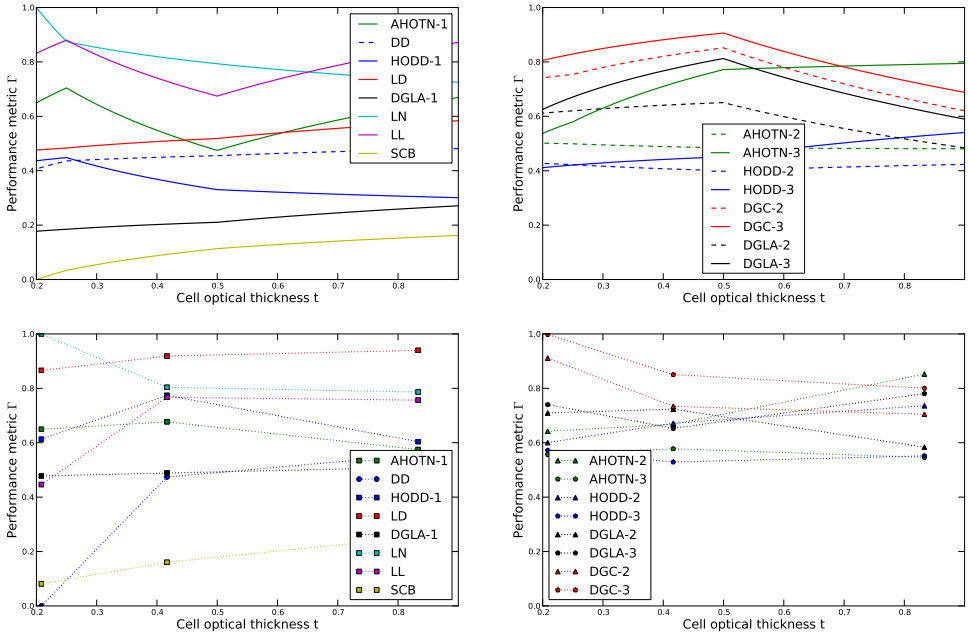


Figure F.5: Results of the validation exercise for NEA-II with $\vec{\beta} = (0, 0, 0, 1)$.

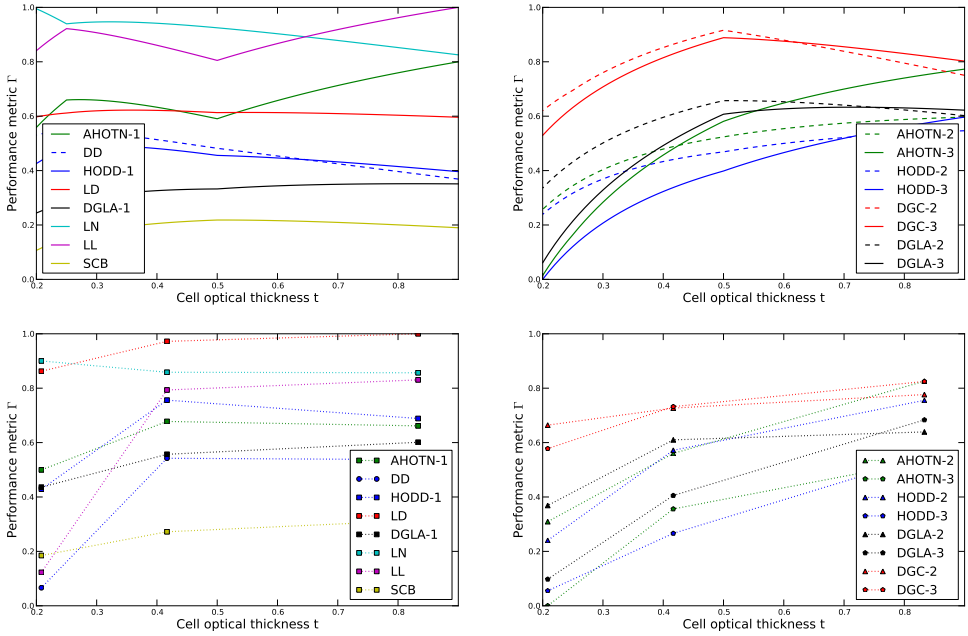


Figure F.6: Results of the validation exercise for NEA-II with $\vec{\beta} = (1, 0, 1, 0)$.

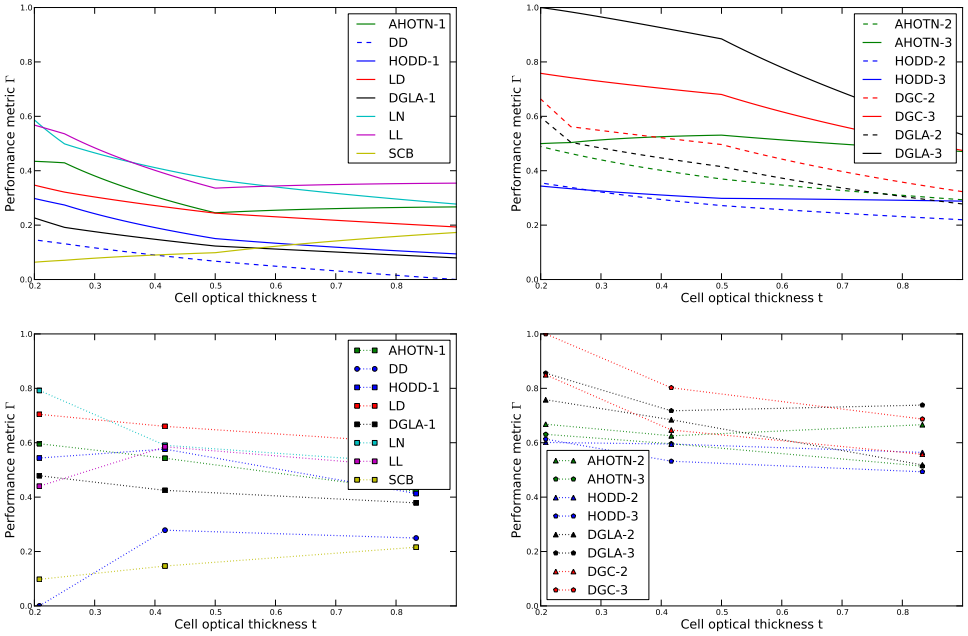


Figure F.7: Results of the validation exercise for NEA-II with $\vec{\beta} = (0, 1, 0, 2)$.

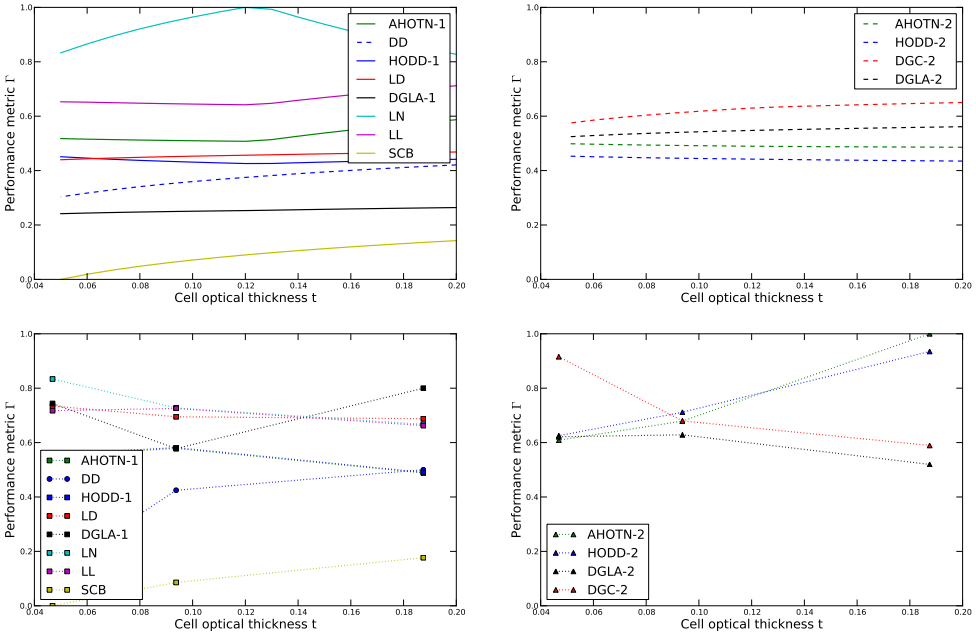


Figure F.8: Results of the validation exercise for NEA-III with $\vec{\beta} = (0, 0, 0, 1)$.

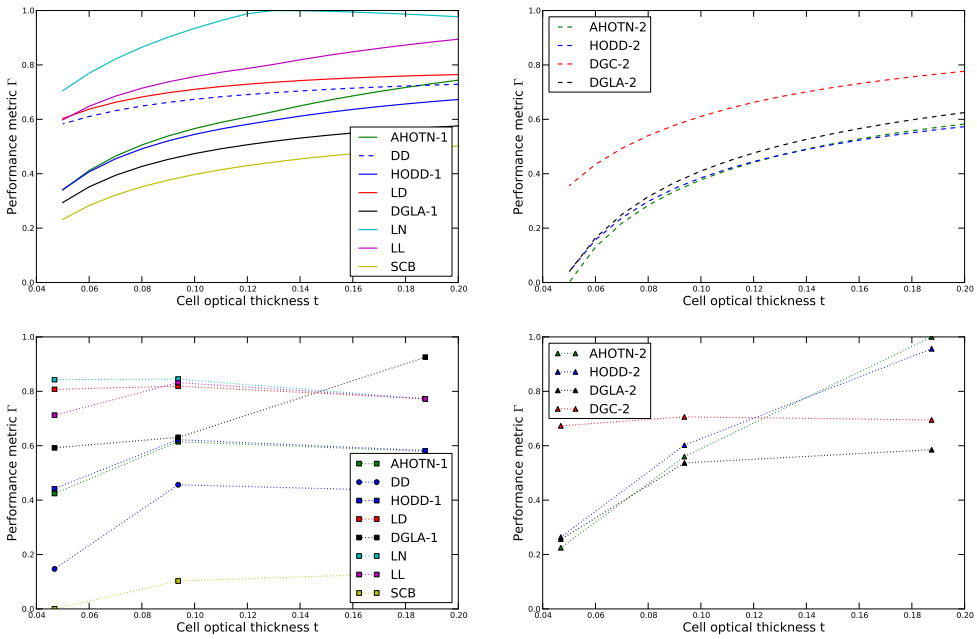


Figure F.9: Results of the validation exercise for NEA-III with $\vec{\beta} = (1, 0, 1, 0)$.

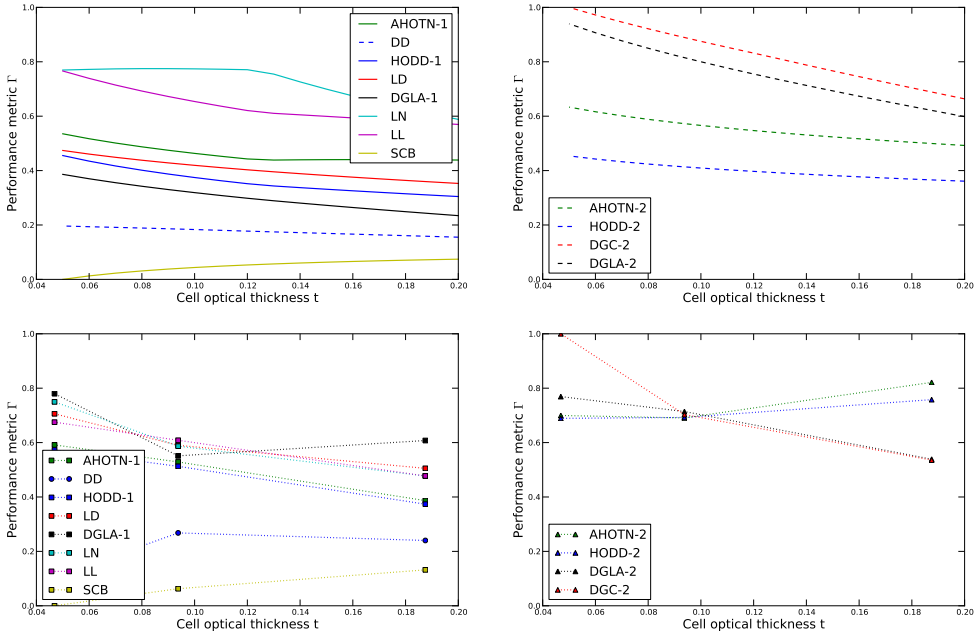


Figure F.10: Results of the validation exercise for NEA-III with $\vec{\beta} = (0, 1, 0, 2)$.

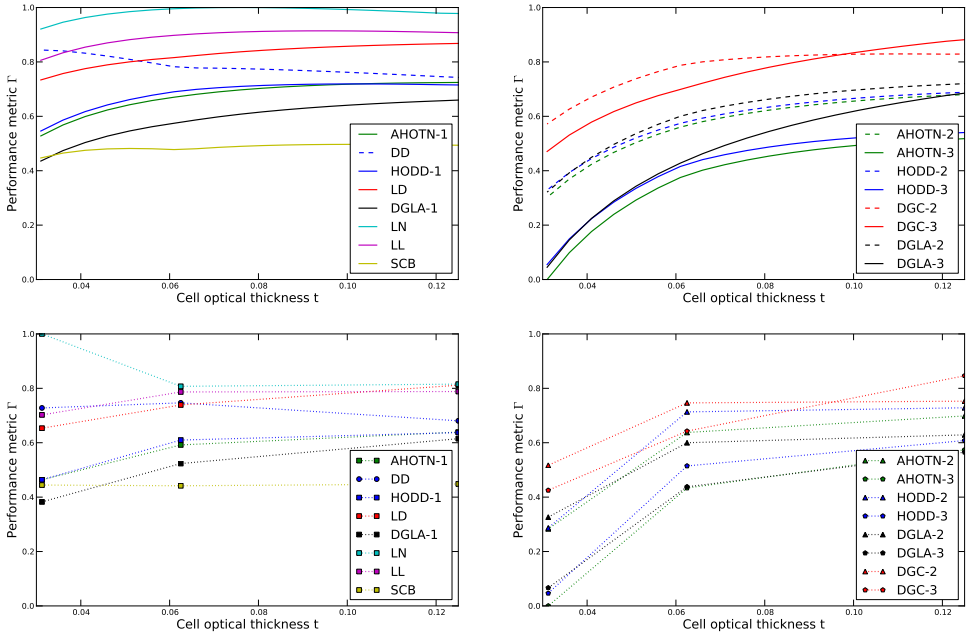


Figure F.11: Results of the validation exercise for NEA-IV with $\vec{\beta} = (1, 0, 1, 0)$.

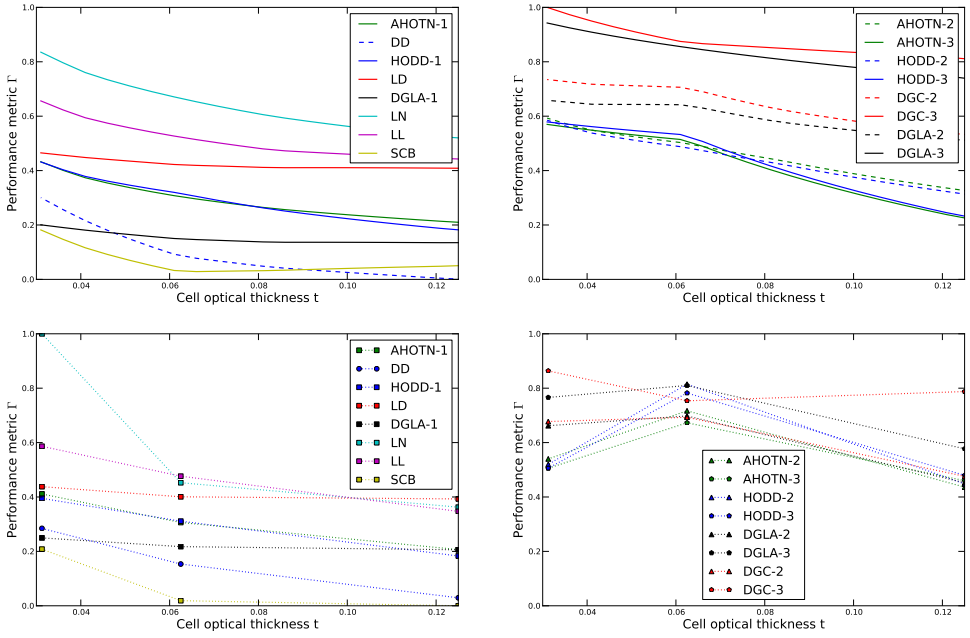


Figure F.12: Results of the validation exercise for NEA-IV with $\vec{\beta} = (0, 1, 0, 2)$.